

[PDF] 2022  
DOMINA CONCURSOS

APOSTILA

# CONCURSO

TOTALMENTE ELABORADA DE ACORDO COM EDITAL

NÍVEL SUPERIOR



**D**OMINA  
CONCURSOS

[WWW.DOMINACONCURSOS.COM.BR](http://WWW.DOMINACONCURSOS.COM.BR)

## QUEM SOMOS

A Domina Concursos, especialista há 8 anos no desenvolvimento e comercialização de apostilas digitais e impressas para Concurso Públicos, tem como foco tornar simples e eficaz a forma de estudo. Com visão de futuro, agilidade e dinamismo em inovações, se consolida com reconhecimento no segmento de desenvolvimento de materiais para concursos públicos. É uma empresa comprometida com o bem-estar do cliente. Atua com concursos públicos federais, estaduais e municipais. Em nossa trajetória, já comercializamos milhares de apostilas, sendo digitais e impressas. E esse número continua aumentando.

## MISSÃO

Otimizar a forma de estudo, provendo apostilas de excelência, baseados nas informações de editais dos concursos públicos, para incorporar as melhores práticas, com soluções inovadoras, flexíveis e de simples utilização e entendimento.

## VISÃO

Ser uma empresa de Classe Nacional em Desenvolvimento de Apostilas para Concursos Públicos, com paixão e garra em tudo que fazemos.

## VALORES

- Respeito ao talento humano
- Foco no cliente
- Integridade no relacionamento
- Equipe comprometida
- Evolução tecnológica permanente
- Ambiente diferenciado
- Responsabilidade social

## PROIBIDO CÓPIA

Não é permitida a revenda, rateio, cópia total ou parcial sem autorização da Domina Concursos, seja ela cópia virtual ou impressa. Independente de manter os créditos ou não, não importando o meio pelo qual seja disponibilizado: link de download, Correios, etc...

Caso houver descumprimento, o autor do fato poderá ser indiciado conforme art. 184 do CP, serão buscadas as informações do responsável em nosso banco de dados e repassadas para as autoridades responsáveis.

# Conhecimentos básicos

*"É melhor você tentar algo,  
vê-lo não funcionar e  
aprender com isso, do que  
não fazer nada."*

Mark Zuckerberg

# LÍNGUA PORTUGUESA

*“Todos os seus sonhos podem  
se tornar realidade se você tem  
coragem para persegui-los”*

*Walt Disney*

## Compreensão De Textos Variados

### Introdução

A compreensão da leitura na língua materna, segunda língua ou língua estrangeira, requer várias habilidades linguísticas e metalinguísticas inter-relacionadas que permitem ao leitor a produção do sentido do texto a partir de saberes nos níveis: léxico, morfosintático e semântico. Além disso, habilidades socioculturais – como o conhecimento de mundo – agregam-se ao domínio do sistema de escrita pelo leitor, possibilitando-lhe maior ou menor entendimento da leitura.

Entretanto, embora várias habilidades cognitivas sejam importantes para o desenvolvimento da leitura, as habilidades metacognitivas parecem primordiais para a aprendizagem formal na escola em virtude de sua natureza consciente (KATO, 1999). Nesse sentido, grande número de estudos tem evidenciado uma relação estreita entre a aprendizagem da leitura e as habilidades metalinguísticas (consciências fonológica, morfológica, léxica, sintática e semântica).

A importância da consciência fonológica para a aprendizagem da leitura na língua materna é atestada por estudos estrangeiros e brasileiros (BRADLEY; BRYANT, 1983; CARDOSO-MARTINS, 1995; GUIMARÃES, 2003; MALUF; BARRERA, 1997; MORAIS; ALEGRIA; CONTENT, 1987;

ROAZZI; DOWKER, 1989; TUNMER, 1990). No que se refere à consciência sintática, um dos primeiros estudos focalizando essa habilidade foi realizado por Bowey (1986) com falantes do inglês (língua materna). Os resultados do seu estudo mostraram que o uso de pistas gramaticais facilita o reconhecimento de palavras e a compreensão, tanto de frases quanto de textos. Em outra investigação, também realizada com falantes do inglês, Rego e Bryant (1993) verificaram a existência de uma relação positiva entre a consciência sintática e a leitura de palavras com dificuldades ortográficas.

Dentre os estudos realizados para investigar a influência das habilidades metalinguísticas (consciência fonológica e consciência sintática) sobre o desenvolvimento da capacidade de leitura em português (língua materna), destaca-se primeiramente o realizado por Rego (1995). Sua pesquisa visou investigar como ambas – a consciência fonológica e a consciência sintática – contribuem para explicar as diferenças individuais no desenvolvimento da decodificação e da compreensão na leitura. Os resultados dessa investigação não confirmam os resultados anteriormente encontrados com falantes do inglês, ou seja, os alunos brasileiros não se beneficiaram dessas habilidades metalinguísticas para desenvolver a decodificação na leitura. Porém, os resultados mostraram uma relação entre a consciência sintática e o desempenho em leitura, quando esta envolvia diretamente o uso de informações sintático-semânticas.

Contudo, o estudo desenvolvido por Capovilla, Capovilla e Soares (2004) com alunos brasileiros de 2º a 5º ano do Ensino Fundamental, que investigou consciência fonológica, consciência sintática, competência em leitura, escrita sob ditado e vocabulário, mostrou uma correlação positiva e significativa entre os escores gerais desses cinco testes. De acordo com os autores, seus resultados corroboram as evidências bibliográficas que apontam relações entre leitura, escrita e habilidades metalinguísticas.

Tunmer (1990) e Tunmer e Hoover (1992) explicam a relação entre consciência sintática e competência em leitura, sugerindo que: a) uma estratégia de leitura possível de ser utilizada pelo leitor sintaticamente e semanticamente consciente é inferir o significado das palavras desconhecidas encontradas no texto; b) a consciência sintática possibilita a monitoração eficiente do processo de compreensão (BOWEY, 1986). Assim, outra estratégia que o leitor sintaticamente consciente é capaz de empregar é checar se as palavras que ele está lendo combinam gramaticalmente com as outras palavras do texto. Um estudo que deu suporte a essa ideia foi realizado por Weber (1970 apud TUNMER; HOOVER, 1992), no qual ela analisou os erros cometidos por alunos da 1ª série na leitura oral e verificou que os bons leitores tinham mais facilidade que os maus leitores para reconhecerem e corrigirem seus erros relativos à leitura de palavras que não combinavam com o significado ou com a sintaxe das sentenças. Tal resultado sugere que os bons leitores têm mais consciência da própria compreensão ou incompreensão do texto.

No contexto da aprendizagem de uma língua estrangeira, Kato (1999), cuja língua materna é o japonês, comenta que, em sua aprendizagem de leitura do português do Brasil, recorreu às inferências para descobrir o significado de palavras desconhecidas. A autora afirma que um leitor

proficiente em sua língua materna pode compartilhar entre as duas línguas as estratégias e os procedimentos que utiliza para a leitura, compensando com vantagens o déficit no domínio linguístico da língua estrangeira. Além disso, a autora sugere que conhecer o universo cultural da língua em que o texto foi escrito, a utilização das pistas do próprio texto e de suas ilustrações pode facilitar o processo de compreensão do leitor não proficiente na língua estrangeira.

Kleiman (2004), ao pesquisar como ocorrem as estratégias de inferências léxicas na leitura de inglês por brasileiros, conclui que o conhecimento do vocabulário numa língua estrangeira é limitado e que a eficácia das estratégias de inferência de significado do léxico através do contexto é determinante no momento da compreensão. A capacidade do aluno de autorregular o processo de inferência léxica vê-se refletida na compreensão e na ampliação do seu vocabulário.

Em síntese, juntamente com Braggio (1992), sugere-se que no ato de leitura várias habilidades são requeridas, ou seja, o leitor precisa mobilizar não apenas seu conhecimento da língua, considerada holisticamente, mas também seu conhecimento de mundo, suas experiências etc., e é o conjunto dessas habilidades que possibilitará a compreensão integrada do texto.

É nesse contexto teórico que este estudo se insere, tendo como objetivo central pesquisar como se dá a compreensão da leitura em espanhol por estudantes brasileiros. As hipóteses propostas foram as seguintes:

- a) dentre os alunos participantes, os que apresentam melhor desempenho na compreensão da leitura de textos em português (língua materna) apresentarão também melhor desempenho nas tarefas de compreensão de textos em espanhol;
- b) o nível de conhecimento do vocabulário receptivo em espanhol estará significativamente relacionado à compreensão dos textos em espanhol;
- c) os alunos que demonstram maior capacidade para utilizar as informações (pistas) do texto no momento da leitura são mais capazes de identificar o significado das palavras dedutivamente;
- d) o desempenho dos participantes nas tarefas de avaliação das habilidades metassintáticas mostra-se significativamente relacionado ao desempenho na compreensão da leitura dos textos em espanhol.

### **Método**

Este estudo foi realizado em uma escola pública de Curitiba, Paraná, Brasil. Participaram desta investigação duas turmas de alunos, uma do 8º e outra do 9º ano, num total de 67 alunos de ambos os sexos, com idades variando de 12 a 18 anos. No momento da aplicação dos instrumentos de coleta de dados, os alunos haviam completado, respectivamente, dois e três anos letivos de aprendizagem do idioma espanhol como língua estrangeira.

A pesquisa foi realizada de forma que os instrumentos permitissem identificar os níveis de compreensão da leitura em português (língua materna) e em espanhol (língua estrangeira), bem como a compreensão do vocabulário em espanhol e as habilidades sintáticas dos participantes. Assim, foram aplicadas cinco provas:

- a) teste de vocabulário por imagens: PPVT-III Peabody (DUNN; ARRIBAS, 2006);
- b) quatro textos para avaliar a compreensão da leitura em espanhol: testes ECL-1 e ECL-2 (CRUZ, 2005a, b) e dois textos do teste diagnóstico psicopedagógico em leitura e escrita (DI SCALA; CANTÚ, 2003);
- a) dois textos para avaliação da compreensão da leitura de textos expositivos em português, um para o 8º e outro para o 9º ano, denominados texto de compreensão de leitura em português A e B respectivamente: TCLP\_A e TCLP\_B, pertencentes ao teste "Avaliação da compreensão leitora de textos expositivos: para fonoaudiólogos e psicopedagogos" de Saraiva, Moojen e Munarski (2009);
- b) uma tarefa, criada pelas próprias pesquisadoras, para avaliar a compressão do significado de 23 palavras retiradas dos textos utilizados para a compreensão da leitura em espanhol, sendo dez falsos

cognados (tarefa aplicada antes e após da leitura dos textos);

c) três tarefas de avaliação da consciência sintática: julgamento gramatical, correção gramatical e categorização de palavras. Essas provas foram desenvolvidas pelas pesquisadoras, usando como referência as Provas de Consciência Sintática (PCS) de Capovilla e Capovilla (2006), traduzidas e adaptadas.

O teste Evaluación de la Comprensión Lectora (ECL) apresenta dois níveis que abarcam todo o âmbito da Educação Primária: ECL1, para os alunos de 2º e 3º ano (6 a 8 anos). Investiga a capacidade para captar o sentido de textos escritos de uso habitual e para analisar alguns aspectos simples próprios de diferentes tipos de textos. Avalia o conhecimento do significado das palavras, de sinônimos e antônimos e, também, a compreensão do significado de frases e a capacidade para integrar as informações contidas num texto. Já o ECL-2, para os alunos de 3º a 6º ano (de 9 a 11 anos), oferece textos de maior dificuldade que os do ECL-1, a fim de avaliar o conhecimento de sinônimos e antônimos, do significado das palavras, e do significado de frases em sentido literal e em sentido figurado.

Ao analisar os testes, escolheu-se um texto do ECL-1 (denominado TCLE\_C: Texto C de compreensão leitora em espanhol) e outro do ECL-2 (TCLE\_D: Texto D de compreensão leitora em espanhol) considerando os conhecimentos adquiridos pelos alunos participantes do 8º e 9º ano da escola onde se realizou a investigação.

Também para a compreensão de texto em espanhol se elegeram dois textos (denominados TCLE\_A: Texto A de compreensão leitora em espanhol e TCLE\_B: Texto B de compreensão leitora em espanhol) do teste DIP (Ie) diagnóstico psicopedagógico em leitura e escrita. Com relação a esses textos, são requeridos dois níveis de compreensão, avaliados respectivamente por dois tipos de perguntas: o primeiro (constituído de três perguntas) visa reconstruir a organização superestrutural e macroestrutural do relato, enquanto o segundo (duas perguntas) indaga aspectos inferenciais em que o aluno, mediante indícios que lhe oferece o próprio texto, deve concluir sobre conteúdos não presentes explicitamente no texto, ainda que dedutíveis a partir de pistas.

Para manter o mesmo padrão de pontuação em todas as tarefas, os resultados obtidos pelos participantes nas diferentes tarefas foram transformados em percentagens. Os dados coletados passaram por diversos procedimentos de análises estatísticas. Quanto ao nível de significância, adotou-se como parâmetro um alfa de 0,05 – valores de p inferiores a este foram considerados estatisticamente significativos.

## Resultados E Discussão

Compreensão da leitura: em espanhol

(língua estrangeira) e em português (língua materna)

O Quadro 1 mostra o desempenho de todos os participantes na compreensão da leitura dos textos em espanhol e em português. No que se refere à compreensão dos textos em espanhol, verifica-se que apenas no texto D os participantes tiveram média inferior a 50%. Destaca-se que esse resultado era esperado, pois tanto o texto quanto as questões referentes à sua compreensão apresentam maior nível de complexidade quando comparados aos outros três textos.

Em relação aos textos que avaliaram a compreensão da leitura em português, verificou-se situação semelhante, ou seja, os participantes obtiveram um desempenho médio de 42,54% no texto TCLP\_B. Esse resultado, provavelmente, deve-se ao fato de que embora o conteúdo do texto fosse acessível, as questões referentes à sua compreensão tinham nível mais elevado de complexidade quando comparadas às questões do TCLP\_A.

Quadro 1 Desempenho médio dos participantes nas tarefas de compreensão de leitura em espanhol e em português

	Média	d.p.
Espanhol		



TCLE_A	69,18	28,12
TCLE_B	65,12	31,10
TCLE_C	57,46	29,20
TCLE_D	31,84	18,52
Português		
TCLP_A	71,19	23,26
TCLP_B	42,54	20,55

Legenda: TCLE = Tarefa de compreensão da leitura em espanhol; TCLP = Tarefa de compreensão da leitura em português.

A fim de apresentar os resultados de forma parcimoniosa, realizou-se uma análise fatorial exploratória dos resultados das tarefas de compreensão de leitura em espanhol e português, para verificar se os dois tipos de tarefas poderiam ser agrupados em fatores gerais.

A avaliação da consistência das tarefas de compreensão da leitura em espanhol realizou-se por meio do cálculo de alfa de Cronbach, o qual revelou a possibilidade de agrupar a pontuação relativa aos textos (TCLE\_A; TCLE\_B) numa única dimensão (TCL\_Esp). Entretanto, os textos TCLE\_C e TCLE\_D participarão das análises de forma independente.

A análise fatorial das tarefas de compreensão de leitura em português revelou que as duas tarefas utilizadas poderiam ser agrupadas numa única dimensão. Assim, nas próximas análises será utilizada uma única pontuação de compreensão da leitura em português (CLT\_Por).

Destaca-se que o desempenho dos participantes nas tarefas de compreensão de leitura, tanto em espanhol quanto em português, não mostrou diferenças significativas segundo o ano escolar (8º ou 9º ano), motivo pelo qual nas análises seguintes envolvendo essas duas variáveis os participantes serão considerados como um grupo único.

O Quadro 2 mostra as correlações entre o desempenho dos participantes na compreensão da leitura em espanhol e em português (língua materna). Inicialmente, observa-se que não existe correlação significativa entre o desempenho no agrupamento CLT\_Esp e nos outros textos em espanhol (TCLE\_C; TCLE\_D), sendo que o desempenho na compreensão dos dois últimos estão positiva e significativamente relacionados.

Quadro 2 Correlação entre a pontuação na compreensão da leitura em espanhol e em português (língua materna)

		1	2	3	4
Espanhol					
1	CLT_Esp	—	.17	-.04	.33*
2	TCLE_C		—	.33*	.40*
3	TCLE_D			—	.33*
Português					
4	CLT_Por				—

Legenda: TCLE = Tarefa de compreensão de leitura em espanhol; CLT\_Esp = Compreensão leitora de textos em espanhol; CLT\_Por = Compreensão leitora de textos em português.

Nota: \* = Correlação significativa com  $p < 0,05$ .

Além disso, verificou-se uma correlação significativa e positiva entre a compreensão da leitura em português (CLT\_Por) e os indicadores de compreensão da leitura em espanhol (CLT\_Esp; TCLE\_C; TCLE\_D), ou seja, esses dados mostram a existência de uma relação entre o desempenho na compreensão da leitura em português (língua materna) e o desempenho na compreensão da leitura em espanhol (língua estrangeira). O resultado é especialmente expressivo, considerando-se o número de participantes e as características das tarefas (graus de dificuldade). No entanto, para que esse resultado possa ser generalizado, devem-se realizar investigações com um número maior de participantes.

É importante salientar, ainda, que os achados corroboram os estudos de Kato (1999) e Kleiman (2004), quando afirmam que um leitor que tem maior compreensão da leitura em língua materna também apresentará maior capacidade para compreender a língua estrangeira.

### **Relação Entre O Conhecimento De Vocabulário E A Compreensão De Textos Em Espanhol**

Os resultados do teste de vocabulário em Espanhol Peabody PPVT-III, cuja média global foi igual a 57,32 (d.p. = 8,55), não revelaram diferenças significativas entre as médias de desempenho dos dois anos escolares incluídos no estudo ( $t = 0,54$ ;  $p = 0,59$ ). O desempenho médio foi superior a 50% tanto no 8º ( $M = 56,75$ ; d.p. = 7,45) quanto no 9º ano ( $M = 57,87$ ; d.p. = 9,58), sugerindo que não houve variações significativas entre os níveis de desempenho no teste.

Foi um resultado inesperado, tendo em vista que os alunos do 9º ano tinham um ano a mais de participação no processo de ensino-

-aprendizagem de espanhol. Destaca-se que o currículo escolar prevê um conteúdo progressivo e gradual e, por isso, esperava-se que os estudantes dos dois diferentes anos (8º e 9º) diferissem nos níveis de desempenho nas tarefas em espanhol.

Com relação à tarefa de vocabulário PPVT-III, infere-se que o desempenho dos participantes não foi produto de seu real conhecimento das palavras e seu significado, mas resultado das tentativas de acerto ("chute") de qual imagem correspondia à palavra anunciada. Por exemplo, em um dos itens da prova, quando foi solicitado ao aluno que indicasse qual das quatro figuras representa melhor o significado da palavra artefato: 1) ferro; 2) quadro com um peixe; 3) carteira e 4) frasco virado com cápsulas espalhadas, a resposta dada por grande parte dos alunos foi a 2, quando a correta era a 1. A partir da pergunta da aplicadora do teste sobre o motivo da resposta apresentada, descobriu-se que os alunos relacionaram o quadro que tinha o peixe com arte, e esta, por sua vez, com artefato.

Respostas como essa parecem reforçar a conclusão de Almeida (2002), que, ao investigar as estratégias utilizadas pelos alunos para compreensão do vocabulário em língua estrangeira, identificou que eles demonstram fazer inferências utilizando tanto as pistas do contexto como os conhecimentos linguísticos da língua materna.

O Quadro 3 apresenta a correlação (Correlação de Pearson) entre a pontuação de compreensão da leitura em espanhol e do teste PPVT-

III. Surpreendentemente, a análise não revelou uma relação estatisticamente significativa entre essas variáveis.

Quadro 3 Correlação entre a pontuação nas tarefas de compreensão da leitura em espanhol e no teste de vocabulário em Espanhol PPVT-III

	PPVT-III	
	Pearson-r	p
CLT_Esp	0,17	0,18
TCLE_C	0,13	0,30
TCLE_D	0,02	0,89

Legenda: TCLE\_C e TCLE\_D = Tarefas de compreensão da leitura em espanhol; CLT\_Esp =

Compreensão da leitura de textos em espanhol; PPVT-III = Teste de Vocabulário por Imagens Peabody.

Para analisar esses resultados, é importante considerar primeiramente o teste PPVT-III para avaliar o vocabulário receptivo-auditivo.

Assim, ele não avaliou todas as dimensões relativas ao conhecimento de vocabulário em espanhol dos alunos, pois não era um teste de vocabulário expressivo. Portanto, acredita-se que a ausência de relação entre os dados obtidos pode ser explicada por limitação do teste utilizado, o que indica a necessidade de outras avaliações, com outros instrumentos, para uma avaliação mais completa e precisa do vocabulário.

Outra possível explicação para a ausência de uma relação significativa entre o desempenho no vocabulário e a compreensão de textos em espanhol é a de que, em níveis elementares de conhecimento da língua estrangeira, as estratégias utilizadas para compreensão da leitura estariam menos dependentes do conhecimento do significado da totalidade das palavras, sendo o contexto do texto e as pistas gramaticais os principais aliados na tarefa de compreensão (BOWEY, 1986; KLEIMÁN, 2004; TUNMER, 1990; TUNMER; HOOVER, 1992).

### **O Contexto Do Texto E A Aquisição De Vocabulário Em Espanhol (Língua Estrangeira)**

Para identificar a capacidade dos participantes em utilizar o contexto do texto e inferir o significado das palavras desconhecidas, avaliou-se a variação na compreensão do significado de 23 palavras, pertencentes aos textos em espanhol, antes e depois da leitura (dentre elas, dez falsos cognatos), utilizando o teste t de Student pareado, o que permitiu comparar a pontuação do grupo nos dois momentos. O resultado mostrou um aumento significativo no desempenho médio dos participantes na identificação de palavras depois de realizarem a leitura dos textos ( $t = -9,85$ ;  $p < 0,05$ ).

Ao analisar o desempenho dos alunos individualmente, verificou-se que só um participante (1,5%) teve desempenho inferior na tarefa de vocabulário realizada depois da leitura dos textos em espanhol, 11 (16,4%) tiveram o mesmo percentual de desempenhos na tarefa pré-leitura e pós-leitura (embora não necessariamente nas mesmas palavras), 55 (82,1%) apresentaram aumento no número de respostas corretas. Tal resultado reitera, mais uma vez, os argumentos de Tunmer (1990) e Tunmer e Hoover (1992), pois a leitura dos textos em espanhol possibilitou a aquisição de um vocabulário que os participantes não possuíam.

Os resultados do teste de correlação de Pearson (Quadro 4) indicam que a variação na compreensão do vocabulário (antes e depois da leitura dos textos) apresentou relação significativa e positiva com três das quatro tarefas de compreensão de leitura.

Quadro 4 Relação entre a pontuação nas tarefas de compreensão da leitura em espanhol e a variação na compreensão do vocabulário dos textos pré e pós-leitura

Variação na compreensão do vocabulário dos textos		
	Pearson-r	P
TCLE_A	0,38*	0,01
TCLE_B	0,33*	0,01
TCLE_C	0,35*	0,01
TCLE_D	0,09	0,49

Legenda: TCLE = Tarefa de compreensão da leitura em espanhol. Fonte: Dados da pesquisa.

Nota: \* = Correlação significativa com  $p < 0,05$ .

É importante salientar que a compreensão do texto TCLE\_D que não apresentou relação significativa com a variação de conhecimento do vocabulário foi também a que demonstrou o desempenho médio mais baixo (Quadro 1), o que sugere que quanto maior é a compreensão de um texto, maior a

possibilidade de inferir o significado das palavras desconhecidas encontradas nele. Considera-se que a relação positiva verificada pela análise de correlação apresentada indica que os participantes conseguiram refletir sobre o sentido global dos textos menos complexos (TCLE\_A, TCLE\_B e TCLE\_C) e inferir o significado das palavras desconhecidas, estratégia que, embora extremamente adequada, não foi suficientemente eficaz para a inferência do significado das palavras contidas no texto mais complexo (TCLE\_D). Em síntese, sugere-se que quando a compreensão de um texto exige habilidades mais sofisticadas das que as que o leitor domina (inclusive em termos lexicais) ele não consegue criar um contexto significativo que sirva de base para a inferência do significado das palavras desconhecidas.

### As Habilidades Metassintáticas E A Compreensão Da Leitura De Textos Em Espanhol

Uma análise do desempenho dos participantes nas tarefas que avaliam as habilidades metassintáticas por meio do teste t de Student não mostrou diferenças estatisticamente significativas entre as médias dos alunos nas tarefas de julgamento gramatical ( $t = 0,92$ ;  $p = 0,36$ ); correção gramatical ( $t = 0,21$ ;  $p = 0,84$ ) e categorização de palavras ( $t = -0,96$ ;  $p = 0,34$ ), motivo pelo qual as análises estatísticas serão realizadas considerando-se todos os participantes em um único grupo. Esperava-se que os estudantes do 9º ano tivessem maior facilidade para resolver essas tarefas, especialmente as de julgamento e correção gramatical, tendo em vista que elas apresentavam frases cuja estrutura sintática é diferente do português. Entretanto, como já foi dito, o desempenho médio dos alunos dos dois anos escolares (8º e 9º) não foi significativamente diferente. Sugere-se que em estudos posteriores seja adotada diferenciação maior entre os grupos, caso exista interesse em examinar diferenças no desenvolvimento das habilidades metalinguísticas, o que pode realizar-se idealmente com o uso de dados obtidos longitudinalmente.

Os níveis de desempenho dos participantes nas tarefas gramaticais (Quadro 5) revelaram habilidade média tanto na tarefa de julgamento gramatical (64,63%) como na tarefa de categorização de palavras (42,69%). Entretanto, verificou-se nível bastante básico na tarefa de correção gramatical (18,10%). Comparando-se os resultados nas tarefas de julgamento gramatical e de correção gramatical verifica-se que a tarefa de julgamento foi mais facilmente resolvida, enquanto que a tarefa de correção apresentou maior dificuldade. Esse resultado confirma o esperado, pois a tarefa de correção exige que o examinando demonstre explicitamente seu conhecimento da gramática da língua estrangeira (o que entre esses participantes era incipiente).

Quadro 5 Desempenho médio nas tarefas de avaliação das habilidades metassintáticas em espanhol

Tarefas de avaliação	Média	d.p.	Mínimo	Máximo
Julgamento gramatical	64,63	12,62	25,00	90,00
Correção gramatical	18,10	15,02	0,00	60,00
Categorização de palavras	42,69	14,07	13,33	73,33

Relacionando o resultado nas tarefas de compreensão de leitura de textos em espanhol e o desempenho nas tarefas de avaliação da consciência sintática por meio da análise da correlação de Pearson (Quadro 6), não foi possível identificar uma relação significativa entre o desempenho dos participantes em nenhuma das tarefas de compreensão da leitura em espanhol e a tarefa de categorização de palavras. A tarefa de correção gramatical apresentou relação positiva e significativa com o fator CLT\_Esp. E a tarefa de compreensão do texto TCLE\_D de espanhol apresentou correlação significativa com a tarefa de julgamento gramatical.

Quadro 6 Correlação entre a pontuação das tarefas de compreensão de leitura em espanhol e as tarefas de avaliação das habilidades metassintáticas

	(1)	(2)	(3)
	Julgamento	Correção	Categorização

Habilidades metassintáticas				
1	Julgamento gramatical	—	0,16	0,18
2	Correção gramatical		—	0,26*
3	Categorização de palavras			—
Compreensão da leitura				
4	CTL_Esp	0,11	0,24*	0,20
6	TCLE_C	0,19	0,15	0,24
7	TCLE_D	0,26*	0,07	0,22

Legenda: TCLE = Tarefa de compreensão da leitura em espanhol. Fonte: Dados da pesquisa.

Nota: \* = Correlação significativa com  $p < 0,05$ .

É importantes salientar que a ausência de relação significativa entre as variáveis de compreensão da leitura e habilidades metassintáticas pode ser decorrente do reduzido número de participantes neste estudo e, por isso, não se tenha encontrado correlação semelhante à encontrada nos trabalhos em língua portuguesa (CAPOVILLA; CAPOVILLA; SOARES, 2004; GUIMARÃES, 2003; REGO, 1995;

REGO; BRYANT, 1993). Entretanto, destaca-se a relação significativa entre o desempenho na compreensão dos textos A e B em espanhol (CTL\_Esp) e o desempenho na tarefa de correção gramatical, o que sugere que o conhecimento explicitamente manifesto nessa tarefa é relevante para a compreensão dos textos menos complexos, embora não tenha possibilitado boa compreensão do texto mais complexo (TCLE\_D). Por outro lado, a relação encontrada entre o desempenho na compreensão do texto TCLE\_D e o desempenho na tarefa de julgamento gramatical é mais difícil de ser explicada. Na realidade, como o desempenho na tarefa de julgamento gramatical pode ser produto de um conhecimento implícito, pois essa tarefa não exigia a correção das frases agramaticais, acredita-se que outras investigações seriam necessárias para explicar por que o resultado dessa tarefa relacionou-se justamente à compreensão do texto mais complexo. Aparentemente, o conhecimento avaliado na tarefa de julgamento gramatical é relevante para a compreensão de textos mais complexos (aqui representados pelo TCLE\_D) e, por isso, verificou-se relação significativa entre essas duas variáveis, ainda que o julgamento gramatical efetuado pelos participantes deste estudo não tenha sido decorrente do conhecimento explícito das regras gramaticais do espanhol (língua estrangeira).

Enfim, dadas as relações encontradas entre o desempenho nas tarefas de compreensão da leitura em espanhol e nas tarefas de avaliação da consciência sintática, infere-se que a compreensão da leitura em espanhol dos participantes – embora relacionada às habilidades sintáticas – depende também de outras habilidades cognitivas, dentre elas a capacidade de apreender o sentido do texto e a utilização dos conhecimentos linguísticos da língua materna. Entretanto, para concluir de forma mais precisa, são necessários outros estudos, com aprendizes de espanhol de níveis mais avançados, para que se possa verificar a possível relação entre as habilidades metassintáticas dos leitores e a compreensão de textos com estrutura complexa.

---



---



---



---



---



---



---

## Sinônimos e Antônimos

Os **sinônimos** e os **antônimos** designam palavras (substantivos, adjetivos, verbos, complementos, etc.), que segundo seu significado, ora se assemelham (sinônimos) e ora são opostas (antônimos).

A **semântica** é o ramo da linguística encarregada de estudar as palavras e seus significados. Para tanto, enfoca nos estudos dos seguintes conceitos: sinônimos, antônimos, parônimos e homônimos.

### Sinônimos

Do grego, o termo sinônimo (*synonymós*) é formado pelas palavras “*syn*” (com); e “*onymia*” (nome), ou seja, no modo literal significa aquele que está com o nome ou mesmo semelhante a ele. Não obstante, a sinonímia é o ramo da semântica que estuda as palavras sinônimas, ou aquelas que possuem significado ou sentido semelhante, sendo muito utilizadas nas produções dos textos, uma vez que a repetição das palavras empobrece o conteúdo.

### Tipos de Sinônimos

Embora, muito estudiosos da área advogam sobre a inexistência de palavras sinônimas (com valor semântico idêntico), posto que para eles, cada palavra possui um significado distinto; de acordo com a aproximação semântica entre as palavras sinônimas, elas são classificadas de duas maneiras:

- **Sinônimos Perfeitos:** são as palavras que compartilham significados idênticos, por exemplo: léxico e vocabulário; morrer e falecer; após e depois.
- **Sinônimos Imperfeitos:** são as palavras que compartilham significados semelhantes e não idênticos, por exemplo: feliz e alegre; cidade e município; córrego e riacho.

### Exemplos de Sinônimos

Segue abaixo alguns exemplos de palavras sinônimas:

- Adversário e antagonista
- Adversidade e problema
- Alegria e felicidade
- Alfabeto e abecedário
- Ancião e idoso
- Apresentar e expor
- Belo e bonito
- Brado e grito
- Bruxa e feiticeira
- Calmo e tranquilo
- Carinho e afeto
- Carro e automóvel
- Cão e cachorro
- Casa e lar
- Contraveneno e antídoto
- Diálogo e colóquio

- Encontrar e achar
- Enxergar e ver
- Extinguir e abolir
- Gostar e estimar
- Importante e relevante
- Longe e distante
- Moral e ética
- Oposição e antítese
- Percurso e trajeto
- Perguntar e questionar
- Saboroso e delicioso
- Transformação e metamorfose
- Translúcido e diáfano

### **Antônimos**

Do grego, o termo antônimo corresponde a união das palavras “*anti*” (algo contrário ou oposto) e “*onymia*” (nome). A antonímia é o ramo da semântica que se debruça nos estudos sobre as palavras antônimas. Do mesmo modo que os sinônimos, os antônimos são utilizados como recursos estilísticos na produção dos textos.

Exemplos de Antônimos

Segue abaixo alguns exemplos de palavras antônimas:

- Aberto e fechado
- Alto e baixo
- Amor e ódio
- Ativo e inativo
- Bendizer e maldizer
- Bem e mal
- Bom e mau
- Bonito e feio
- Certo e errado
- Doce e salgado
- Duro e mole
- Escuro e claro
- Forte e fraco

- Gordo e magro
- Grosso e fino
- Grande e pequeno
- Inadequada e adequada
- Ordem e anarquia
- Pesado e leve
- Presente e ausente
- Progredir e regredir
- Quente e frio
- Rápido e lento
- Rico e pobre
- Rir e chorar
- Sair e entrar
- Seco e molhado
- Simpático e antipático
- Soberba e humildade
- Sozinho e acompanhado

A **Semântica** é a parte da linguística que estuda o significado das palavras, a parte significativa do discurso. Cada palavra tem seu significado específico, porém podemos estabelecer relações entre os significados das palavras, assemelhando-as umas às outras ou diferenciando-as segundo seus significados.

**SINONÍMIA:** Sinonímia é a divisão na Semântica que estuda as **palavras sinônimas**, ou aquelas que possuem significado ou sentido semelhante.

Algumas palavras mantêm relação de significado entre si e representam praticamente a mesma ideia. Estas palavras são chamadas de **sinônimos**.

Ex: certo, correto, verdadeiro, exato.

Sendo assim, **SINÔNIMOS** são palavras que possuem significados semelhantes.

A contribuição greco-latina é responsável pela existência de numerosos pares de sinônimos:

- adversário e antagonista;
- translúcido e diáfano;
- semicírculo e hemiciclo;
- contraveneno e antídoto;
- moral e ética;
- colóquio e diálogo;



- transformação e metamorfose;
- oposição e antítese.

**ANTONÍMIA:** É a relação entre palavras de significado oposto

Outras palavras, ainda, possuem significados completamente divergentes, de forma que um se opõe ao outro, ou nega-lhe o significado. Estas palavras são chamadas de **antônimos**.

Ex: direita / esquerda, preto / branco, alto / baixo, gordo / magro.

Desta forma, **ANTÔNIMOS** são palavras que opõem-se no seu significado.

Observação: A antonímia pode originar-se de um prefixo de sentido oposto ou negativo:

- bendizer e maldizer;
- simpático e antipático;
- progredir e regredir;
- concórdia e discórdia;
- ativo e inativo;
- esperar e desesperar;
- comunista e anticomunista;
- simétrico e assimétrico.

Quanto à significação, as palavras são divididas nas seguintes categorias:

### **Sinônimos**

As palavras que possuem significados próximos são chamadas **sinônimos**. Veja alguns exemplos:

casa - lar - moradia – residência

longe – distante

delicioso – saboroso

carro - automóvel

Observe que os sentidos dessas palavras são **próximos**, mas não são exatamente equivalentes. Dificilmente encontraremos um sinônimo perfeito, uma palavra que signifique exatamente a mesma coisa que outra.

Há uma pequena diferença de significado entre palavras sinônimas. Veja que, embora **casa** e **lar** sejam sinônimos, ficaria estranho se falássemos a seguinte frase: Comprei um novo lar.

Obs.: o uso de palavras sinônimas pode ser de grande utilidade nos processos de retomada de elementos que inter-relacionam as partes dos textos.

### **Antônimos**

São palavras que possuem significados opostos, contrários. Exemplos:

mal / bem

ausência / presença

fraco / forte

claro / escuro

subir / descer

cheio / vazio

possível / impossível

Sinônimos e antônimos resultam das relações de proximidade e contrariedade que as palavras estabelecem umas com as outras. As relações de sinonímia e antonímia são estudadas pela semântica.

São **sinônimas** as palavras que apresentam **significados semelhantes**.

São **antônimas** as palavras que apresentam **significados opostos**.

### **Sinônimos**

Palavras sinônimas são palavras que apresentam um significado aproximado na representação de uma ideia. Embora o sentido de palavras sinônimas seja próximo, não é exatamente equivalente, sendo rara a existência de sinônimos perfeitos, ou seja, de palavras diferentes que signifiquem exatamente a mesma coisa.

Isto ocorre porque, mesmo apresentando significados equivalentes, as palavras possuem conotações diferentes. Os termos podem ser mais eruditos ou mais populares, apresentando uma carga cultural diferente, podem ser mais concretos ou mais abstratos, podem transmitir intensidades diferentes,...

Além disso, a relação de sinonímia entre duas palavras não é recíproca, ou seja, a substituição de um termo pelo outro não ocorre nos dois sentidos. É preciso ter em consideração o contexto em que se insere a palavra. Assim, a escolha de um sinônimo deve ser feita de forma contextualizada, para que não haja alteração semântica da mensagem.

O uso de sinônimos é essencial na diversificação vocabular, evitando a repetição. Na produção textual, os sinônimos, além de serem usados como um recurso estilístico, são essenciais na retomada de elementos que aparecem ao longo de todo o texto.

### **Exemplos de sinônimos**

#### **Sinônimos de importante:**

- significativo;
- considerável;
- prestigiado;
- indispensável;
- fundamental;
- ...

#### **Sinônimos de necessário:**

- essencial;
- fundamental;
- forçoso;
- obrigatório;

- imprescindível;
- ...

**Sinônimos de problema:**

- dificuldade;
- adversidade;
- contratempo;
- defeito;
- dilema;
- enigma;
- ...

**Sinônimos de conhecimento:**

- sabedoria;
- estudo;
- compreensão;
- know-how;
- convívio;
- ...

**Sinônimos de desenvolver:**

- crescer;
- progredir;
- evoluir;
- melhorar;
- aprimorar;
- expor;
- ...

**Sinônimos de realizar:**

- fazer;
- efetuar;
- executar;
- acontecer;
- suceder;
- conseguir;

- ...

**Sinônimos de mostrar:**

- expor;
- apresentar;
- manifestar;
- indicar;
- demonstrar;
- exhibir-se;
- ...

**Sinônimos de portanto:**

- logo;
- assim;
- isto posto;
- à vista disso;
- por conseguinte;
- ...

**Sinônimos de porém:**

- mas;
- contudo;
- todavia;
- falha;
- senão;
- ...

**Antônimos**

Palavras antônimas são palavras que apresentam um significado contrário na representação de uma ideia. Além de contrariedade e oposição, os antônimos podem também estabelecer correlação e complementaridade.

A antonímia é habitualmente estabelecida entre palavras diferentes, com radicais diferentes, mas os antônimos podem ser formados também por prefixos de negação, como: in-, des-, a-. Os antônimos podem ainda ser representados por palavras que já apresentam prefixos cujos significados são contraditórios.

**Antônimos com radicais diferentes:**

- bom e mau;
- bonito e feio;

- alto e baixo.

**Antônimos com prefixos de negação:**

- feliz e infeliz;
- atento e desatento;
- típico e atípico.

**Antônimos com prefixos contraditórios:**

- exteriorizar e interiorizar;
- progressão e regressão;
- ascendente e descendente.

Tal como os sinônimos, os antônimos são também utilizados como recursos estilísticos na produção textual, devendo também ser analisados em contexto.

**Exemplos de antônimos****Antônimos de dedicado:**

- desinteressado;
- desapegado;
- faltoso;
- desaplicado;
- relapso;
- ...

**Antônimos de pontual:**

- atrasado;
- retardado;
- durável;
- genérico;
- irresponsável;
- ...

**Antônimos de supérfluo:**

- necessário;
- preciso;
- útil;
- importante;
- indispensável;

- ...

**Antônimos de progredir:**

- regredir;
- retroceder;
- involuir;
- estagnar;
- permanecer;
- ...

**Antônimos de essencial:**

- desnecessário;
- supérfluo;
- inútil;
- secundário;
- acessório;
- ...

**Antônimos de provisório:**

- definitivo;
- permanente;
- duradouro;
- efetivo;
- estável;
- ...

**Antônimos de acender:**

- apagar;
- extinguir;
- desligar;
- esmorecer;
- acalmar;
- ...

**Antônimos de mal:**

- bem;
- corretamente;

- adequadamente;
- benefício;
- favor;
- ...

**Antônimos de subsequente:**

- precedente;
- antecedente;
- anterior;
- prévio;
- primeiro;
- ...

É muito importante termos o hábito de ler bons livros, histórias em quadrinhos, jornais, e outros, para que cada vez o nosso vocabulário se torne mais aperfeiçoado.

Porque como você sabe, não podemos escrever da mesma maneira como falamos, pois a escrita precisa estar de acordo com as regras gramaticais da língua.

Então, quando falamos em sinônimos, lembramos de significado. E esse significado nos leva à ideia do dicionário, porque ele deve ser nosso companheiro constante.

Qual é a palavra que tem o mesmo sentido de menino?

Podemos dizer garoto, assim como podemos também falar que caridade é o mesmo que bondade. Apenas houve a mudança de palavra, mas o significado permaneceu o mesmo.

Por isso, dizemos que “sinônimo” significa semelhança de sentido.

Observe algumas palavras:

casa – residência  
alegria – felicidade  
percurso – trajeto  
questionar – perguntar  
brincadeira – diversão  
carinho- afeto  
calmo – tranquilo

Os antônimos significam palavras contrárias, inversas de sentido. Perceba:

claro – escuro  
dia – noite  
bondade – maldade  
bonito – feio  
limpo – sujo  
correto – errado  
largo – estreito  
alto – baixo

---

---

## Divisão Silábica

Você sabe como separar as sílabas corretamente? Para fazer isso é preciso saber algumas regras da língua portuguesa. Confira!

Imagine que você está escrevendo uma redação na escola ou em algum processo avaliativo. Entre tantas palavras, uma delas não coube inteiramente na linha que você escrevia. Então, o que se deve fazer? Bom, nesses casos é indicado separar o vocábulo em duas partes, colocando um hífen (-) entre elas.

E logo vem a dúvida: onde colocar? Não se deve separá-las de qualquer maneira. É preciso, portanto, saber as regras de divisão silábica e assim conseguir escrever dentro da norma culta da língua portuguesa. Veja a seguir essas normas e aplique-as em seu cotidiano.

Dividindo as sílabas

Para realizar uma divisão correta, é preciso ter em mente, a princípio, que em todas as sílabas deve haver pelo menos uma vogal, sem exceções. Por essa razão, essa norma se torna geral. Conheça agora as regras práticas.

Não se separam

### Ditongos e tritongos

Palavras que possuem, respectivamente, duas e três vogais juntas. Na separação silábica elas pertencem a uma mesma sílaba.

**Exemplos:** cau-le, ân-sia, di-nhei-ro, trei-no, des-mai-a-do, U-ru-guai, sa-guão, Pa-ra-guai, a-ve-ri-guou, quais-quer, etc.

### Dígrafos

São encontros consonantais, isto é, duas consoantes juntas, que possuem um mesmo som. Alguns devem ser separados, mas outros não. Esse é o caso do: **ch, lh, nh, gu e qu.**

**Exemplos:** chu-va, fa-cha-da, es-ta-nho, fro-nha, a-que-la, co-lhei-ta, fi-lha, ni-nho, quei-jo, etc.

### Encontros consonantais com L e R

Quando duas consoantes estão juntas na palavra e a segunda é **l** ou **r**, não há a separação delas. Observe:

**Exemplos:** fla-gran-te, gló-ria, pla-no, cla-va, a-pre-sen-tar, a-brir, re-tra-to, re-gra, a-bran-dar, dra-gão, tra-ve, etc.

Nessa regra há uma exceção, lembrem-se dela: **ab-rup-to.**

### Encontros consonantais iniciais

Se a palavra tiver duas sílabas juntas no início, elas não são separáveis. Entenda.

**Exemplos:** gnós-ti-co, pneu-má-ti-co, mne-mô-ni-co, gno-mo, psi-có-lo-go, pneu-mo-ni-a, etc.

### Palavra terminada em consoante

Em nenhuma hipótese uma palavra que termine com consoante terá uma divisão silábica em que a consoante fique isolada no final. Nesse sentido, a última letra se une à anterior.

**Exemplos:** sub-lin-gual, su-ben-ten-der, en-xá-guam, a-guen-tar, etc.



Separam-se

### Ditongo decrescente + vogal

São palavras formadas por três vogais, mas não é a mesma coisa que o tritongo. Nessas palavras, a formação é feita com uma vogal (a, e, o) + semivogal (i,u) + uma outra vogal (a,e,o). Preste atenção!

**Exemplos:** prai-a, tei-a, joi-a, sa-bo-rei-e, es-tei-o, ar-roi-o, etc.

**OBS:** A formação do tritongo é diferente, sendo semivogal + vogal + semivogal: Paraguai (“u” e “i” são semi e “a” é vogal).

### Hiatos

Quando há um encontro de duas vogais. Diferem-se do ditongo pela forma que são pronunciadas.

**Exemplos:** sa-ú-de, Sa-a-ra, ca-o-olho, du-e-lo, etc.

### Outros dígrafos

Como já dito, dígrafo ocorre quando duas consoantes juntas forma um único som. Nos casos de: *rr, ss, sc, sc, xs, e xc* eles devem ser separados.

**Exemplos:** bar-ro, as-sun-to, guer-ra, sos-se-go, des-çam, cres-ço, etc.

### Encontros consonantais

Com exceção dos casos já citados, onde a segunda consoante é L ou R, nos outros casos a separação ocorre.

**Exemplos:** de-cep-ção, ab-do-me, sub-ma-ri-no, ap-ti-dão, con-vic-ção, as-tu-to, ap-to, cír-cu-lo, rit-mo, etc.

### Vogais idênticas

*aa, ee, ii, oo, uu* e os grupos consonantais *cc, cc*, também são separados.

**Exemplos:** Sa-a-ra, com-pre-en-do, xi-i-ta, vo-o, pa-ra-cu-u-ba; oc-ci-pi-tal, in-fec-ção, etc.

### Divisão Silábica

Como sabemos, as **sílabas** são fonemas pronunciados por meio de uma **única emissão de voz** e também que a **base** das **sílabas** da língua portuguesa são as vogais: a - e - i - o - u. Assim, todo **fonema** pronunciado em uma única emissão de voz tem, pelo menos, uma **vogal**.

É importante ressaltarmos que, em algumas palavras, os **fonemas /i/ e /u/ não são vogais**, já que aparecem apoiados a outra(s) vogal(is), formando uma só emissão de voz (uma sílaba). Essas vogais que apoiam as outras são chamadas de **semivogais**. O que diferencia as **vogais** das **semivogais** é o fato de que as últimas **não** desempenham o papel de **núcleo silábico**. A palavra “papai”, por exemplo, é formada por **duas sílabas (dissílaba)**, sendo a segunda formada por uma **vogal** (a) e por uma **semivogal** (i).

A par dessas informações, podemos afirmar que, para saber o **número de sílabas** que compõem as palavras, basta identificar **quantas vogais** há nessa palavra.

### Vejamos os exemplos:

- pipoca – pi – po – ca (emissão de três fonemas sequenciais que estão ligados a vogais);
- aparelho – a – pa – re – lho (emissão de quatro fonemas sequenciais que estão ligados a vogais);
- pernambucana – per – nam – bu – ca - na (emissão de cinco fonemas sequenciais que estão ligados a vogais).

**Classificação das palavras quanto ao número de sílabas**

- Monossílabas: palavras que possuem apenas uma sílaba: pé, flor, mão.
- Dissílabas: palavras que possuem duas sílabas: balão (ba-lão); suco (su-co); santo (san-to).
- Trissílabas: palavras que possuem três sílabas: hóspede (hós-pe-de); lareira (la-rei-ra); sapato (sa-pa-to).
- Polissílabas: palavras que possuem quatro ou mais sílabas: literatura (li-te-ra-tu-ra); amaciante (a-ma-ci-an-te); sambódromo (sam-bó-dro-mo).

**Divisão silábica**

→ Os dígrafos “ch”, “lh”, “nh”, “gu” e “qu” devem pertencer a uma **única sílaba**:

chu – va

o – lho

fe - char

que – ri – do

vo - zi – nho

→ Os **dígrafos** “rr”, “ss”, “sc”, “sç”, “xs” e “xc” devem ser separados em **sílabas diferentes**.

car – ro - ça

as – sas – si – no

cres – cer

nas – ceu

ex – ce – ção

→ **Ditongos e tritongos** devem permanecer na **mesma sílaba**.

U – ru – guai

ba – lai – o

→ Os **hiatos** devem ser separados em **duas sílabas** distintas.

di – a

ca – de – a – do

ba – ú

→ Os encontros consonantais devem ser separados, exceto aqueles cuja segunda **consoante** é “l” ou “r”.

bru – to

blu – sa

cla - ro

tra - go

→ Os **encontros consonantais** que **iniciam palavras** são mantidos **juntos** na **divisão silábica**.

pneu – má – ti – co

gno – mo

### Regras para divisão silábica

Na modalidade escrita, indicamos a divisão silábica com o hífen. Essa separação obedece às regras de silabação.

#### Não se separam:

**a) as letras com que representamos os dígrafos ch, lh e nh.**

#### Exemplos:

- ca-cha-ça
- pa-lho-ça
- ama-nhe-cer

**b) os encontros consonantais que iniciam sílaba.**

#### Exemplos:

- a-blu-ção
- a-cla-rar
- re-gra-do
- a-bran-dar
- sa-la-man-dra
- ca-tra-ca

**c) a consoante inicial seguida de outra consoante.**

#### Exemplos:

- gno-mo
- mne-mô-ni-co
- psi-có-ti-co

**d) as letras com que representamos os tritongos.**

#### Exemplos:

- a-guen-tar
- sa-guão
- Pa-ra-guai
- ar-guiu
- en-xa-guam

**Separam-se:**

**a) as letras com que representamos os dígrafos rr, ss, sc, sç e xc.**

**Exemplos:**

- car-ro
- pás-sa-ro
- des-ci-da
- cres-ça
- ex-ce-len-te

**b) as letras com que representamos os hiatos.**

**Exemplos:**

- sa-ú-de
- cru-el
- gra-ú-na
- re-cu-o
- vo-o

**c) as consoantes seguidas que pertencem a sílabas diferentes.**

**Exemplos:**

- ab-di-car
- cis-mar
- ab-dó-men
- bis-ca-te
- sub-lo-car
- as-pec-to

### OBSERVAÇÕES

a) Não separamos as vogais dos ditongos decrescentes.

**Exemplos:** or-dei-ro, ju-deu, mau.

b) As vogais dos ditongos crescentes aceitam dupla partição.

**Exemplos:** cá-rie/cá-ri-e, sá-bio/sá-bi-o.

A **separação silábica** representa um dos requisitos relacionados à linguagem escrita e, como tal, compõe uma das tantas competências que precisamos dispor, em se tratando de tal circunstância comunicativa. Assim dizendo, esse fato está submetido a regras predefinidas, e, portanto, precisa ser incorporado o quanto antes ao nosso conhecimento.

Ocupemo-nos em verificar algumas particularidades inerentes a esse fato da língua. Constatemos, pois, as elucidações dispostas a seguir:

\* As letras que formam os dígrafos “rr”, “ss”, “sc”, “sç”, “xs”, e “xc” devem permanecer em sílabas diferentes. Verifiquemos alguns casos:

**ex – ce – ção**

**des – cer**

**ter – ra**

**pás – sa – ro...**

\* Os dígrafos “ch”, “nh”, “lh”, “gu” e “qu” pertencem a uma única sílaba. Vejamos:

**guer – ra**

**ni – nho**

**chu – va**

**quei – jo...**

\* Os hiatos não devem permanecer na mesma sílaba. São exemplos:

**ca – de – a – do**

**ju – í – za**

**La – ís...**

\* Os ditongos e tritongos devem pertencer a uma única sílaba. Constatemos:

**Pa – ra – guai**

**a – ve – ri – guei**

**cai – xa**

**fei – xe**

\* Os encontros consonantais que ocorrem em sílabas internas não devem permanecer juntos, a não ser aqueles em que a segunda consoante é “l” ou “r”. Vejamos alguns exemplos:

flau – ta (permaneceram juntos, pois a segunda letra é representada pelo “l”)

pra – to (o mesmo ocorre com esse exemplo)

ap – to

ab – dô – men

cír – cu – lo...

Observações passíveis de nota:

Alguns grupos consonantais iniciam palavras, por isso não devem ser separados. Observemos alguns casos:

pneu – mo – ni – a

pneu – má – ti – co

psi – có – lo – go...

### **Sílaba e Divisão Silábica**

De forma geral, uma sílaba é um conjunto de fonemas (menores unidades sonoras que constroem uma palavra) formados por vogais e consoantes que são pronunciados num só impulso de voz. Podemos classificar uma palavra e sua sílaba de acordo com: sua **tonicidade** (se uma palavra possui sílaba tônica ou átona) ou o **número de sílabas** (quantidade de sílabas que uma palavra possui).

### Classificação quanto à tonicidade

Em uma palavra, uma sílaba sempre será pronunciada com mais força do que as outras. Essas sílabas são chamadas de **tônicas**, enquanto todas as outras de menor intensidade são chamadas de **sílabas átonas**, como podemos ver nos exemplos a seguir, onde as sílabas marcadas correspondem às tônicas as não marcadas às átonas:

#### Exemplos

an-ti-**pá**-ti-co, ve-lo-ci-**da**-de, lí-qui-do

Dependendo da posição da sílaba tônica em uma palavra, podemos classifica-las ainda em **oxítonas**, **paroxítonas** e **proparoxítonas**, como podemos ver na tabela abaixo:

Classificação	Posição da sílaba tônica	Exemplos
Oxítona	Última sílaba tônica	café, quintal, guaraná
Paroxítona	Penúltima sílaba tônica	velocidade, repórter, digno
Proparoxítona	Antepenúltima sílaba tônica	simpático, próximo, lâmina

### Classificação quanto ao número de sílabas

As palavras podem ser classificadas também quanto ao número de sílabas: palavras de uma sílaba só são **monossílabas**, duas sílabas são chamadas de **dissílabas**, três sílabas são as **trissílabas** e as palavras de quatro sílabas ou mais são chamadas de **polissílabas**.

#### Divisão silábica

A divisão silábica das palavras geralmente é baseada de acordo com a sua pronúncia, mas existem algumas particularidades, como vemos na tabela abaixo:

Ocorrência	Orientação	Exemplos
Ditongo e Tritongo	Não separar	Uruguai – U-ru- <b>guai</b> , Faixa – <b>Fai</b> -xa
Hiato	Separar	Saúde – <b>Sa-ú</b> -de
Dígrafos ss, rr, sc, sç, xc	Separar	Carrossel – <b>Car-ros-sel</b> , Cresça – <b>Cres-ça</b> , Exceção – <b>Ex-ce-ção</b> , Piscina – <b>Pis-ci</b> -na
Consoante não seguida de vogal	Deixar na sílaba da esquerda	Magnífico – <b>Mag-ní</b> -fi-co
Prefixos + vogal	Separar	Desigualdade – <b>De-si</b> -gual-da-de

#### Dígrafo

Confira o que é um dígrafo e quais são os dígrafos existentes visitando a nossa página: Dígrafo.

**Acentuação****Regras de Acentuação Gráfica**

Baseiam-se na constatação de que, em nossa língua, as palavras mais numerosas são as paroxítonas, seguidas pelas oxítonas. A maioria das paroxítonas termina em -a, -e, -o, -em, podendo ou não ser seguidas de "s". Essas paroxítonas, por serem maioria, não são acentuadas graficamente. Já as proparoxítonas, por serem pouco numerosas, são sempre acentuadas.

Proparoxítonas

Sílabas tônicas: antepenúltima

As proparoxítonas são todas acentuadas graficamente. Exemplos:

trágico, patético, árvore

Paroxítonas

Sílabas tônicas: penúltima

Acentuam-se as paroxítonas terminadas em:

l	fácil
n	pólen
r	cadáver
ps	bíceps
x	tórax
us	vírus
i, is	júri, lápis
om, ons	iândom, íons
um, uns	álbum, álbuns
ã(s), ão(s)	órfã, órfãs, órfão, órfãos
ditongo oral (seguido ou não de s)	jóquei, túneis

**Acentuação Gráfica**

O português, assim como outras línguas neolatinas, apresenta acento gráfico. Sabemos que toda palavra da Língua portuguesa de duas ou mais sílabas possui uma sílaba tônica. Observe as sílabas tônicas das palavras arte, gentil, táxi e mocotó. Você constatou que a tonicidade recai sobre a sílaba inicial em arte, a final em gentil, a inicial em táxi e a final em mocotó.

Além disso, você notou que a sílaba tônica nem sempre recebe acento gráfico. Portanto, todas as palavras com duas ou mais sílabas terão acento tônico, mas nem sempre terão acento gráfico. A tonicidade está para a oralidade (fala) assim como o acento gráfico está para a escrita (grafia). É importante aprender as regras de acentuação pois, como vimos acima, independem da fonética.

Abaixo estão descritas as regras de acentuação gráfica de forma descomplicada. Trata-se de assunto relativamente simples, basta memorizar as regras. Entendemos que o conhecimento sobre separação de sílabas é pré-requisito para melhor assimilação desse tema.

A Reforma Ortográfica veio descomplicar e simplificar a língua portuguesa notadamente nesta parte de acentuação gráfica.

- 11 Acentuam-se as palavras monossílabas tônicasterminadas em a, e, o, seguidas ou não de s.

Ex: já, fé, pés, pó, só, ás.

- 22 Acentuam-se as palavras oxítonasterminadas em a, e, o, seguidas ou não de s, em, ens. Ex: cajá, café, jacaré, cipó, também, parabéns, metrô, inglês alguém, armazém, conténs, vinténs.

Não se acentuam: as oxítonas terminadas em i e u, e em consoantes nem os infinitivos em i, seguidos dos pronomes oblíquos lo, la, los, las

Ex: ali, caqui, rubi, bambu, rebu, urubu, sutil, clamor, fi-lo, puni-la, reduzi-los, feri-las.

- 33 Acentuam-se as palavras paroxítonas exceto aquelas terminadas em a, e, o, seguidas ou não de s, em, ens, bem como prefixos paroxítonos terminados em i ou r.

Ex: dândi, júri, órfã, César, mártir, revólver, álbum, bênção, bíceps, espelho, famosa, medo, ontem, socorro, polens, hifens, pires, tela, super-homem.

Atenção: Acentuam-se as paroxítonas terminados em ditongo oral seguido ou não de s.

Ex: jóquei, superfície, água, área, aniversário, ingênuos.

- 44 Acentuam-se as palavras proparoxítonas sem exceção.

Ex: ótimo, incômoda, podíamos, abóbora, bússola, cântaro, dúvida, líquido, mérito, nórdico, política, relâmpago, têmpora.

- 55 Acentuam-se os ditongos abertos ei, oi, eu, seguidos ou não de s em palavras monossílabas e oxítonas.

Ex: carretéis, dói, herói, chapéu, anéis.

Atenção: Pela nova ortografia não se acentuam ditongos abertos ei, oi, eu, seguidos ou não de s em palavras paroxítonas.

Ex: ideia, plateia, assembleia.

- 66 Não se acentua, pela nova ortografia, palavras paroxítonas com hiato oo seguidos ou não de s.

Ex: voos, enjoo, abençoo.

- 77 Também não se acentuam as palavras paroxítonas com hiato ee.

Ex: creem, leem, veem, deem.

- 88 Acentuam-se sempre as palavras que contenham i, u: tônicas; formam hiatos; formam sílabas sozinhas ou são seguidos de s; não seguidas de nh; não precedidas de ditongo em paroxítonas; nem repetidas.

Ex: aí, balaústre, baú, egoísta, faísca, heroína, saída, saúde, viúvo, juízes, Piauí. Pela regra exposta acima, não se acentuam: rainha, xiita, ruim, juiz, feiura.

- 99 Pela nova ortografia, não se acentua com acento agudo u tônico dos grupos que, qui, gue, gui: argui, arguis, averigue, averigues, oblique, obliques, apazigues.



- 1010 Da mesma forma não se usa mais o trema: aguento, frequente, tranquilo, linguiça, aguentar, arguição, unguento, tranquilizante. Emprega-se o til para indicar a nasalização de vogais: afã, coração, devoções, maçã, relação etc.
- 1111 O acento diferencial foi excluído. Mantém-se apenas nestas quatro palavras, para distinguir uma da outra que se grafam de igual maneira:

A acentuação é um tema inerente aos postulados gramaticais que, indiscutivelmente, concebe-se como um fator relevante, em se tratando da linguagem escrita. Trata-se do fenômeno relacionado com a intensidade em que as sílabas se apresentam quando pronunciadas, podendo ser em maior ou menor grau. Quando proferidas com mais intensidade, classificam-se como tônicas, e quando soadas de maneira mais sutil, como átonas.

Ainda enfatizando acerca da importância do assunto em pauta, há outro detalhe pertinente: o fato de ter havido algumas mudanças em decorrência da implantação da Nova Reforma Ortográfica. Cabendo ressaltar, portanto, que os referidos postulados, abaixo descritos, encontram-se condizentes a esta. Para tanto, analisemos:

De acordo com a posição da sílaba tônica, as palavras classificam-se em:

Oxítonas – aquelas em que a sílaba tônica se encontra demarcada na última sílaba.

Exemplos: café, cipó, coração, armazém...

Paroxítonas – a sílaba tônica é penúltima sílaba.

Exemplos: caderno – problema – útil – automóvel...

Proparoxítonas – a sílaba tônica é a antepenúltima sílaba.

Exemplos: lâmpada – ônibus – cárcere – cônego...

Monossílabos átonos e tônicos

Os vocábulos que possuem apenas uma sílaba - ora caracterizados como monossílabos - também são proferidos de modo mais e/ou menos intenso. De modo a compreendermos como se efetiva tal ocorrência, analisemos:

Que lembrança darei ao país que me deu  
tudo o que lembro e sei, tudo quanto senti? (Carlos Drummond de Andrade)

Atendo-nos a uma análise, percebemos que os monossílabos “que”, “ao”, “me”, “o”, “e” são átonos, visto que são pronunciados tão fracamente que se apoiam na palavra subsequente. Já os monossílabos representados por “deu” e “sei” demonstram ser dotados de autonomia fonética, caracterizando-se, portanto, como tônicos.

Regras fundamentais:

Monossílabos tônicos

Graficamente, acentuam-se os monossílabos terminados em:

-a(s): chá, pá...

-e(s): pé, ré,...

-o(s): dó, nó...

Entretanto, os monossílabos tu, noz, vez, par, quis, etc., não são acentuados.

Observações passíveis de nota:

\* Os monossílabos tônicos formados por ditongos abertos -éis, -éu, -ói recebem o acento:

Exemplos: réis, véu, dói.

\* No caso dos verbos monossilábicos terminados em -ê, a terceira pessoa do plural termina em eem. Essa regra se aplica à nova ortografia, perceba:

Ele vê - Eles veem  
Ele crê – Eles creem  
Ele lê – Eles leem

Forma verbal que antes era acentuada agora é grafada sem o sinal gráfico.

\* Diferentemente ocorre com os verbos monossilábicos terminados em “-em”, haja vista que a terceira pessoa termina em “-êm”, embora acentuada. Perceba:

Ele tem – Eles têm  
Ela vem – Elas vêm

\* Oxítonas:

Acentuam-se todas as oxítonas terminadas em a, e, o, seguidas ou não de “s”.

Pará, café, carijó, armazém, parabéns...

\* Paroxítonas:

Acentuam-se todos os vocábulos terminados em:

-l: amável, fácil, útil...

-r: caráter, câncer...

-n: hífen, próton...

Observação: Quando grafadas no plural, não recebem acento: polens, hifens...

-x: látex, tórax...

-ps: fórceps, bíceps...

-ã(s): ímã, órfãs...

-ão(s): órgão, bênçãos...

-um(s): fórum, álbum...

-on(s): elétron, nêutron...

-i(s): táxi, júri...

-u(s): Vênus, ônus...

-ei(s): pônei, jôquei...

-ditongo oral(crescente ou decrescente), seguido ou não de “s”:  
história, série, água, mágoa...

Observações importantes:

a) De acordo com a nova ortografia, os ditongos terminados em –ei e –oi, não são mais acentuados. Perceba como eram antes e como agora são grafados:

Antes	Depois
Coreia	Coreia
plebeia	plebeia
ideia	ideia
Odisseia	Odisseia

Antes	Depois
jboia	jboia
asteroide	asteroide
paranoia	paranoia
heroico	heroico

Entretanto, o acento ainda permanece nas oxítonas terminadas em –éu, -ói e éis:

chapéu – herói - fiéis...

b) Não serão mais acentuados o “i” e “u” tônicos quando, depois de ditongo, formarem hiato: Note:

Antes	Depois
Saupe	Saupe
bocaiúva	bocauva
feíra	feira
bonina	bonna

No entanto, o acento permanece se a palavra for oxítona e o “i” ou “u” estiverem seguidos de “s” ou no final da palavra. Confira:

Piauí – tuiuí(s) – sauí(s)...

O mesmo acontece com o “i” e o “u” tônicos dos hiatos, não antecidos de ditongos:

saída – saúde – juíza – saúva – ruído...

\* As formas verbais que possuem o acento na raiz com o “u” tônico precedido das letras “q” e “g” e seguido de “e” ou “i” não serão mais acentuadas. Veja:

Antes	Depois
Apazigue (verbo apaziguar)	apazigue
Argui (verbo arguir)	argui

Atenção:

- Quando o verbo admitir duas pronúncias diferentes, usando “a” ou “i” tônicos, essas vogais serão acentuadas:

Exemplos:

eu águo, eles águam, eles enxáguam (a tônico); eu delínquo, eles delínquem (í tônico).  
tu apazíguas, que eles apazíguem.

- Se a tônica, na pronúncia, cair sobre o u, ele não será acentuado:

Exemplos:

Eu averiguo, eu aguo.

\* Não será mais usado o acento agudo para diferenciar determinados vocábulos, tais como:

Antes	Depois
para = preposição/ para = verbo parar	para
pela = preposição/ pela = verbo pelar	pela
pólo = substantivo/ pólo = combinação antiga e popular de “por” e “lo”	polo
pêlo = substantivo/ pelo = combinação da preposição com o artigo	pelo
pêra = substantivo/ pera = preposição referente ao português arcaico	pera

Contudo, o acento permanece para diferenciar algumas palavras, representadas por:

pôde = 3ª pessoa do pretérito perfeito do indicativo (verbo poder)

pode = 3ª pessoa do presente do indicativo (verbo poder)

pôr = verbo

por = preposição

### **Livro Didático: Seu Papel nas Aulas de Acentuação Gráfica**

Com a difusão da "Pedagogia Tecnicista" no sistema educacional brasileiro, a partir da década de 1970, o uso do livro didático sofreu alterações quanto aos conceitos e a forma como passaram a ser apresentados. Anteriormente a esta fase, os materiais didáticos - As Antologias - desempenhavam o papel de auxílio das aulas. O caráter auxiliar dos materiais didáticos, depois da década de 1960, foi praticamente extinto e substituído por um papel de destaque.

Em razão das necessidades econômicas e sociais da industrialização, o ensino deixou de ter uma preocupação essencialmente conceitual, enquanto a rapidez e a praticidade tornaram-se seu enfoque e levaram os livros didáticos a uma posição de direcionamento e orientação do trabalho escolar. O professor assumiu o "segundo plano" no processo ensino-aprendizagem e o livro passou a ocupar o "primeiro plano". Em lugar do material didático, o professor se transformou em auxiliar das atividades didáticas favorecendo a leitura e a realização de exercício dos livros didáticos cujo uso tornou-se obrigatório no sistema educacional brasileiro.

A imagem do professor foi diretamente atingida, pois ser professor deixou de significar domínio de conhecimento e passou a representar submissão às instruções do livro didático. Essa mudança provocou a dependência do professor e até dos alunos em relação ao uso do material didático. De acordo com Machado (1996), a dependência da escola em relação aos livros didáticos vem acarretando o rebaixamento da qualidade dos conteúdos ministrados na disciplina de Língua Portuguesa. Ao encontro dessa posição, os dados das avaliações oficiais (SAEB/INEP, 2002) mostram que os alunos do ensino fundamental e médio vêm apresentando defasagem crescente, cerca de dois a três anos de atraso entre a série em que se encontram e os conhecimentos que deveriam dominar, na aprendizagem de língua portuguesa. Para Batista (1997) e Travaglia (1996), o desempenho insatisfatório dos alunos pode ser explicado pela ineficiência das metodologias de ensino de Língua Portuguesa que vêm sendo utilizadas pelas escolas. Particularmente em relação ao ensino de gramática, os Parâmetros Curriculares Nacionais (BRASIL, 1998) assinalam a existência de graves lacunas teóricas e práticas.

Cezar, Romualdo e Calsa (2006) observam que o desempenho insatisfatório dos alunos é decorrente também da falta de compreensão sobre a necessidade de aprendizagem da língua portuguesa por parte dos falantes nativos do português. É comum os alunos questionarem o porquê e para quê são obrigados a frequentar esta disciplina com uma carga horária equivalente a outras, como a matemática, considerada mais importante para sua formação escolar. Para muitos, a aprendizagem formal da língua portuguesa não tem um significado concreto e útil, porque a linguagem formal é utilizada apenas no ambiente escolar (escrito) ou em situações muito especiais (palestras, apresentações, concursos, entre outros) com as quais não se identificam. Esse comportamento sugere não compreenderem a função de cada uma das variedades e modalidades linguísticas, como a oral e a escrita, tanto em seu registro coloquial como o culto ou padrão. Segundo a literatura (TRAVAGLIA, 1996; CALSA, 2002; CAGLIARI, 2002), a escola tem ensinado conceitos gramaticais incompletos, imprecisos e, às vezes, incorretos que não promovem reflexão sobre a importância dessa aprendizagem para a formação ampla e diversificada desses indivíduos em relação à língua portuguesa.

Frente às considerações sobre as defasagens existentes no processo de aprendizagem da língua portuguesa, este artigo tem por objetivo identificar os procedimentos utilizados por dois professores - um de final do primeiro ciclo e outro de início do segundo ciclo fundamental - de uma escola pública central do município de Maringá-PR, no ensino de um conteúdo de gramática. Buscou-se verificar o uso do livro didático em sala de aula no ensino de acentuação gráfica, um tema que tem gerado confusão conceitual dos alunos por envolver conceitos e procedimentos geralmente ensinados sem a necessária distinção do conceito de tonicidade. Não ensinados adequadamente, esses conteúdos além de gerar confusão conceitual favorecem a instalação de obstáculos epistemológicos que dificultam ou impedem aprendizagens posteriores.

## Uso do Livro Didático no Ensino de Gramática

Na década de 1960, como afirma Berger (1976), o sistema educacional brasileiro passou a ser fortemente atrelado ao sistema político do país. Com a ascensão dos militares foi introduzida a vertente pedagógica Tecnicista, de origem norte-americana. Esta modalidade de ensino foi ao encontro da necessidade de escolarização rápida e técnica dos trabalhadores que precisavam qualificar-se como mão-de-obra industrial.

Segundo Ghiraldelli (1991) e Munakata (1996 apud SILVA, 1998), os objetivos da Pedagogia Tecnicista foram atingidos com maior precisão por meio do uso dos livros didáticos que, nesse período, tiveram seu espaço escolar ampliado ao se tornarem obrigatórios. Em decorrência disso, em pouco tempo os professores deixaram de ser considerados a principal fonte de saber e planejamento e passaram a basear sua atuação didática nesses manuais. Com essa nova modalidade de ensino, o professor deixou de ser um educador autônomo para tornar-se um mero instrutor.

Para Soares (2001), a maior demanda de alunos no ensino fundamental e médio, a qualificação ligeira dos professores, e a redução salarial que levou muitos a buscarem métodos de ensino menos exigentes em termos de dedicação profissional acabou por provocar o uso intensivo do livro didático. Consolidou-se então uma tradição de uso do livro didático no sistema educacional brasileiro, e uma crescente dependência do professor em relação a esses manuais. A fidelidade a esses materiais, de acordo com Silva (1996, p. 12), vem provocando uma espécie de "anemia cognitiva" nos professores, pois segui-los representa alimentar e cristalizar "um conjunto de rotinas altamente prejudiciais ao processo educacional do professorado e do alunado". Essa dependência está diretamente relacionada à má qualidade da formação do professor e sua superação exige políticas educacionais que promovam a autonomia conceitual e didática desses profissionais. Para o autor, os livros didáticos devem informar, orientar e instruir o processo de ensino-aprendizagem e não impor uma forma de ensinar ao professor.

Em assentimento com o pensamento do autor, Lajolo (1996) lembra que os livros didáticos desempenham um papel fundamental na educação escolar, pois, dentre os outros elementos que compõem o processo ensino-aprendizagem, parece ser o de maior influência sobre as decisões e ações do professor. De acordo com a autora, no Brasil, a adoção do livro didático continua tendo como finalidade determinar os conteúdos e procedimentos de ensino tendo em vista as lacunas existentes na formação do professor e na organização do sistema educacional. Como consequência, para fugir do uso inadequado do livro didático, o professor deve avaliar sua qualidade e abordagem conceitual, pois nem sempre o referencial teórico corresponde aos conteúdos e exercícios presentes nesses manuais. Além disso, devem ser observadas suas incoerências, erros e conceitos incompletos.

Lajolo (1996, p. 8) lembra, contudo, que a má qualidade conceitual e técnica do livro pode se transformar em um material didático satisfatório a partir da identificação e discussão de seus erros com os alunos. Para ela "não há livro que seja à prova de professor: o pior livro pode ficar bom na sala de um bom professor e o melhor livro desanda na sala de um mau professor. Pois o melhor livro [...], é apenas um livro, instrumento auxiliar da aprendizagem". Nenhum livro didático, por melhor que seja, pode ser utilizado sem adaptações. Machado (1996) também chama a atenção para o fato de que mais importante que a qualidade do material didático é a formação do professor, pois ele precisa estar preparado para o desenvolvimento de um ensino qualificado, que inclui a análise dos livros didáticos adotados pela instituição escolar.

Em um estudo sobre os livros didáticos utilizados no sistema educacional brasileiro, Machado (1996) constatou que, além da falta de regularidade de sua atualização que tem provocado a baixa qualidade de seus conteúdos, apresentam custo demasiadamente alto para o padrão de consumo da maioria da população. O autor assinala que a melhoria da qualidade dos livros didáticos depende do estímulo dos órgãos governamentais e de uma maior qualificação dos professores. Neste caso, é imprescindível o desenvolvimento da capacidade crítica dos acadêmicos dos cursos de Pedagogia e das Licenciaturas das diversas áreas de conhecimento em relação ao papel dos livros didáticos no ensino escolar.

Para Pozo (1999), Arnay (1999) e Lacasa (1999), a fragmentação dos conceitos nos manuais didáticos transmite aos alunos uma noção de "falsa ciência", e não os introduz na "cultura científica escolar", função social específica dessa instituição. Segundo Machado (1996, p. 35), a "excessiva

subdivisão dos temas" dos livros didáticos em doses correspondentes à duração de uma hora-aula (50 min.) também corrobora para a fragmentação dos conceitos científicos a ponto de, em alguns casos, tornarem-se irreconhecíveis.

### **Tonicidade e acentuação gráfica**

A capacidade de se comunicar e se expressar por meio da fala é inerente ao ser humano e a esta capacidade dá-se o nome de linguagem. Para realizá-la, utiliza-se o sistema denominado língua. Sabe-se, pelos estudos realizados por Saussure (1990), que a língua é um fato social, é exterior ao indivíduo, convencional, pertencente a uma comunidade linguística. Ao usá-la individualmente, o falante concretiza, por exclusão, as possibilidades que ela oferece, no ato de fala. Ao se comunicar, o falante faz uso da estrutura psíquica denominada pelo estudioso de signo linguístico, que é composto de um conceito, o significado, e uma imagem acústica, o significante. Ambos ocorrem simultaneamente no ato da fala.

Os sinais físicos que se produzem na fala são os sons - os fonemas - que podem realizar-se de maneiras variadas. Para Câmara Jr. (2002, p. 118), o fonema é um "conjunto de articulações dos órgãos fonadores cujo efeito acústico estrutura formas lingüísticas e constitui numa enunciação o mínimo segmento distinto". Os fonemas são unidades abstratas mínimas, indivisíveis e distintivas da língua. São abstratas por serem os tipos ideais de sons constantes do sistema língua, as possibilidades dos falantes e não a sua concretização. São indivisíveis uma vez que não podem ser separadas em unidades menores.

Além dos aspectos segmentais da fala (linearidade dos signos linguísticos), a comunicação envolve elementos suprasegmentais: os acentos e tons da língua. Os acentos manifestam-se pela altura, intensidade e duração de um vocábulo, consideradas suas propriedades acústicas. Os tons estão relacionados à altura do som. Apesar da língua portuguesa não usar os tons como elementos diferenciadores do léxico, em alguns casos os aspectos suprasegmentais são importantes para a distinção e significação de um vocábulo.

Em língua portuguesa, a tonicidade está vinculada às suas origens greco-latinas. A língua latina teve um enriquecimento gramatical ao entrar em contato com o alfabeto e as regras gramaticais gregas. Contudo, não incorporou os acentos gráficos gregos como marca de tonicidade. A gramática latina marca a acentuação das palavras pela intensidade da sílaba entre breve e longa. Em latim não há palavras oxítonas, portanto, todos os dissílabos são paroxítonos. A sílaba tônica é sempre a penúltima ou antepenúltima. De acordo com Câmara Jr. (2002), os latinos não seguiram os moldes de acentuação gráfica grega em razão de, em língua latina, suas regras serem demasiadamente simples. As línguas modernas de origem latina seguem, basicamente, as regras e nomenclaturas herdadas pelos romanos dos gregos. Portanto, ao se estudar tais línguas, são encontrados termos já usados pelos gregos, como acento agudo, acento circunflexo, prosódia, entre outros.

A definição de sílaba tem sido um dos problemas encontrados nos estudos fonéticos. Há, entre os estudiosos, diversidade de critérios para a análise silábica. Drucksilbe (apud CÂMARA JR., 1970) define sílaba como sendo a emissão do ar por impulso, em que cada um corresponde a uma sílaba, dinâmica ou expiratória. Um segundo critério é o da energia de emissão que corresponde a maior energia de emissão, ou acento silábico, durante a articulação de uma sílaba. Por fim, Brücke (apud CÂMARA JR., 1970, p. 70) conceitua sílaba a partir de seu efeito auditivo, isto é, pela variação da perceptibilidade em uma enunciação contínua. Denomina a sílaba de sonora por observar "que a enunciação, sob o aspecto acústico, se decompõe espontaneamente em segmentos, ou sílabas, assinalados por um ponto máximo de perceptibilidade [...]".

Independente do critério utilizado, a conceituação de sílaba sempre envolve o ápice silábico que, pelos apontamentos de Borba (1975, p. 52), corresponde à tensão máxima a que se chega ao pronunciá-la. Para o autor, a sílaba se compõe de "uma tensão crescente e uma tensão decrescente. A primeira parte da sílaba é crescente até chegar à tensão máxima [...], a partir da qual começa a tensão decrescente". O ápice silábico, normalmente, é uma vogal. Câmara Jr. (2002) destaca que a vogal sempre é o ponto de maior tensão da sílaba. No caso dos ditongos haverá sempre uma vogal como ápice, sendo a outra denominada semivogal.

Quando formados por mais de uma sílaba, os vocábulos sempre têm uma delas pronunciada de forma mais intensa, contraponto à sílaba átona, que é pronunciada de forma mais branda. Identificar

a sílaba tônica dos vocábulos formais é uma das grandes dificuldades encontradas no processo de aprendizagem escolar, em especial, na fase de alfabetização.

Para Cagliari (2002), esse problema surge principalmente pelo fato de a escola não apresentar a tonicidade das palavras como uma ocorrência da pronúncia e não da escrita. A tonicidade é identificada nas palavras somente quando alguém busca verificar a posição em que se encontra a sílaba tônica. O autor assinala que, durante o processo de alfabetização, a escola não deve abordar a diferenciação das sílabas átonas e tônicas a partir de seu conceito. Ele acredita que elas devem ser estudadas em conjunto com a tomada de consciência dos alunos sobre o ritmo da fala.

### **Desenvolvimento da pesquisa**

O presente artigo teve por objetivo investigar os procedimentos utilizados pelos professores e livros didáticos de língua portuguesa no ensino de gramática do ensino fundamental, em particular, em relação ao conteúdo de acentuação gráfica e tonicidade. A amostra da pesquisa foi constituída por dois professores do ensino fundamental - um de 4.<sup>a</sup> e um de 5.<sup>a</sup> série de uma escola pública de Maringá-PR - selecionados a partir de seu aceite em participar da pesquisa.

Tomando como referência Lüdke e André (1986), para atingir os objetivos da pesquisa, optou-se por uma abordagem qualitativa dos dados considerada a mais adequada para a compreensão da dinâmica presente no ambiente escolar. Os dados foram coletados por meio de dois instrumentos: observações de aulas de gramática e análises de livros didáticos. Foram observadas as aulas que abordaram o tema tonicidade e acentuação gráfica, critério que definiu a quantidade de horas de observação em cada série (4.<sup>a</sup> série quatro horas e meia e 5.<sup>a</sup> série, duas horas). As observações contemplaram o desenvolvimento das atividades: apresentação do conteúdo, exercícios, uso do livro didático e outros materiais, avaliação do conteúdo. Os livros didáticos foram analisados quanto aos procedimentos subjacentes à apresentação e exercício do conteúdo.

### **Apresentação e discussão dos resultados**

Para a análise, foi utilizado o livro da coleção A Escola é Nossa, de Márcia Paganini Cavéquia (2004) - 4.<sup>a</sup> série. O volume é composto por sete unidades subdivididas em oito tópicos entre eles Pensando sobre a língua e Caderno de Ortografia, únicos em que são encontrados os conteúdos investigados - acentuação gráfica e tonicidade.

Em relação à segunda etapa do ensino fundamental foi analisado o livro de 5.<sup>a</sup> série da coleção Ler, entender e criar, de Maria das Graças Vieira e Regina Figueiredo (2004). Nesta coleção cada volume é composto por dez unidades subdivididas em sete tópicos. Os conteúdos de acentuação gráfica e tonicidade estão presentes no tópico Veja como se escreve.

O livro didático da 4.<sup>a</sup> série apresenta o conceito de sílaba tônica, classificação das palavras e regras de acentuação somente no Caderno de atividades de acentuação e ortografia, parte do Caderno de Ortografia. As explicações e os exercícios propostos apresentam os dois conteúdos de forma desvinculada. Para introduzir o conceito de sílaba tônica, o livro solicita que o aluno pronuncie várias vezes a palavra menina e indique a sílaba mais forte. Logo após, apresenta o conceito gramatical e exemplifica a classificação das palavras, conforme a posição da sílaba mais forte: oxítonas, paroxítonas e proparoxítonas.

Ao explicar a acentuação gráfica de palavras oxítonas, apresenta várias palavras como amor, cipó, calor, funil e José, com o intuito de demonstrar que nem todas essas palavras são acentuadas e que para fazê-lo corretamente deve-se observar sua terminação. O exercício denominado Complete solicitado para treinar esses conteúdos parece induzir os alunos à observação da terminação de cada vocábulo descartando a identificação da sílaba tônica.

Em outro exercício, é solicitado ao aluno que justifique o porquê da presença ou ausência do acento gráfico em um conjunto de palavras oxítonas. Segundo as orientações fornecidas ao professor, são consideradas corretas somente as respostas que explicam a acentuação a partir de regras de acentuação. Esse tipo de abordagem faz com que os alunos tomem como verdade a ideia de que o acento gráfico aparece somente em vocábulos nos quais tem uma sílaba mais forte e, assim, deixa de dar a ênfase necessária ao fato de que o acento solicitado é o gráfico. Com esse procedimento, não fica claro para os alunos que independentemente de sua grafia toda palavra possui uma sílaba tônica, com exceção dos monossílabos átonos.

Com relação à acentuação das palavras paroxítonas e proparoxítonas, o livro apresenta somente um quadro com palavras deste tipo acentuadas graficamente. Sobre esse tema são apresentados dois exercícios: o primeiro solicita a acentuação gráfica de vocábulos e sua transcrição no caderno por ordem alfabética; o segundo solicita a busca de palavras paroxítonas e proparoxítonas em jornais e revistas. Somente dois exercícios do livro sugerem a relação entre os conceitos de tonicidade e acentuação gráfica. Nesses exercícios, é solicitado aos alunos que indiquem ou pintem a sílaba tônica e, por meio das tentativas auditivas exigidas, é favorecida a percepção dos alunos quanto a tonicidade e sua relação com a acentuação gráfica (Figura 1)

FIGURA 1 - EXERCÍCIOS DE IDENTIFICAÇÃO DA SÍLABA TÔNICA DO LIVRO DIDÁTICO DE 4.ª SÉRIE

O livro didático da 5.ª série aborda os conteúdos tonicidade e regras de acentuação gráfica no tópico Veja como se escreve. Nas unidades anteriores, o direcionamento gramatical vinculou-se diretamente à escrita de determinados vocábulos envolvendo aspectos relativos aos dígrafos. Nesta unidade, quando apresentadas, as questões de acentuação são relacionadas à separação silábica dos vocábulos. Para a realização do exercício, é necessário que os alunos retornem ao tópico Outra leitura, pois a tarefa refere-se a um texto contido neste item no qual é solicitado que sejam grifadas as sílabas mais fortes das duas palavras que compõem o seu título: *Atrás do gato*. Nessa atividade, é desconsiderado o monossílabo "do" por meio do qual poderiam ser resgatados os conceitos estudados anteriormente integrando-os à atividade presente.

Depois do primeiro exercício, o livro apresenta a diferença entre sílabas tônicas e átonas, bem como a classificação das palavras conforme a posição da sílaba tônica. Apresenta como exemplos, vocábulos com e sem acento gráfico, *Bidu*, *gato* e *amigo*. Tais exemplos podem ser considerados importantes para o aprendizado, em favor da independência existente entre sílaba tônica e acento gráfico. Isto facilita a percepção do aluno sobre as convenções da língua portuguesa, como o caso dos acordos ortográficos.

Para a introdução da acentuação gráfica de palavras oxítonas são apresentados dezesseis vocábulos com e sem acento gráfico, dos quais se solicita leitura em voz alta para identificação auditiva quanto a sua sílaba tônica. Depois desta etapa, os alunos devem identificar a sílaba tônica e sua classificação. O último exercício relaciona a acentuação gráfica à terminação dos vocábulos oxítonos com o objetivo de que os alunos associem esses dois elementos e elaborem uma regra gramatical apresentada em um quadro logo abaixo.

Depois de apresentadas as regras ortográficas, solicita-se que os alunos encontrem cinco palavras oxítonas que recebam acento gráfico e, logo em seguida, elaborem frases. A elaboração de frases permite aos alunos a percepção de que o vocábulo permanece com acento gráfico independente da localização sonora que ele assume em uma frase. No último exercício é solicitada a busca em jornais e revistas dos vocábulos ensinados, reproduzindo os exercícios apresentados nos livros didáticos do primeiro ciclo.

Os vocábulos paroxítonos são abordados na sétima unidade do livro, os vocábulos oxítonos, sexta unidade e proparoxítonos na oitava unidade. Essa fragmentação de conteúdos afins, segundo a literatura, não permite que os alunos percebam as relações existentes entre os temas. Além disso, nos três casos, a classificação é apresentada no item Veja como se escreve, embora o tema relacionado à sílaba tônica se refira a um aspecto próprio da oralidade, enquanto a acentuação gráfica trata de um aspecto da língua escrita. Neste exercício novamente é solicitada a separação de sílabas antes da classificação dos vocábulos. A única mudança em relação às atividades propostas para as palavras oxítonas é tão somente a posição das sílabas tônicas. Em outro exercício é solicitada a decisão do aluno sobre a necessidade ou não de acentuação gráfica estabelecendo uma relação direta entre tonicidade e acento gráfico.



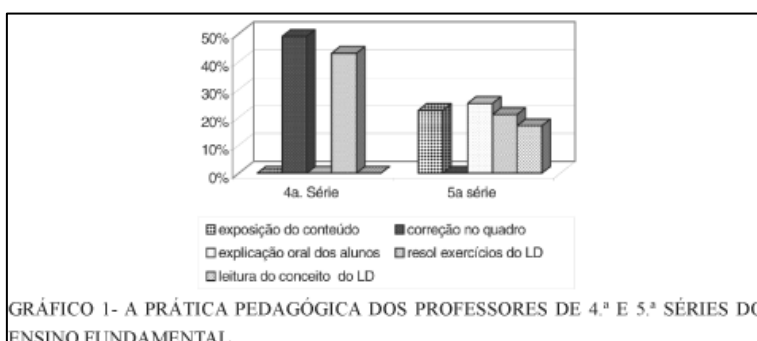
Quanto aos vocábulos proparoxítonos sua apresentação ocorre, como nas outras unidades, no tópico Veja como se escreve da oitava unidade do livro. A classificação é abordada por meio de três exercícios estruturalmente iguais: em um deles é apresentada a regra gramatical de acentuação das palavras proparoxítonas sem justificar o porquê desta norma; no último exercício sobre classificação e acentuação gráfica é sugerida uma atividade em grupo para a revisão do conteúdo gramatical das unidades anteriores. Seu foco são os vocábulos acentuados graficamente e desconsidera as palavras que não possuem acento gráfico, embora sejam submetidas às mesmas regras.

A comparação entre os dois livros didáticos mostra que no de 4.<sup>a</sup> série o conteúdo é apresentado de forma integrada e o de 5.<sup>a</sup> série tende a sua fragmentação. No primeiro manual, primeiramente, é abordado o conceito de sílaba tônica e, posteriormente, são apresentadas as regras de acentuação gráfica para a resolução dos exercícios. Este tipo de procedimento parece ser mais adequado ao desenvolvimento do tema, pois leva o aluno a compreender que quase todos os vocábulos possuem uma sílaba tônica e que somente alguns são grafados devido à vigência ortográfica da norma. O livro direcionado à segunda etapa do ensino fundamental aborda o conteúdo de acentuação em unidades distintas, revisadas em conjunto somente no tópico final. Nessas situações são priorizados os vocábulos acentuados graficamente e a estrutura dos exercícios mantém-se relacionada à classificação das palavras quanto à sua tonicidade.

Apesar das diferenças, o modo como os dois livros didáticos apresentam o conteúdo sobre tonicidade e regras de acentuação favorece o estabelecimento de confusão conceitual por parte de alunos e professores, pois não mostra que a sílaba tônica é um aspecto presente na fala e as regras de acentuação na escrita. Marcando a importância dessa distinção, assinala que não diferenciar esses dois aspectos limita o processo de instrumentalização linguística dos alunos.

#### Quanto às observações de aulas

Comparando os dados das observações com as propostas dos Parâmetros Curriculares Nacionais (BRASIL, 1997) sobre o ensino de ortografia, pode-se afirmar que a professora de 4.<sup>a</sup> série manifesta uma postura pedagógica distanciada desses documentos e similar aos pressupostos teórico-metodológicos da Pedagogia Tecnicista, cujo foco é o livro didático. Do tempo total da aula, 43% (115 min.) foram dedicados à resolução de exercícios do livro didático e 49% (130 min.) à correção desses exercícios no quadro de giz. Além disso, a professora de 4.<sup>a</sup> série não fez uso do tempo das aulas observadas para expor e explicar oralmente o conteúdo gramatical (Gráfico 1).



Nas aulas de 5.<sup>a</sup> série para a exposição oral do conteúdo sem o livro didático, o professor fez uso de 23% (27 min.) do tempo de aula, 25% (30 min.) para retomada oral deste tema por parte dos alunos, 20% (25 min.) para a resolução de exercícios dos livros didáticos e 17% (20 min.) para retomada do conteúdo por meio do livro didático. Nas aulas observadas, em média de 13% (13 min.) do tempo da aula foram usados para recados, brincadeiras, enquanto a cópia de exercícios do quadro de giz, 2% (3 min). Este professor não corrigiu exercícios no quadro de giz, utilizando-se de outros recursos para o ensino do conteúdo em foco.

As observações de aula mostraram que os dois professores investigados - a professora da 4.<sup>a</sup> série e o professor da 5.<sup>a</sup> série - utilizaram como recurso básico de ensino o livro didático. A conduta dos entrevistados mostra-se consistente com as considerações de Silva (1996, p. 13), segundo as quais o desempenho insatisfatório dos alunos pode estar vinculado ao uso do livro didático no direcionamento da atuação pedagógica dos professores. Para o autor, esse comportamento pode levar os professores a uma "anemia cognitiva" e ao rebaixamento da qualidade de seu trabalho.

Além dos prejuízos causados pelo uso quase exclusivo do livro didático, é importante ressaltar que o pouco tempo de exposição do conteúdo para os alunos, como constatado nas observações realizadas na turma de 4.<sup>a</sup> série, favorece uma aprendizagem insatisfatória dos conteúdos. Segundo Dorneles (1987), a redução do tempo de aula para a realização desse tipo de atividade é considerada um dos mecanismos seletivos da escola. Isto significa que aos sujeitos que têm menos condições de saber ou aprender o conteúdo escolar em outras situações são privadas as oportunidades necessárias à aprendizagem na instituição designada socialmente para tanto. Em outros termos, pode-se dizer que a escola não está cumprindo seu papel de transmissor do saber escolar científico a todos os cidadãos de forma equitativa.

Em contrapartida, o professor de 5.<sup>a</sup> série parece ter mantido certa coerência na distribuição do tempo de desenvolvimento das quatro categorias de atividades - exposição oral, resolução de exercícios do livro didático, resolução de exercícios no quadro e leitura do conceito gramatical que o livro didático apresenta (Gráfico 1). Observa-se que nenhum dos dois professores apresentou a acentuação gráfica como uma norma convencionada pelo conjunto social. Segundo Morais (2002), se abordado desta maneira, os alunos poderiam compreender que certos conteúdos são apenas convenções temporárias e arbitrárias que precisam ser memorizadas e conscientizadas para aquisição de uma melhor competência na linguagem oral, leitura e escrita.

Estudos anteriores como os de Cagliari (1986) e Morais (2002) enfatizam que é na 4.<sup>a</sup> e 5.<sup>a</sup> séries do ensino fundamental o momento mais apropriado para a abordagem do conceito de sílaba tônica e acentuação gráfica, pois às séries seguintes restaria o encargo de retomar esse conteúdo apenas quando necessário, dedicando-se ao desenvolvimento de outros conceitos gramaticais.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## Tipos de Frases

Existem cinco tipos de frases que são: exclamativas, declarativas, imperativas, interrogativas e optativas. Saiba mais detalhes e veja exemplos.

Chamamos de frase toda sentença com sentido completo, podendo ser constituída por uma ou mais palavras, com ou sem a presença de verbos. Por meio da frase, seja ela dita ou escrita, expressamos emoções, ideias, ordens, apelos e qualquer outro conteúdo cujo propósito seja o de comunicar alguma coisa a alguém. Existem cinco tipos de frases, caracterizados com base em suas particularidades.

### Frases exclamativas

Frases exclamativas são aquelas cujo objetivo é expressar algum tipo de emoção de forma intensa. Quando o emissor produz esse tipo de frase, ele está revelando determinado sentimento (urgência, medo, preocupação, pressa etc.) e, como sugere o próprio nome, essa revelação de emoções é seguida por um ponto de exclamação. Veja alguns exemplos:

Nossa, que dia difícil!

Que Lua maravilhosa!

Não saia de casa hoje!

Ela é a moça mais linda do mundo!

Que medo!

### Frases declarativas

As frases declarativas são aquelas que usamos para afirmar alguma coisa. Elas podem ser afirmativas ou negativas e levam um ponto final. Veja:

#### Frases declarativas afirmativas:

Ele estudou para a prova.

Jorge aprendeu a fazer churrasco.

No próximo feriado, iremos para o Rio de Janeiro.

#### Frases declarativas negativas:

Ele não estudou para a prova.

Jorge não aprendeu a fazer churrasco.

Não viajaremos no próximo feriado.

### Frases imperativas

Esse tipo de frase é utilizado quando o emissor precisa dar alguma ordem, fazer pedidos ou pedir conselhos a alguém. Podem ser finalizadas com ponto final ou de exclamação e, como no exemplo anterior, também se dividem em afirmativas ou negativas. Confira alguns exemplos:

#### Frases imperativas afirmativas

Esqueça!

Vire à esquerda.

Siga o táxi!

#### Frases imperativas negativas

Não esqueça!

Não vire à esquerda.

Não siga o táxi!

Frases interrogativas

Nesse tipo de frase, o emissor tem a intenção de fazer uma pergunta. A frase poderá ser terminada com um ponto final, no caso de um questionamento indireto, ou por um ponto de interrogação, quando a pergunta for direta. Veja a diferença:

Por favor, o médico já chegou?

Por favor, estou ligando para saber se o médico já chegou.

Você sabe onde está a caneta?

Eu só quero saber se você viu onde está a caneta.

Que horas você vem?

Eu gostaria de saber que horas você vem.

Frases optativas

Nesse tipo de frase, a pontuação final também é uma exclamação, e seu intuito é o de expressar algum tipo de desejo. Confira:

Que os anjos te protejam!

Espero que a cirurgia ocorra bem!

Muito sucesso no novo empreendimento!

Que vontade de comer brigadeiro!

Frase nominal e frase verbal

Outra classificação possível para as frases, além dos tipos citados acima, é dividi-las em verbais ou nominais. As frases verbais são aquelas que são compostas também por verbos; e as frases nominais são aquelas que não têm verbos em sua formação. Vejamos as diferenças entre esses dois tipos:

Frase nominal:

Cuidado!

Coisa esquisita...

Que triste!

Frase verbal:

Venham todos para a abertura da casa.

Filhos são uma grande responsabilidade.

O tombo foi muito feio.

---

---

---

**Pontuação:**

Os sinais de pontuação são recursos de linguagem empregados na língua escrita edesempenham a função de demarcadores de unidades e de sinalizadores de limitesde estruturas sintáticas nos textos escritos. Assim, os sinais de pontuação cumprem o papel dos recursos prosódicos, utilizados na fala para darmos ritmo, entoação e pausas e indicarmos os limites sintáticos e unidades de sentido.

Como na fala temos o contato direto com nossos interlocutores, contamos também com nossos gestos para tentar deixar claro aquilo que queremos dizer. Na escrita, porém, são os sinais de pontuação que garantem a coesão e a coerência interna dos textos, bem como os efeitos de sentidos dos enunciados.

Vejamos, a seguir, quais são os sinais de pontuação que nos auxiliam nos processos de escrita:

**Ponto ( . )**

Indicar o final de uma **frase declarativa**:

Gosto de sorvete de goiaba.

b) Separar **períodos**:

Fica mais um tempo. Ainda é cedo.

c) **Abreviar** palavras:

Av. (Avenida)

V. Ex.<sup>a</sup> (Vossa Excelência)

p. (página)

Dr. (doutor)

**Dois-pontos ( : )**

Iniciar fala de **personagens**:

O aluno respondeu:

– Parta agora!

b) Antes de **apostos** ou **orações apositivas, enumerações** ou **sequência de palavras** que explicam e/ou resumem ideias anteriores.

Esse é o problema dos caixas eletrônicos: não tem ninguém para auxiliar os mais idosos.

Anote o número do protocolo: 4254654258.

c) Antes de **citação direta**:

Como já dizia Vinícius de Moraes: “Que o amor não seja eterno posto que é chama, mas que seja infinito enquanto dure.”

**Reticências ( ... )**

Indicar **dúvidas** ou **hesitação**:

Sabe... andei pensando em uma coisa... mas não é nada demais.

b) **Interromper** uma frase incompleta sintaticamente:

Quem sabe se tentar mais tarde...

c) **Concluir** uma frase gramaticalmente incompleta com a intenção de estender a reflexão:

“Sua tez, alva e pura como um foco de algodão, tingia-se nas faces duns longes cor-de-rosa...” (Cecília - José de Alencar)

d) **Suprimir** palavras em uma transcrição:

“Quando penso em você (...) menos a felicidade.” (Canteiros - Raimundo Fagner)

### Parênteses ( )

**Isolar** palavras, frases intercaladas de caráter **explicativo, datas** e também podem **substituir a vírgula ou o travessão**:

Manuel Bandeira não pôde comparecer à Semana de Arte Moderna (1922).

"Uma manhã lá no Cajapió (Joca lembrava-se como se fora na véspera), acordara depois duma grande tormenta no fim do verão." (O milagre das chuvas no Nordeste- Graça Aranha)

### Ponto de Exclamação ( ! )

Após **vocativo**

Ana, boa tarde!

b) Final de **frases imperativas**:

Cale-se!

c) Após **interjeição**:

Ufa! Que alívio!

d) Após palavras ou frases de caráter **emotivo, expressivo**:

Que pena!

### Ponto de Interrogação ( ? )

Em **perguntas diretas**:

Quantos anos você tem?

b) Às vezes, aparece com o **ponto de exclamação** para **ênfaticamente** o enunciado:

Não brinca, é sério?!

### Vírgula ( , )

De todos os sinais de pontuação, a **vírgula** é aquele que desempenha o maior número de funções.

Ela é utilizada para **marcar uma pausa do enunciado** e tem a finalidade de nos **indicar** que os termos por ela separados, apesar de participarem da mesma **frase** ou **oração**, não formam uma **unidade sintática**. Por outro lado, quando há uma **relação sintática entre termos da oração**, não se pode separá-los por meio de vírgula.

Antes de explicarmos quais são os casos em que devemos utilizar a vírgula, vamos explicar primeiro os casos em que **NÃO** devemos usar a vírgula para separar os seguintes termos:

Sujeito de Predicado;

Objeto de Verbo;

Adjunto adnominal de nome;

Complemento nominal de nome;

Predicativo do objeto do objeto;

Oração principal da Subordinada substantiva (desde que esta não seja apositiva nem apareça na ordem inversa).

### **Casos em que devemos utilizar a vírgula:**

#### **A vírgula no interior da oração**

Utilizada com o objetivo de separar o **vocativo**:

Ana, traga os relatórios.

O tempo, meus amigos, é o que nos confortará.

b) Utilizada com o objetivo de separar **apostos**:

Valdirene, minha prima de Natal, ligou para mim ontem.

Caio, o aluno do terceiro ano B, faltou à aula.

c) Utilizada com o objetivo de separar o **adjunto adverbial** antecipado ou intercalado:

Quando chegar do trabalho, procurarei por você.

Os políticos, muitas vezes, são mentirosos.

d) Utilizada com o objetivo de separar elementos de uma **enumeração**:

Estamos contratando assistentes, analistas, estagiários.

Traga picolé de uva, groselha, morango, coco.

e) Utilizada com o objetivo de **isolar expressões explicativas**:

Quero o meu suco com gelo e açúcar, ou melhor, somente gelo.

f) Utilizada com o objetivo de **separar conjunções intercaladas**:

Não explicaram, porém, o porquê de tantas faltas.

g) Utilizada com o objetivo de **separar o complemento pleonástico** antecipado:

A ele, nada mais abala.

h) Utilizada com o objetivo de **isolar o nome do lugar na indicação de datas**:

Goiânia, 01 de novembro de 2016.

Utilizada com o objetivo de **separar termos coordenados assindéticos**:

É pau, é pedra, é o fim do caminho.

Utilizada com o objetivo de **marcar a omissão de um termo**:

Ele gosta de fazer academia, e eu, de comer. (omissão do verbo gostar)

### **Casos em que se usa a vírgula antes da conjunção e:**

Utilizamos a vírgula quando as **orações coordenadas** possuem **sujeitos** diferentes:

Os banqueiros estão cada vez mais ricos, e o povo, cada vez mais pobre.

2) Utilizamos a vírgula quando a **conjunção “e”** repete-se com o objetivo de **ênfatar** alguma ideia (polissíndeto):

E eu canto, e eu danço, e bebo, e me jogo nos blocos de carnaval.

3) Utilizamos a vírgula **quando a conjunção “e” assume valores distintos que não retratam sentido de adição (adversidade, consequência, por exemplo):**

Chorou muito, e ainda não conseguiu superar a distância.

### **A vírgula entre orações**

A vírgula é utilizada entre orações nas seguintes situações:

Para separar as **orações subordinadas adjetivas explicativas:**

Meu filho, de quem só guardo boas lembranças, deixou-nos em fevereiro de 2000.

b) Para separar as **orações coordenadas sindéticas e assindéticas**, com exceção das orações iniciadas pela conjunção “e”:

Cheguei em casa, tomei um banho, fiz um sanduíche e fui direto ao supermercado.

Estudei muito, mas não consegui ser aprovada.

c) Para separar **orações subordinadas adverbiais** (desenvolvidas ou reduzidas), principalmente se estiverem antepostas à **oração principal:**

"No momento em que o tigre se lançava, curvou-se ainda mais; e fugindo com o corpo apresentou o gancho." (**O selvagem** - José de Alencar)

d) Para separar as **orações intercaladas:**

"– Senhor, disse o velho, tenho grandes contentamentos em estar plantando-a..."

e) Para separar as **orações substantivas** antepostas à principal:

Quando sai o resultado, ainda não sei.

### **Ponto e vírgula ( ; )**

Utilizamos **ponto e vírgula** para **separar os itens** de uma sequência de outros itens:

Antes de iniciar a escrita de um texto, o autor deve fazer-se as seguintes perguntas:

O que dizer;

A quem dizer;

Como dizer;

Por que dizer;

Quais objetivos pretendo alcançar com este texto?

Utilizamos **ponto e vírgula** para **separar orações coordenadas** muito extensas ou **orações coordenadas** nas quais **já se tenha utilizado a vírgula:**

"O rosto de tez amarelenta e feições inexpressivas, numa quietude apática, era pronunciadamente vultuoso, o que mais se acentuava no fim da vida, quando a bronquite crônica de que sofria desde moço se foi transformando em opressora asma cardíaca; os lábios grossos, o inferior um tanto tenso."

(**O Visconde de Inhomirim** - Visconde de Taunay)



**Travessão ( — )**

Utilizamos o **travessão** para iniciar a **fala de um personagem** no **discurso direto**:

A mãe perguntou ao filho:

— Já lavou o rosto e escovou os dentes?

b) Utilizamos o **travessão** para indicar **mudança do interlocutor** nos **diálogos**:

— Filho, você já fez a sua lição de casa?

— Não se preocupe, mãe, já está tudo pronto.

c) Utilizamos o **travessão** para **unir** grupos de palavras que indicam **itinerários**:

Disseram-me que não existe mais asfalto na rodovia Belém—Brasília.

d) Utilizamos o **travessão** também para substituir a vírgula em expressões ou **frases explicativas**:

Pelé — o rei do futebol — anunciou sua aposentadoria.

**Aspas ( “ ” )**

As aspas são utilizadas com as seguintes finalidades:

Isolar palavras ou expressões que fogem à norma culta, como gírias, estrangeirismos, palavrões, neologismos, arcaísmos e expressões populares:

A aula do professor foi “irada”.

Ele me pediu um “feedback” da resposta do cliente.

b) Indicar uma **citação direta**:

“Ia viajar! Viajei. Trinta e quatro vezes, às pressas, bufando, com todo o sangue na face, desfiz e refiz a mala”. (**O prazer de viajar** - Eça de Queirós)

**Fique Atento!**

Caso haja necessidade de destacar um termo que já está inserido em uma sentença destacada por aspas, esse termo deve ser destacado com marcação simples ('), não dupla (").

**Veja Agora Algumas Observações Relevantes:**

Dispensam o uso da vírgula os termos coordenados ligados pelas conjunções e, ou, nem.

**Observe:**

Preferiram os sorvetes de creme, uva e morango.

Não gosto nem desgosto.

Não sei se prefiro Minas Gerais ou Goiás.

Caso os termos coordenados ligados pelas conjunções e, ou, nem aparecerem repetidos, com a finalidade de enfatizar a expressão, o uso da vírgula é, nesse caso, obrigatório.

**Observe:**

Não gosto nem do pai, nem do filho, nem do cachorro, nem do gato dele.

# LEGISLAÇÃO

*“O sucesso é a soma de  
pequenos esforços repetidos  
dia após dia”*

*Robert Collier*

# LEI COMPLEMENTAR Nº 13, DE 03 DE JANEIRO DE 1994<sup>1</sup>

Dispõe sobre o Estatuto dos Servidores Públicos Civis do Estado do Piauí, das autarquias e das fundações públicas estaduais e dá outras providências.

O Governador do Estado do Piauí,  
Faço saber que o Poder Legislativo decreta e eu sanciono a seguinte Lei Complementar:

## **TÍTULO I** **Capítulo Único** **Das Disposições Preliminares**

Art. 1º - Esta Lei Complementar institui o regime jurídico dos Servidores Públicos Civis do Estado do Piauí, das autarquias e das fundações públicas estaduais, abrangendo os Poderes Legislativo, Executivo e Judiciário.

Art. 2º - Para os efeitos desta Lei Complementar, servidor é a pessoa legalmente investida em cargo publico.

Art. 3º - Cargo público é o conjunto de atribuições e responsabilidades cometidas a um servidor, dentro da estrutura organizacional da Administração Direta, das autarquias e das fundações públicas estaduais.

Parágrafo Único - Os cargos públicos, acessíveis a todos os brasileiros,<sup>2</sup> são criados por lei, com denominação própria, número certo e vencimentos pagos pelos cofres públicos, para provimento em caráter efetivo ou em comissão.

Art. 4º - É proibida a prestação de serviços gratuitos, salvo os casos previstos em lei.

Art. 5º - É proibido o desvio de função ou atribuir-se ao servidor encargos ou serviços diferentes daqueles próprios de seu cargo.

## **TÍTULO II** **Do Provimento e da Vacância** **CAPÍTULO I** **Do Provimento** **SEÇÃO I** **Das Disposições Gerais**

<sup>1</sup> Publicada no DOE nº 12, de 18.01.94.

<sup>2</sup> Vide inciso I do art. 37, da Constituição Federal, com a redação da EC nº 19 de 04.06.98, in DOU nº 106-E, de 05.06.98, *verbis*: “Art.37- A administração pública direta e indireta de qualquer dos Poderes da União, dos Estados, do Distrito Federal e dos Municípios obedecerá aos princípios de legalidade, impessoalidade, moralidade, publicidade e eficiência e, também, ao seguinte: I – os cargos, empregos e funções públicas são acessíveis aos brasileiros que preencham os requisitos estabelecidos em lei, assim como aos estrangeiros, na forma da lei.”. Vide art. 95 da Lei nº 6.815, de 19.08.80 (Estatuto do estrangeiro), *verbis*: “Art. 95-O estrangeiro residente no Brasil goza de todos os direitos reconhecidos aos brasileiros, nos termos da Constituição e das leis.”

Art. 6º – São requisitos básicos para a investidura em cargo público:

I - a nacionalidade brasileira;

II - o gozo dos direitos políticos;

III - a quitação com as obrigações militares e eleitorais;

IV - o nível de escolaridade exigido para o exercício do cargo;

V - a idade mínima de dezoito anos;

VI - aptidão física e mental.

§ 1º - As atribuições do cargo podem justificar as exigências de outros requisitos estabelecidos em lei.<sup>3</sup>

§ 2º - Às pessoas portadoras de deficiência<sup>4</sup> é assegurado o direito de se inscrever em concurso público para provimento de cargo cujas atribuições sejam compatíveis com a deficiência de que são portadoras; para tais pessoas serão reservadas até 20% (vinte por cento) das vagas oferecidas no concurso.

Art. 7º - No âmbito do Poder Executivo, o provimento dos cargos públicos, inclusive das autarquias e fundações públicas, far-se-á por ato do Governador do Estado, permitida a delegação de competência.

---

<sup>3</sup> Vide § 3º do art. 39 da CF, com a redação da EC nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*: “Art. 39-A União, os Estados, o Distrito Federal e os Municípios instituirão conselho de política de administração e remuneração de pessoal, integrado por servidores designados pelos respectivos Poderes. § 3º - Aplica-se aos servidores ocupantes de cargo público o disposto no art. 7º, IV, VII, VIII, IX, XII, XIII, XV, XVI, XVII, XVIII, XIX, XX, XXII e XXX, podendo a lei estabelecer requisitos diferenciados de admissão quando a natureza do cargo o exigir.”

<sup>4</sup> Vide inciso VIII do art. 37 da Constituição Federal, a seguir, “Art. 37, inciso VIII – A lei reservará percentual dos cargos e empregos públicos para as pessoas portadoras de deficiência e definirá os critérios de sua admissão.”  
Vide “Lei nº 4.835, de 23 de maio de 1996, DOE nº 101, de 27.05.96, *verbis*:

Define o percentual de cargos e empregos públicos para pessoas portadoras de deficiências, os critérios de sua admissão na Administração Pública e dá outras providências.

O GOVERNADOR DO ESTADO DO PIAUÍ

FAÇO saber que o Poder Legislativo decreta e eu sanciono a seguinte Lei:

Art. 1º - A Administração Pública Estadual Direta, Indireta e Fundacional, quando da realização de concursos públicos para provimento de vagas referentes a cargos e empregos públicos, fixará o percentual mínimo de 10% (dez por cento) das vagas a serem providas, para destinação aos portadores de deficiências.

§ 1º - Os editais dos concursos públicos fixarão os tipos de deficiência que garantirão aos seus portadores as inscrições nos respectivos processos seletivos.

§ 2º - Os mecanismos necessários para a avaliação e aferição de competência dos portadores de deficiências deverão ser definidos nos editais dos concursos públicos.

Art. 2º - Os portadores de deficiências terão asseguradas suas inscrições nos concursos públicos a serem realizados pela Administração Pública Estadual Direta, Indireta e Fundacional, isentos do pagamento de quaisquer taxas.

Art. 3º - Os editais de realização dos concursos públicos definirão, de forma objetiva, os critérios de admissão dos portadores de deficiências que sejam aprovados e classificados no processo seletivo.

Art. 4º - Esta Lei entrará em vigor da data de sua publicação, revogadas as disposições em contrário.

PALACIO DE KARNAK, em Teresina (PI), 23 de maio de 1996.

Francisco de Assis de Moraes Souza

Governador do Estado

Kleber Dantas Eulálio

Secretário de Governo

Carlos Alberto Teles de Sousa

Secretário de Administração

Parágrafo Único - Nos demais Poderes, o ato de provimento compete à autoridade indicada na respectiva legislação.<sup>5</sup>

Art. 8º - A investidura em cargo público ocorrerá com a posse.

Art. 9º - São formas de provimento de cargo público:

I - nomeação;

II - promoção;

III - transferência;

IV - readaptação;

V - reversão;

VI - aproveitamento;

VII - reintegração;

VIII - recondução.

## **SEÇÃO II** **Da Nomeação**

Art. 10 - A nomeação far-se-á:

I - em caráter efetivo, quando se tratar de cargo isolado de provimento efetivo ou de carreira;

II - em comissão, para cargos de confiança, de livre nomeação e exoneração.

§ 1º - O exercício de cargo em comissão exigirá de seu ocupante integral dedicação ao serviço, podendo o servidor ser convocado sempre que houver interesse da administração.

§ 2º - A designação para função de direção, assessoramento e chefia intermediários, de competência dos dirigentes de órgãos e entidades administrativas, recairá, exclusivamente, em servidor de carreira ou de cargo isolado de provimento efetivo.<sup>6</sup>

---

<sup>5</sup> Vide Resolução nº 174/91, art. 17, IV (Legislação prevista para o Poder Legislativo); Vide Lei nº 3.716/79 (Lei de Organização Judiciária do Estado do Piauí), art. 21, XXI, (Legislação prevista para o Poder Judiciário); Vide Lei Complementar nº 12/93 (Legislação prevista para o Ministério Público), art. 12, VI; Vide Lei nº 4.721/94 (Cria o Fundo de Modernização do Tribunal de Contas do Estado (FMTC) e dá outras providências), art. 76, III e parágrafo único do art. 77; Vide a exceção prevista no § 6º do art. 88 da Constituição Estadual, atualmente com a seguinte redação: § 6º - Os auditores, em número de cinco, com atribuições definidas em lei, serão nomeados pelo Governador do Estado, dentre bacharéis em Ciências Jurídicas e Sociais, em Ciências Econômicas, em Ciências Contábeis e Administração Pública, mediante prévia aprovação em concurso público.

<sup>6</sup> Vide inciso V do art. 37 da Constituição Federal, com nova redação dada pela Emenda Constitucional nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*:

“Art. 37.....  
.....

Art. 11 - A nomeação para cargo de carreira ou cargo isolado de provimento efetivo depende de prévia habilitação em concurso público de provas ou de provas e títulos, obedecidos a ordem de classificação e o prazo de sua validade.<sup>7</sup>

### **SEÇÃO III** **Do Concurso Público**

Art. 12 - O concurso público será de provas ou de provas e títulos, podendo ser realizado em duas etapas, conforme dispuser o edital, garantida a participação das entidades sindicais na fiscalização.

Art. 13 - O concurso público terá validade de até 2 (dois) anos, podendo ser prorrogado uma única vez, por igual período.<sup>8</sup>

§ 1º - O prazo de validade do concurso e as condições de sua realização serão fixadas em Edital, que será publicado no Diário Oficial do Estado e em jornal diário de grande circulação.

§ 2º - Não se abrirá novo concurso enquanto houver candidato aprovado em concurso anterior, com prazo de validade não expirado.

§ 3º - O concurso, uma vez aberto, deverá estar homologado no prazo de 12 (doze) meses.

### **SEÇÃO IV** **Da Posse e do Exercício**

Art. 14 - Posse é a investidura em cargo público e exercício é o efetivo desempenho das atribuições do cargo.

Art. 15 - A posse dar-se-á pela assinatura do respectivo termo, no qual deverão constar a declaração de bens e valores que constituem o patrimônio do servidor e a declaração quanto ao exercício ou não de outro cargo, emprego ou função pública ou privada.

---

V – As funções de confiança, exercidas exclusivamente por servidores ocupantes de cargo efetivo, e os cargos em comissão, a serem preenchidos por servidores de carreira nos casos, condições e percentuais mínimos previstos em lei, destinam-se apenas às atribuições de direção, chefia e assessoramento.”

<sup>7</sup> Vide inciso II do art. 37 da Constituição Federal, com nova redação dada pela Emenda Constitucional nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*:

“Art. 37.....

II – a investidura em cargo ou emprego público depende de aprovação prévia em concurso público de provas ou de provas e títulos, de acordo com a natureza e a complexidade do cargo ou emprego, na forma prevista em lei, ressalvadas as nomeações para cargo em comissão declarado em lei de livre nomeação e exoneração.”

<sup>8</sup> Vide inciso III do art. 37 da Constituição Federal, *verbis*:

“Art. 37.....

III – O prazo de validade do concurso público será de até dois anos, prorrogável uma vez, por igual período.”

§ 1º - Do termo de posse, assinado pela autoridade competente e pelo servidor, constará o compromisso do empossado de fiel cumprimento de seus deveres funcionais e de suas atribuições no cargo.

§ 2º - Poderá haver posse mediante procuração com poderes específicos para tal fim, inclusive o de assinar o termo e firmar o compromisso.

§ 3º - A posse ocorrerá no prazo de 30 (trinta) dias, contados da publicação do ato de provimento prorrogável por mais 30 (trinta dias) a requerimento do interessado. Se o servidor estiver em licença, ou afastado, legalmente, o prazo será contado do término do impedimento.

§ 4º - Será tornado sem efeito o ato de provimento, se a posse não ocorrer no prazo previsto no § 3º deste artigo.

§ 5º - A autoridade que der posse verificará se foram satisfeitas as condições legais para a investidura, na forma do disposto no art. 6º desta Lei Complementar.<sup>9</sup>

Art. 16 - Só haverá posse nos casos de nomeação para cargo de provimento efetivo ou em comissão e na reversão.

Art. 17 - A posse em cargo público dependerá de prévia inspeção médica oficial.

Parágrafo único – Só poderá ser empossado aquele que for julgado apto física e mentalmente para o exercício do cargo.

Art. 18 – É de 30 (trinta) dias o prazo para o servidor entrar em exercício, contados da data da posse. Findo o prazo e não estando em exercício, o servidor será exonerado.

§ 1º - Ao dirigente do órgão ou entidade para onde for designado o servidor compete dar-lhe exercício.

§ 2º - Ao entrar em exercício, o servidor apresentará ao órgão competente os elementos necessários ao seu assentamento individual.

§ 3º - É obrigatório o registro da frequência do servidor na unidade administrativa onde tem lotação na conformidade com as normas regulamentares.

§ 4º - O início, a suspensão, a interrupção e o reinício do exercício serão registrados no assentamento individual do servidor.

§ 5º - Será considerado como de efetivo exercício o período de tempo realmente necessário ao deslocamento do servidor, quando designado para servir em outra localidade. Se o servidor estiver afastado, legalmente, o prazo será contado a partir do término do afastamento.

§ 6º - A promoção não interrompe o tempo de exercício, que é contado na nova classe a partir da data da publicação do ato que promover o servidor.

---

<sup>9</sup> Vide §§ 1º, 2º e 3º da Lei Federal nº 8.429, de 02 de junho de 1992.

Art. 19 - Ao entrar em exercício, o servidor nomeado para cargo de provimento efetivo ficará sujeito a estágio probatório por período de 24 (vinte e quatro) meses,<sup>10</sup> durante o qual a sua aptidão e capacidade serão objeto de avaliação para o desempenho do cargo, observado, também, os seguintes fatores:

I - assiduidade;

II- disciplina;

III - produtividade;

IV - responsabilidade.

§ 1º - Antes de terminar o período do estágio probatório, será submetida a homologação da autoridade dirigente do órgão ou da entidade pública, a avaliação do desempenho do servidor, realizada de acordo com o que dispuser o regulamento.

§ 2º - O servidor não aprovado no estágio probatório será exonerado ou, se estável, reconduzido ao cargo anteriormente ocupado.<sup>11</sup>

§ 3º - Não haverá para o servidor, no período do estágio probatório, promoção, progressão ou transferência, permitida a readaptação, na forma do regulamento.

## **SEÇÃO V** **Da Estabilidade**

Art. 20 - O servidor, nomeado por concurso público para cargo de provimento efetivo, adquirirá estabilidade no serviço público ao completar 2 (dois) anos de efetivo exercício.<sup>12</sup>

Art. 21 - O servidor estável só perderá o cargo em virtude de sentença judicial, transitada em julgado, ou de processo administrativo disciplinar no qual lhe seja assegurada ampla defesa.<sup>13</sup>

---

<sup>10</sup> Vide art. 41 da Constituição Federal, com redação dada pela EC nº 19, de 04.06.98, DOU 106-E, de 05.06.98, *verbis*: “Art. 41 – São estáveis após três anos de efetivo exercício os servidores nomeados para cargo de provimento efetivo em virtude de concurso público.” Vide art. 28 da EC nº 19 de 14.06.98, *verbis*: “É assegurado o prazo de dois anos de efetivo exercício para aquisição da estabilidade aos atuais servidores em estágio probatório, sem prejuízo da avaliação a que se refere o § 4º do art. 41 da Constituição Federal.”

<sup>11</sup> Vide súmulas 21 e 22 do Supremo Tribunal Federal: “Súmula nº 21 – funcionário em estágio probatório não pode ser exonerado nem demitido sem inquérito ou sem as formalidades legais de apuração de sua capacidade” e “Súmula nº 22 – O estágio probatório não protege o funcionário contra a extinção do cargo.”

<sup>12</sup> Vide art. 41 e seu § 4º da Constituição Federal, com redação dada pela EC nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*:

“Art. 41 – São estáveis após três anos de efetivo exercício os servidores nomeados para cargo de provimento efetivo em virtude de concurso público.

.....  
4º - Como condição para aquisição da estabilidade, é obrigatória a avaliação especial de desemprego por comissão instituída para essa finalidade.”

<sup>13</sup> Vide art. 41 e seus §§ e incisos da Constituição Federal, com redação dada pela EC nº 19, de 04.06.98, DOU 106-E, de 05.06.98, *verbis*:

“Art. 41 - São estáveis após três anos de efetivo exercício os servidores nomeados para cargo de provimento efetivo em virtude de concurso público.

§ 1º - O servidor público estável só perderá o cargo:

I – em virtude de sentença judicial transitada em julgado;



§ 1º - Invalidada a demissão do servidor estável, será ele reintegrado, na forma do art. 31, desta Lei Complementar, e o eventual ocupante de seu cargo reconduzido ao cargo de origem, aproveitado em outro cargo ou posto em disponibilidade, sem direito a indenização.<sup>14</sup>

§ 2º - Extinto o cargo ou declarada a sua desnecessidade, o servidor estável ficará em disponibilidade remunerada, até o seu adequado aproveitamento em outro cargo.<sup>15</sup>

## SEÇÃO VI Da Promoção

Art. 22 – Promoção é a elevação do servidor<sup>16</sup> à classe imediatamente superior àquela a que pertence, na respectiva carreira.

---

II – mediante processo administrativo em que lhe seja assegurada ampla defesa;

III – mediante procedimento de avaliação periódica de desempenho, na forma de lei complementar, assegurada ampla defesa.

§ 2º - Invalidada por sentença judicial a demissão do servidor estável, será ele reintegrado, e o eventual ocupante da vaga, se estável, reconduzido ao cargo de origem, sem direito a indenização, aproveitado em outro cargo ou posto em disponibilidade com remuneração proporcional ao tempo de serviço.

§ 3º - Extinto o cargo ou declarada a sua desnecessidade, o servidor estável ficará em disponibilidade, com remuneração proporcional ao tempo de serviço, até seu adequado aproveitamento em outro cargo.

§ 4º - Como condição para aquisição da estabilidade, é obrigatória a avaliação especial de desempenho por comissão instituída para essa finalidade.”

Vide, também, o art. 169 e seu § 4º e 5º da Constituição Federal, com redação dada pela EC nº 19, de 04.06.98, DOU 106-E, de 05.06.98, *verbis*:

“Art. 169 – A despesa com pessoal ativo e inativo da União, dos Estados, do Distrito Federal e dos Municípios não poderá exceder os limites estabelecidos em lei complementar.

.....  
§ 4º - Se as medidas adotadas com base no parágrafo anterior não forem suficientes para assegurar o cumprimento da determinação da lei complementar referida neste artigo, o servidor estável poderá perder o cargo, desde que ato normativo motivado de cada um dos Poderes especifique a atividade funcional, o órgão ou unidade administrativa objeto da redução de pessoal.

§ 5º - O servidor que perder o cargo na forma do parágrafo anterior fará jus a indenização correspondente a um mês de remuneração por ano de serviço.”

<sup>14</sup> Vide § 2º do art. 41 da Constituição Federal, com redação dada pela EC nº 19, de 04.06.98, DOU 106-E, de 05.06.98, *verbis*:

“Art. 41 - .....

.....  
§ 2º - Invalidada por sentença judicial a demissão do servidor estável, será ele reintegrado, e o eventual ocupante da vaga, se estável, reconduzido ao cargo de origem, sem direito a indenização, aproveitado em outro cargo ou posto em disponibilidade com remuneração proporcional ao tempo de serviço.”

<sup>15</sup> Vide § 3º do art. 41 da Constituição Federal, com redação dada pela EC nº 19, de 04.06.98, DOU 106-E, de 05.06.98, *verbis*:

“Art. 41 - .....

.....  
§ 3º - Extinto o cargo ou declarada a sua desnecessidade, o servidor estável ficará em disponibilidade, com remuneração proporcional ao tempo de serviço, até seu adequado aproveitamento em outro cargo.”

Vide, também, art. 33 da Constituição Federal, com redação dada pela EC nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*:

“Art. 33 – Consideram-se servidores não estáveis, para fins do art. 169, § 3º, II, da Constituição Federal aqueles admitidos na administração direta, autárquica e fundacional sem concurso público de provas ou de provas e títulos após o dia 5 de outubro de 1983.”

<sup>16</sup> Vide § 2º do art. 39 da Constituição Federal, com redação dada pela EC nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*:

“Art. 39 – A União, os Estados, o Distrito Federal e os Municípios instituirão conselho de política de administração e remuneração de pessoal, integrado por servidores designados pelos respectivos Poderes.

.....  
§ 2º - A União, os Estados e o Distrito Federal manterão escolas de governo para formação e aperfeiçoamento dos servidores públicos, constituindo-se a participação nos cursos um dos requisitos para a promoção na carreira, facultada, para isso, a celebração de convênios ou contratos entre os entes federados.”

Vide inciso IV do art. 7º, art. 62 a 64 da Lei Complementar Estadual nº 12, de 18.12.93.

§ 1º - A promoção obedecerá aos critérios de merecimento e antigüidade de classe, exigindo sempre o interstício regulamentar.

§ 2º - As promoções serão realizadas de seis em seis meses, observadas as normas do regulamento.

§ 3º - O merecimento será aferido segundo critérios objetivos, indicados em regulamento.

§ 4º - Em cada órgão da administração estadual funcionará uma Comissão Permanente de avaliação do servidor, para fins de promoção.

## **SEÇÃO VII** **Da Transferência**

Art. 23 - Transferência é a movimentação do servidor de um cargo de provimento efetivo para outro cargo vago, da mesma denominação e vencimento de quadro diverso, dentro da Administração Direta da autarquia e da fundação pública.

Art. 24 - A transferência poderá ser atendida a pedido do servidor ou processada de ofício no interesse da administração.

Art. 25 - A readaptação é a investidura do servidor em cargo de atribuições e responsabilidades compatíveis com a limitação que tenha sofrido em sua capacidade física ou mental, verificada em inspeção médica.

Parágrafo Único - A readaptação será efetivada em cargo de atribuições afins, respeitada a habilitação exigida; se julgado incapaz para o serviço público o readaptando será aposentado.

Art. 26 - A transferência, por permuta, far-se-á a pedido das partes interessadas, observada a conveniência da administração.

Art. 27 - Não se dará transferência, se já abertas as inscrições para concurso ou se ainda houver candidato habilitado em concurso anterior, para o cargo a ser provido.

## **SEÇÃO VIII** **Da Reversão**

Art. 28 - A reversão é o reingresso no serviço público do servidor aposentado por invalidez, quando, por junta médica oficial, forem declarados insubsistentes os motivos da aposentadoria.

§ 1º - A reversão far-se-á, de preferência, no mesmo cargo, ou em cargo vago da mesma denominação e vencimento.

§ 2º - Não poderá reverter o aposentado que já tiver completado 60 (sessenta) anos de idade ou 30 (trinta) anos de tempo de serviço.

Art. 29 - Somente por necessidade do serviço e no interesse público, a critério da administração, dar-se-á a reversão de aposentado.

## **SEÇÃO IX Do Aproveitamento**

Art. 30 - Aproveitamento é o reingresso no serviço público do servidor em disponibilidade.

§ 1º - Será obrigatório o aproveitamento do servidor em disponibilidade em cargo de atribuições e vencimentos compatíveis com o anteriormente ocupado.

§ 2º - O retorno à atividade do servidor em disponibilidade far-se-á em vaga que vier a ocorrer nos órgãos da Administração Direta, das autarquias e das fundações públicas, respectivamente da origem do servidor.

§ 3º - Será tornado sem efeito o aproveitamento e cassada a disponibilidade se o servidor não entrar em exercício no prazo de 30 (trinta) dias, salvo doença comprovada por junta médica oficial.

## **SEÇÃO X Da Reintegração**

Art. 31 - A reintegração é a reinvestidura do servidor estável no cargo anteriormente ocupado ou no cargo resultante de sua transformação, quando invalidada a sua demissão, por decisão administrativa ou sentença judicial, transitada em julgado, com ressarcimento de todas as vantagens.

§ 1º - A reintegração será feita no cargo anteriormente ocupado; encontrando-se provido o cargo, o seu eventual ocupante será reconduzido ao cargo de origem ou aproveitado em outro cargo de igual padrão, sem direito a indenização.

§ 2º - Se extinto o cargo anteriormente exercido, o servidor ficará em disponibilidade remunerada até o seu posterior aproveitamento.<sup>17</sup>

## **SEÇÃO XI Da Recondução**

Art. 32 - Recondução é o retorno do servidor estável ao seu cargo de origem, em decorrência da reintegração de seu anterior ocupante.

Parágrafo Único - Aplica-se à recondução, no que couber, o disposto no artigo anterior.

---

<sup>17</sup> Vide § 3º do art. 41 da Constituição Federal, com redação dada pela EC nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*:

“Art. 41 - .....  
§ 3º - Extinto o cargo ou declarada a sua desnecessidade, o servidor estável ficará em disponibilidade, com remuneração proporcional ao tempo de serviço, até seu adequado aproveitamento em outro cargo.”

## **CAPÍTULO II**

### **Da Vacância**

Art. 33 - A vacância do cargo público decorrerá de:

- I – exoneração;
- II - demissão;
- III - promoção;
- IV- transferência;
- V- readaptação;
- VI - aposentadoria;
- VII - posse em outro cargo inacumulável;
- VIII – falecimento.

Art. 34 - A exoneração de cargo público dar-se-á a pedido do servidor, ou de ofício.

Parágrafo Único - A exoneração de ofício dar-se-á:

- I - quando não satisfeitas as condições do estágio probatório;
- II - quando, tendo tomado posse, o servidor não entrar em exercício, no prazo determinado;
- III - a juízo da autoridade competente, quando se tratar de cargo em comissão.<sup>18</sup>

Art. 35 - Quando se tratar de função de direção, chefia e assessoramento dar-se-á a vacância por dispensa, a pedido ou de ofício, ou por destituição.

## **CAPÍTULO III**

### **Da Remoção e da Substituição**

#### **SEÇÃO I**

#### **Da Remoção**

---

<sup>18</sup> Vide §§ 4º, 6º e 7º do art. 169 da Constituição Federal, com nova redação dada pela EC nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*:

“Art. 169 – A despesa com pessoal ativo e inativo da União, dos Estados, do Distrito Federal e dos Municípios não poderá exceder os limites estabelecidos em lei complementar.

.....  
§ 4º - Se as medidas adotadas com base no parágrafo anterior não forem suficientes para assegurar o cumprimento da determinação da lei complementar referida neste artigo, o servidor estável poderá perder o cargo, desde que ato normativo motivado de cada um dos Poderes especifique a atividade funcional, o órgão ou unidade administrativa objeto da redução de pessoal;

§ 6º - O cargo objeto da redução prevista nos parágrafos anteriores será considerado extinto, vedada a criação de cargo, emprego ou função com atribuições iguais ou assemelhadas pelo prazo de quatro anos;

§ 7º - Lei federal disporá sobre as normas gerais a serem obedecidas na efetivação do disposto no § 4º.”

Art. 36 – Remoção é o deslocamento do servidor, a pedido, de ofício ou por permuta, no âmbito do mesmo quadro, com ou sem mudança de sede e sem que se modifique a sua situação funcional.

Art. 37 - A remoção far-se-á a pedido, atendida a conveniência do serviço e de ofício ou por permuta, no interesse da administração.

§ 1º - Dar-se-á a remoção, a pedido, para outra localidade, independentemente de vaga, para acompanhar cônjuge ou companheiro, ou por motivo de saúde do servidor, cônjuge, companheiro ou dependente, condicionada à comprovação por junta médica.

§ 2º - A remoção será sempre motivada por escrito pela autoridade competente, sob pena de nulidade.<sup>19</sup>

Art. 38 - O Chefe do Poder Executivo, no interesse público, fica autorizado a proceder ao deslocamento do cargo de uma classe para outra.

## **SEÇÃO II**

### **Da Substituição**

Art. 39 - Os servidores investidos em função de direção ou chefia e os ocupantes de cargos em comissão terão substitutos indicados em regimento interno ou, no caso de omissão, designados pela autoridade competente.

§ 1º - O substituto assumirá automaticamente o exercício do cargo ou função de direção, assessoramento ou chefia nos afastamentos ou impedimentos regulamentares do titular.

§ 2º - O substituto fará jus à gratificação pelo exercício da função de direção, assessoramento ou chefia, paga na proporção dos dias de efetiva substituição.

§ 3º - Não cabe gratificação ao servidor, quando a substituição for inerente às atribuições do seu cargo, salvo se o período da substituição ultrapassar a 30 (trinta) dias corridos.

## **TÍTULO III**

### **Dos Direitos e Vantagens**

#### **CAPÍTULO I**

#### **Do Vencimento e da remuneração**

Art. 40 - Vencimento é a retribuição pecuniária pelo exercício de cargo público, com valor fixado em lei.<sup>20</sup>

<sup>19</sup> Vide incisos III e V do art. 145 do Código Civil, *verbis*:

“Art. 145 – É nulo o ato jurídico:

.....  
III – Quando não revestir a forma prescrita em lei (arts. 82 e 130).  
.....

V – Quando a lei taxativamente o declarar nulo ou lhe negar efeito.”

<sup>20</sup> Vide incisos I a III do § 1º do art. 39 da Constituição Federal, com nova redação dada pela EC nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*: “Art. 39 – A União, os Estados, o Distrito Federal e os Municípios instituirão conselho de política de administração e remuneração de pessoal, integrado por servidores designados pelos respectivos Poderes.

§ 1º - A fixação dos padrões de vencimento e dos demais componentes do sistema remuneratório observará:

§ 1º - Nenhum servidor poderá perceber mensalmente, a título de remuneração<sup>21</sup>, importância superior à soma dos valores percebidos, em espécie, a qualquer título, no âmbito dos respectivos Poderes, pelo Deputado Estadual, pelo Desembargador e pelo Secretário de Estado, não se incluindo neste teto o salário-família e as vantagens previstas no parágrafo único do art. 206 e nos incisos I, II, III, IV, VII, IX, X, XI e XII, do art. 55, desta Lei Complementar.<sup>22</sup>

§ 2º - É vedada a vinculação ou equiparação de vencimentos, para efeito de remuneração do pessoal do serviço público, ressalvado os casos previstos na Constituição Federal.<sup>23</sup>

Art. 41 – Remuneração é o vencimento do cargo efetivo, acrescido das vantagens pecuniárias permanentes, estabelecidas em lei.

§ 1º - A remuneração dos cargos em comissão compreende o vencimento e a gratificação de representação, fixados em lei.<sup>24</sup>

---

I – a natureza, o grau de responsabilidade e a complexidade dos cargos componentes de cada carreira;

II – os requisitos para a investidura;

III – as peculiaridades dos cargos.”

<sup>21</sup> Vide art. 7º da LC nº 09, de 12.03.92, *verbis*: “Art. 7º - O limite máximo de remuneração dos servidores públicos civis e militares ativos e inativos, da Administração Pública do Estado do Piauí, é o valor da remuneração percebida, em espécie, a qualquer título, pelo Secretário de Estado, nessa qualidade.

§ 1º - Não se incluem, no cômputo do limite máximo de remuneração, as seguintes vantagens, previstas em Lei:

I - gratificação adicional por tempo de serviço;

II - gratificação de representação de gabinete, pelo exercício de cargo ou função de direção, chefia e assessoramento;

III - gratificação natalina;

IV - adicional de férias;

V - adicional noturno;

VI - adicional pelo exercício de atividades insalubres, perigosas ou penosas;

VII - progressão horizontal e/ou biênio por tempo de serviço.

§ 2º - No caso da remuneração dos servidores ativos, os proventos dos inativos e as pensões ultrapassarem o limite máximo, será o excedente automaticamente eliminado.

Vide, também, o Art. 8º - Fica fixada em até quarenta vezes a relação de valores entre a maior e a menor remuneração dos servidores públicos civis e militares ativos e inativos, da Administração Pública do Estado do Piauí.”

<sup>22</sup> Vide ADIN - 1331-9. Relator: MINISTRO FRANCISCO REZEK. O STF, por maioria de votos, referendou a decisão do Presidente (Min. Sepúlveda Pertence), que deferira, em parte, a medida liminar que suspendera, até a decisão final da ação, a eficácia da expressão "prevista no parágrafo único do art. 206". Plenário em 16.09.98. Aguardando Julgamento

<sup>23</sup> Vide incisos X, XI, XIII e XIV do art. 37 da Constituição Federal, com nova redação dada pela Emenda Constitucional nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*:

“Art. 37.....

.....

X – A remuneração dos servidores públicos e o subsídio de que trata o § 4º do art. 39 somente poderão ser fixados ou alterados por lei específica, observada a iniciativa privativa em cada caso, assegurada revisão geral anual, sempre na mesma data e sem distinção de índices;

XI – A remuneração e o subsídio dos ocupantes de cargos, funções e empregos públicos da administração direta, autárquica e fundacional, dos membros de qualquer dos Poderes da União, dos Estados, do Distrito Federal e dos Municípios, dos detentores de mandato eletivo e dos demais agentes políticos e os proventos, pensões ou outra espécie remuneratória, percebidos cumulativamente ou não, incluídas as vantagens pessoais ou de qualquer outra natureza, não poderão exceder o subsídio mensal, em espécie, dos Ministros do Supremo Tribunal Federal;

XIII – É vedada a vinculação ou equiparação de quaisquer espécies remuneratórias para o efeito de remuneração de pessoal do serviço público;

XIV – Os acréscimos pecuniários percebidos por servidor público não serão computados nem acumulados para fins de concessão de acréscimos ulteriores.

<sup>24</sup> Vide §§ 4º, 5º e 8º do art. 39 e art. 135 da Constituição Federal, com nova redação dada pela Emenda Constitucional nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*:

“Art. 39 - A União, os Estados, o Distrito Federal e os Municípios instituirão conselho de política de administração e remuneração de pessoal, integrado por servidores designados pelos respectivos Poderes.

§ 4º - O membro de Poder, o detentor de mandato eletivo, os Ministros de Estado e os Secretários Estaduais e Municipais serão remunerados exclusivamente por subsídio fixado em parcela única, vedado o acréscimo de qualquer gratificação, adicional, abono, prêmio, verba de representação ou outra espécie remuneratória, obedecido, em qualquer caso, o disposto no art. 37, X e XI;

§ 2º - O servidor nomeado para cargo em comissão poderá fazer opção pelo vencimento de seu cargo efetivo, acrescido da gratificação de representação do cargo em comissão, para o qual foi nomeado.

Art. 42 - O vencimento do cargo efetivo, acrescido das vantagens de caráter permanente, é irredutível.<sup>25</sup>

§ 1º - Salvo por imposição legal, ou mandado judicial, nenhum desconto incidirá sobre a remuneração ou proventos.

§ 2º - Mediante autorização do servidor, poderá haver consignação em folha de pagamento a favor de terceiros, a critério da administração e com reposição de custos, salvo quanto aos recolhimentos sindicais e associações representativas de classe.

§ 3º - As reposições e indenizações ao erário serão descontadas em parcelas mensais não excedentes à décima parte da remuneração ou provento, em valores atualizados.<sup>26</sup>

§ 4º - O servidor perderá a remuneração dos dias em que faltar ao serviço e a parcela de remuneração diária, proporcional aos atrasos, ausências e saídas antecipadas, iguais ou superiores a 60 (sessenta) minutos.

## **CAPÍTULO II** **Das Vantagens**

Art. 43 - Além do vencimento, poderão ser pagas ao servidor:

I – indenizações;

II- gratificações;

III – adicionais.

§ 1º - As indenizações não se incorporam ao vencimento ou provento para qualquer efeito e não servem de base para cálculo de quaisquer outras vantagens.

---

§ 5º - Lei da União, dos Estados, do Distrito Federal e dos Municípios poderá estabelecer a relação entre a maior e a menor remuneração dos servidores públicos, obedecido, em qualquer caso, o disposto no art. 37, XI;

§ 8º - A remuneração dos servidores públicos organizados em carreira poderá ser fixada nos termos do § 4º.

Art. 135 – Os servidores integrantes das carreiras disciplinadas nas Seções II e III deste Capítulo serão remunerados na forma do art. 39, § 4º.”

Nota: As referidas seções II e III do Capítulo IV da Constituição Federal tratam da mesma matéria constante das seções II e III do Capítulo III do Título IV (arts. 150 a 155 da Constituição do Estado do Piauí).

<sup>25</sup> Vide inciso XV do art. 37 da Constituição Federal, com redação dada pela EC nº 19, de 04.06.98, in DOU 106-E, de 05.06.98, *verbis*:

“Art. 37 .....  
.....

XV – O subsídio e os vencimentos dos ocupantes de cargos e empregos públicos são irredutíveis, ressalvado o disposto nos incisos XI e XIV deste artigo e nos arts. 39, § 4º, 150, II, 153, III, e 153, § 2º, I.”

<sup>26</sup> Vide § 6º do art. 37 da Constituição Federal, *verbis*:

“Art. 37 - .....  
.....

§ 6º - As pessoas jurídicas de direito público e as de direito privado prestadoras de serviços públicos responderão pelos danos que seus agentes, nessa qualidade, causarem a terceiros, assegurado o direito de regresso contra o responsável nos casos de dolo ou culpa.”

§ 2º - As gratificações e os adicionais incorporam-se aos vencimentos e aos proventos, nos casos e condições indicados em lei.<sup>27</sup>

Art. 44 - É vedada a concessão de quaisquer outras vantagens pecuniárias, gratificações e adicionais não previstos em Lei Complementar, bem como em bases e limites superiores aos nela fixados.

Parágrafo Único - A Progressão Horizontal é a passagem do servidor de uma referência para outra imediatamente superior, dentro da mesma classe, observados os interstícios e o tempo de serviço, na carreira, na forma regulamentar.<sup>28</sup>

## **SEÇÃO I** **Das Indenizações**

Art. 45 - Constituem indenizações ao servidor:

I - ajuda de custo;

II - diárias;

III - ajuda de transporte.

Parágrafo Único - Os valores das indenizações, assim como as condições para a sua concessão, serão estabelecidos por ato do respectivo Poder.

## **SUBSEÇÃO I** **Da Ajuda de Custo**

Art. 46 - A ajuda de custo destina-se a compensar as despesas de instalação do servidor que, no interesse do serviço, passar a ter exercício em nova sede, com mudança de domicílio permanente.

§ 1º - Correm por conta da administração as despesas de transportes do servidor e de sua família, compreendendo passagem, bagagem e bens pessoais.

§ 2º - À família do servidor que falecer na nova sede são assegurados ajuda de custo e transporte para a localidade de origem, dentro do prazo de um ano, contado do óbito.

Art. 47 - Será concedido ajuda de custo àquele que, não sendo servidor público, for nomeado para cargo em comissão, com mudança de domicílio.

Parágrafo Único - Nos afastamentos previstos no Capítulo V, desta Lei Complementar, a ajuda de custo será paga pelo órgão requisitante, quando cabível.

---

<sup>27</sup> Vide o segundo rodapé constante do § 1º do art. 40 da Lei Complementar nº 13, de 03.01.94.

<sup>28</sup> As disposições do parágrafo único do art. 44 encontram-se com aplicação suspensa, junto com o parágrafo único do art. 206, ambos da Lei Complementar nº 13, de 03.01.94, por decisão liminar do Exmo. Sr. Presidente do Supremo Tribunal Federal, datada de 13.07.95, referendada, por maioria de votos, pelo Tribunal Pleno, em 16.08.95 – ADI nº 1331-9-Piauí (vide segundo rodapé do § 1º, do art. 40 da Lei Complementar nº 13/94)



Art. 48 - Não será concedida ajuda de custo ao servidor que se afastar do cargo, ou reassumi-lo, em virtude de mandato eletivo.

Art. 49 - A ajuda de custo é calculada sobre a remuneração do servidor, não podendo exceder à importância correspondente a 3 (três) meses.

Art. 50 - O servidor será obrigado a restituir a ajuda de custo quando, injustificadamente, não se apresentar na nova sede, no prazo de 30 (trinta) dias.

## **SUBSEÇÃO II**

### **Das Diárias**

Art. 51 - O servidor que, a serviço, se deslocar da sua sede, em caráter eventual ou transitório, fará jus a passagens e diárias, para cobrir as despesas de alimentação e pousada.

§ 1º - A diária será concedida por dia de afastamento, sendo devida pela metade quando o deslocamento não exigir pernoite fora da sede.

§ 2º - Nos casos em que o deslocamento da sede constituir exigência permanente do cargo, o servidor não fará jus a diárias.

Art. 52 - O valor das diárias será fixado por ato do respectivo Poder, de acordo com a natureza, o local e as condições do serviço.

Art. 53 - O servidor que receber diárias e não se afastar da sede, por qualquer motivo, fica obrigado a restituí-las, integralmente, no prazo de 5 (cinco) dias.

Parágrafo Único - Se o servidor retornar à sede em prazo menor do que o previsto para o seu afastamento, restituirá as diárias recebidas em excesso, no mesmo prazo deste artigo.

## **SUBSEÇÃO III**

### **Da Indenização de Transporte**

Art. 54 - Conceder-se-á indenização de transporte ao servidor que realizar despesas com a utilização de meio próprio de locomoção para a execução de serviços externos, por força das atribuições próprias do cargo.

Parágrafo Único - Conceder-se-á, vale-transporte ao servidor .... VETADO...

## **SEÇÃO II**

### **Das Gratificações e Adicionais**

Art. 55 - Além do vencimento e das indenizações previstas nesta Lei Complementar, serão deferidos aos servidores públicos as seguintes gratificações e adicionais:

I - gratificação pelo exercício de cargo ou função de Direção, Chefia e Assessoramento;<sup>29</sup>

II – gratificação natalina;

III – gratificação pela prestação de serviço extraordinário;

IV - gratificação pelo exercício de atividades insalubres, perigosas e penosas;

V - gratificação pela participação em órgão de deliberação coletiva;

VI - gratificação de representação de gabinete;

VII - gratificação de Controle Interno e Auditoria;<sup>30</sup>

VIII - gratificação por Condições Especiais de Trabalho;

IX - adicional por Tempo de Serviço;

X - adicional Noturno;

XI - adicional de Férias;

XII - adicional de Produtividade.<sup>31</sup>

## **SUBSEÇÃO I**

### **Da Participação pelo Exercício de Cargo ou Função de Direção, Chefia e Assessoramento**

<sup>29</sup> Vide ADIN 1331-9 – Rel. Min. Francisco Resek. O STF, ao julgar a medida liminar, decidiu: “Desse modo, DEFIRO em parte o pedido de liminar para, ad referendum do Plenário, suspender, até decisão desta ação direta, a vigência, no § 1º do art. 40, das expressões “previstas no parágrafo único do art. 206”, assim como da menção, nele contida, aos incisos 00I, VII e XII, do art. 055, todos da LC nº 013, de 03.01.94, do Estado do Piauí. Por MAIORIA de votos, o Tribunal REFERENDOU a decisão do presidente (Ministro Sepúlveda Pertence), que deferira, em parte, a medida liminar que suspendera, até a decisão final da ação, a eficácia da expressão “previstas no parágrafo único do art. 206”, contida no art. 040, assim como da menção, nele contida, aos incisos 00I, VII e XII, do art. 055, todos da LC nº 013, de 03.01.94, do Estado do Piauí. Vencidos, em parte, os Ministros Marco Aurélio e Octavio Gallotti, que deferiam integralmente a medida liminar. Votou o presidente.” Plenário, 16.08.95. Aguardando julgamento do mérito.

<sup>30</sup> Vide ADIN 1331-9 – Rel. Min. Francisco Resek. O STF, ao julgar a medida liminar, decidiu: “Desse modo, DEFIRO em parte o pedido de liminar para, ad referendum do Plenário, suspender, até decisão desta ação direta, a vigência, no art. 40, das expressões “previstas no parágrafo único do art. 206”, assim como da menção, nele contida, aos incisos 00I, VII e XII, do art. 055, todos da Lei Complementar nº 013, de 03.01.94, do Estado do Piauí. Por MAIORIA de votos, o Tribunal REFERENDOU a decisão do presidente (Ministro Sepúlveda Pertence), que deferira, em parte, a medida liminar que suspendera, até a decisão final da ação, a eficácia da expressão “previstas no parágrafo único do art. 206”, contida no art. 040, assim como da menção, nele contida, aos incisos 00I, VII e XII, do art. 055, todos da Lei Complementar nº 013, de 03.01.94, do Estado do Piauí. Vencidos, em parte, os Ministros Marco Aurélio e Octavio Gallotti, que deferiam integralmente a medida liminar. Votou o presidente.” Plenário, 16.08.95. Aguardando julgamento do mérito.

<sup>31</sup> Vide ADIN 1331-9 – Rel. Min. Francisco Resek. O STF, ao julgar a medida liminar, decidiu: “Desse modo, DEFIRO em parte o pedido de liminar para, ad referendum do Plenário, suspender, até decisão desta ação direta, a vigência, no art. 40, das expressões “previstas no parágrafo único do art. 206”, assim como da menção, nele contida, aos incisos 00I, VII e XII, do art. 055, todos da Lei Complementar nº 013, de 03.01.94, do Estado do Piauí. Por MAIORIA de votos, o Tribunal REFERENDOU a decisão do presidente (Ministro Sepúlveda Pertence), que deferira, em parte, a medida liminar que suspendera, até a decisão final da ação, a eficácia da expressão “previstas no parágrafo único do art. 206”, contida no art. 040, assim como da menção, nele contida, aos incisos 00I, VII e XII, do art. 055, todos da Lei Complementar nº 013, de 03.01.94, do Estado do Piauí. Vencidos, em parte, os Ministros Marco Aurélio e Octavio Gallotti, que deferiam integralmente a medida liminar. Votou o presidente.” Plenário, 16.08.95. Aguardando julgamento do mérito.

Art. 56 - Ao servidor investido em cargo em comissão ou função de direção, chefia ou assessoramento é devida uma gratificação pelo seu exercício.

§ 1º - A gratificação, prevista neste artigo, como antecipação do disposto no art. 136, desta Lei Complementar, integra a remuneração do servidor, na proporção de 1/5 (um quinto) por ano, continuado ou não, até o limite de 5/5 (cinco quintos).

§ 2º - O servidor somente fará jus à gratificação de que trata o parágrafo anterior, se tiver exercido, na administração pública, cargo em comissão ou função, por período de 05 (cinco) anos consecutivos ou 10 (dez) anos intercalados.

§ 3º - Quando mais de uma função houver sido desempenhada no período de um ano, a gratificação terá como base de cálculo a função exercida por maior tempo.

§ 4º - Quando o exercício da Função ou Cargo em Comissão de maior valor não corresponder ao período de dois anos, será devida a gratificação imediatamente inferior dentre os exercícios.

§ 5º - Esta gratificação não servirá de base para cálculo de quaisquer outras vantagens ou adicionais que forem devidos ao servidor e somente será concedida mediante comprovação do ato a que se referem o art. 7º e seu parágrafo único desta Lei Complementar.

§ 6º - A gratificação, de que trata este artigo, terá vigência a partir de 1º de janeiro de 1994.

## **SUBSEÇÃO II** **Da Gratificação Natalina**

Art. 57 - A gratificação natalina corresponde a 1/12 (um doze avos) da remuneração a que o servidor fizer jus no mês de dezembro, por mês de exercício.

Parágrafo Único – A fração igual ou superior a 15 (quinze) dias será considerada como mês integral.

Art. 58 - O servidor exonerado perceberá sua gratificação natalina, proporcionalmente aos meses de exercício, calculada sobre a remuneração do mês da exoneração.

## **SUBSEÇÃO III** **Da Gratificação pela Prestação de Serviço Extraordinário**

Art. 59 - A gratificação pela prestação de serviço extraordinário será paga por hora de trabalho prorrogado ou antecipado do expediente normal do servidor.

§ 1º - O serviço extraordinário será remunerado com acréscimo de 50% (cinquenta por cento) em relação à hora normal de trabalho.

§ 2º - Somente em casos excepcionais, a critério da administração poderá ser antecipado ou prorrogado o período normal de trabalho do servidor, não podendo, porém,

exceder a 02 (duas) horas diárias e de 60 (sessenta) dias consecutivos ou 120 (cento e vinte) dias, interpolados, em cada ano.

#### **SUBSEÇÃO IV**

##### **Da Gratificação pelo Exercício de Atividades Insalubres, Perigosas e Penosas**

Art. 60 - Aos servidores que trabalham com habitualidade em locais insalubres ou em contato permanente com substâncias tóxicas, radioativas ou com risco de vida, fazem jus a uma gratificação sobre o vencimento básico do cargo efetivo.

§ 1º - A gratificação de que trata este artigo será calculada sobre o vencimento básico do cargo, na forma e condições estabelecidas em regulamento, observada a legislação federal específica.

§ 2º - O servidor que fizer jus à gratificação de insalubridade e periculosidade deverá optar por uma delas.

§ 3º - O direito à gratificação de que trata este artigo cessa com a eliminação das condições ou dos riscos que deram causa a sua concessão.

§ 4º - A caracterização e a classificação da insalubridade ou da periculosidade serão feitas nas condições disciplinadas na legislação específica.

§ 5º - A servidora gestante ou lactante será afastada das operações e locais previstos neste artigo, exercendo suas atividades em local salubre e em serviço não penoso ou perigoso.

#### **SUBSEÇÃO V**

##### **Da Gratificação pela Participação em Órgãos de Deliberação Coletiva**

Art. 61 - A Gratificação pela Participação em órgão de Deliberação Coletiva (jeton) é fixada, por ato do Governador do Estado, tendo em vista o princípio de hierarquia, a equivalência de funções e a complexidade das respectivas responsabilidades.

§ 1º - O servidor que, pela natureza das atribuições de seu cargo, for membro nato de um Conselho, não fará jus à gratificação de que trata este artigo.

§ 2º - É vedada a participação remunerada do servidor em mais de um órgão de deliberação coletiva.

§ 3º - A gratificação de que trata este artigo será, paga por sessão a que comparecerem os membros dos órgãos de deliberação coletiva e não poderá exceder a 04 (quatro) sessões ordinárias e, excepcionalmente, a 02 (duas) sessões extraordinárias, por mês.

#### **SUBSEÇÃO VI**

##### **Da Gratificação de Representação de Gabinete**

Art. 62 - A Gratificação de Representação de Gabinete será concedida aos servidores requisitados para servirem junto à Governadoria do Estado, à Vice-Governadoria e na estrutura básica do Serviço Social do Estado - SERSE.

§ 1º - A Gratificação de que trata este artigo será calculada mediante a aplicação do percentual de 100% (cem por cento) sobre o vencimento do cargo efetivo do servidor.

§ 2º - Na hipótese do servidor ocupar Cargo ou Função de Chefia e Assessoramento poderá optar pelo valor correspondente à remuneração do respectivo cargo ou função para o qual foi nomeado.

§ 3º - Em nenhum caso, o valor da gratificação poderá exceder à atribuída ao cargo em Comissão de maior símbolo.

§ 4º - A Gratificação, prevista neste artigo, não será incorporada ao vencimento, para qualquer efeito, nem poderá ser percebida, cumulativamente, com a gratificação pela prestação de serviços extraordinários.

### **SUBSEÇÃO VII**

#### **Da Gratificação de Controle Interno e Auditoria**

Art. 63 - A Gratificação de Controle Interno e Auditoria é devida aos servidores integrantes do Grupo Administração Financeira, Contabilidade e Auditoria, da Secretaria da Fazenda e será calculada sobre o vencimento do cargo, na forma e condições a serem estabelecidas pelo Chefe do Poder Executivo, em Regulamento.

### **SUBSEÇÃO VIII**

#### **Da Gratificação por Condições Especiais de Trabalho**

Art. 64 - A Gratificação por Condições Especiais de Trabalho será concedida com vistas ao interesse público de fixar o servidor em determinadas regiões, incentivá-lo no exercício de determinadas funções ou quando estas se realizarem em locais ou por meio e modos ou para fins especiais que reclamem tratamento especial.

Parágrafo Único - A Gratificação, de que trata este artigo, será fixada pelo Chefe do Poder Executivo, após ouvido o Conselho Estadual de Política Salarial, no modo e forma e nas circunstâncias definidas em Regulamentos.

### **Subseção IX**

#### **Do Adicional por Tempo de Serviço**

Art. 65 - O adicional por tempo de serviço é devido à razão de 3% (três por cento) por triênio de serviço público efetivo, incidente sobre o vencimento básico do cargo.

Parágrafo único - O servidor fará jus ao adicional, de que trata este artigo, a partir do mês em que completar o triênio.

## **SUBSEÇÃO X**

### **Do Adicional Noturno**

Art. 66 - O serviço noturno, prestado em horário compreendido entre 22 (vinte e duas) horas de um dia e 5 (cinco) horas do dia seguinte, terá o valor-hora acrescido de 100% (cem por cento) do valor-hora do vencimento básico do cargo.

## **SUBSEÇÃO XI**

### **Do Adicional de Férias**

Art. 67 - Independentemente de solicitação, será pago ao servidor por ocasião das férias, um adicional correspondente a 1/3 (um terço) da remuneração do período de férias.

Parágrafo Único - No caso de o servidor exercer função de Direção, Chefia ou Assessoramento, ou ocupar cargo em Comissão, a respectiva vantagem será considerada no cálculo adicional de que trata este artigo.

## **SUBSEÇÃO XII**

### **Do Adicional de Produtividade**

Art. 68 - O Adicional de Produtividade é devido, exclusivamente, ao servidor ocupante de cargo do Grupo Fisco/Tributação/Arrecadação e Procuradores Fiscais, da Secretaria da Fazenda.

§ 1º - É assegurado o Adicional de Produtividade aos ocupantes dos cargos, previstos neste artigo, quando nomeados pelo Governador do Estado para Cargo em Comissão, de Direção e Assessoramento Superior, ou quando, na Secretaria da Fazenda, exercerem Função de Direção Intermediária, Chefia, Assessoramento, Supervisão e Coordenação ou designados para atividades de arrecadações de tributos.

§ 2º - Não farão jus ao Adicional de Produtividade os servidores no exercício de outras atividades, não previstas neste artigo.

§ 3º - Os valores do Adicional de Produtividade, de que trata este artigo, a forma e as condições de sua percepção serão fixados por Decreto do Governador do Estado, não podendo ultrapassar a 15% (quinze por cento) do crescimento real da receita tributária estadual.<sup>32</sup>

## **SEÇÃO III**

### **Do Salário-Família**

Art. 69 - O salário-família é concedido ao servidor ativo ou inativo, por dependente econômico, no valor fixado em lei.

---

<sup>32</sup> Vide ADIN 1644-0 – Rel. Min. Sepúlveda Pertence. O STF, por maioria de votos, indeferiu o pedido de medida cautelar. Aguardando julgamento do mérito. Plenário 18.09.97.

§ 1º - O salário-família será devido a partir do mês em que o servidor se habilitar ao benefício.

§ 2º - Consideram-se dependentes econômicos para efeito da percepção do salário-família:

I - o cônjuge ou companheiro e os filhos, inclusive os enteados até 21 (vinte e um) anos de idade ou, se estudante, até 24 (vinte e quatro) anos de idade ou, se inválido, de qualquer idade;

II - o menor de 21 (vinte e um) anos que, mediante autorização judicial, viver na companhia e às expensas do servidor, ou do inativo;

III - a mãe e o pai sem economia própria.

Art. 70 - Não se configura a dependência econômica quando o beneficiário do salário-família perceber rendimento do trabalho ou de qualquer outra fonte, pensão ou provento da aposentadoria, em valor igual ou superior ao salário mínimo.

Parágrafo Único - O salário-família não está sujeito a qualquer desconto, ainda que para fim de previdência social.

Art. 71 - Quando o pai e a mãe forem servidores públicos e viverem em comum, o salário-família será pago a um deles; quando separados, será pago a um e outro, de acordo com a distribuição dos dependentes.

Parágrafo Único - Ao pai e à mãe equiparam-se o padrasto e a madrasta e, na falta destes, os representantes legais dos incapazes.

### **CAPÍTULO III**

#### **Das Férias**

Art. 72 - O servidor fará jus a 30 (trinta) dias consecutivos de férias, que podem ser acumuladas, até o máximo de 02 (dois) períodos, no caso de necessidade do serviço, ressalvados os casos em que haja legislação específica.

§ 1º - Para o primeiro período aquisitivo de férias serão exigidos 12 (doze) meses de exercício.

§ 2º - É vedado levar à conta de férias qualquer falta ao serviço.

§ 3º - O servidor exonerado do cargo efetivo, ou em comissão, perceberá indenização relativa ao período das férias a que tiver direito e ao incompleto, na proporção de 1/12 (um doze avos) por mês de efetivo exercício, ou fração superior a 14 (quatorze) dias.

§ 4º - A indenização será calculada com base na remuneração do mês em que for publicado o ato exoneratório.

Art. 73 - O servidor que opera direta e permanentemente com Raios X ou substâncias radioativas gozará 20 (vinte) dias consecutivos de férias, por quadrimestre de atividade profissional, proibida em qualquer hipótese a acumulação.

Art. 74 - As férias não poderão ser interrompidas, salvo motivo de superior interesse público e absoluta necessidade do serviço.

**CAPÍTULO IV**  
**Das licenças**  
**SEÇÃO I**  
**Disposições Gerais**

Art. 75 - Conceder-se-á ao servidor licença:

I - para tratamento de saúde;

II - por motivo de doença em pessoa da família;

III - por acidente em serviço;

IV - por motivo de afastamento do cônjuge ou companheiro;

V - para o serviço militar obrigatório;

VI - para atividade política;

VII - prêmio por assiduidade;

VIII - para tratar de interesses particulares;

IX - para desempenho de mandato classista.

X - à gestante e à paternidade.

§ 1º - Não se concederá licença para tratar de interesses particulares ao servidor ocupante de cargo em comissão ou em estágio probatório.

§ 2º - As licenças previstas nos incisos I, II e III dependem de perícia médica ou junta médica oficial e serão concedidas, pelo prazo indicado no laudo.

§ 3º - O servidor não poderá permanecer em licença da mesma espécie por período superior a 24 (vinte e quatro) meses, salvo nos casos dos incisos IV, V, VI e IX deste artigo.

§ 4º - É vedado o exercício de atividade remunerada durante o período da licença para tratamento da própria saúde ou de pessoa da família.

Art. 76 - A licença concedida dentro de 60 (sessenta) dias do término de outra da mesma espécie será considerada como prorrogação.



## **SEÇÃO II**

### **Da Licença para Tratamento de Saúde**

Art. 77 - Será concedida ao servidor licença para tratamento de saúde, a pedido ou de ofício, com base em pedido oficial, sem prejuízo da remuneração a que fizer jus.

Art. 78 - Para licença até 15 (quinze) dias, a inspeção poderá ser feita por médico do serviço social e, se por prazo superior, por junta médica.

§ 1º - Sempre que necessário, a inspeção médica será realizada na residência do servidor ou no estabelecimento hospitalar onde se encontrar internado.

§ 2º - Inexistindo médico do órgão oficial no local onde se encontra o servidor, será aceito atestado passado por médico particular, homologado pela junta médica.

Art. 79 - Findo o prazo da licença, o servidor deverá reassumir, imediatamente, o exercício, salvo prorrogação pedida antes de findar a licença ou se for o caso, pedir aposentadoria.

Art. 80 - O atestado e o laudo da junta médica não se referirão ao nome ou natureza da doença, salvo quando se tratar de lesões produzidas por acidente em serviço, moléstia profissional ou doença grave, contagiosa ou incurável, especificada em lei.

Art. 81 - O servidor que apresentar indícios de lesões orgânicas ou funcionais será submetido a inspeção médica.

Parágrafo Único - Constitui falta grave a recusa do servidor à inspeção médica.

## **SEÇÃO III**

### **Da Licença por Motivo de Doença em Pessoa da Família**

Art. 82 - Poderá ser concedida licença ao servidor por motivo de doença do cônjuge ou companheiro, padrasto ou madrasta, ascendente, descendente, enteado e colateral consanguíneo ou afim até o segundo grau civil, mediante comprovação por junta médica oficial.

§ 1º - A licença somente será deferida se a assistência direta do servidor for indispensável e não puder ser prestada simultaneamente com o exercício do cargo.

§ 2º - A licença será concedida sem prejuízo da remuneração do cargo efetivo, até 90 (noventa) dias, podendo ser prorrogada por mais até 90 (noventa) dias, mediante parecer da junta médica, e, excedendo estes prazos, sem remuneração.

## **SEÇÃO IV**

### **Da Licença por Acidente em Serviço**

Art. 83 - Será licenciado, com remuneração integral, o servidor acidentado em serviço ou acometido de moléstia profissional.

Art. 84 - Configura acidente em serviço ou doença profissional, o dano físico ou mental sofrido pelo servidor, que se relacione, mediata ou imediatamente, com as atribuições do cargo exercido.

Parágrafo único - Equipara-se ao acidente em serviço o dano:

I - decorrente de agressão sofrida e não provocada pelo servidor no exercício do cargo;

II - sofrido no percurso da residência para o trabalho e vice-versa.

Art. 85 - O servidor acidentado em serviço que necessite de tratamento especializado poderá ser tratado em instituição privada, à conta de recursos públicos.

Parágrafo Único - O tratamento recomendado por junta médica oficial constitui medida de exceção e somente será admissível quando inexisterem meios e recursos adequados em instituição pública.

Art. 86 - A prova do acidente será feita no prazo de 10 (dez) dias, prorrogável quando as circunstâncias o exigirem.

## **SEÇÃO V**

### **Da Licença por Motivo de Afastamento de Cônjuge ou Companheiro**

Art. 87 - Poderá ser concedida licença ao servidor para acompanhar cônjuge ou companheiro que foi deslocado para outro ponto do território nacional, para o exterior ou para o exercício de mandato eletivo dos Poderes Executivo e Legislativo.

§ 1º - A licença será por prazo indeterminado e sem remuneração.

§ 2º - Na hipótese do deslocamento de que trata este artigo, o servidor poderá ser lotado, provisoriamente, em repartição da Administração Pública do Estado, desde que para o exercício de atividade compatível com o seu cargo.

## **SEÇÃO VI**

### **Da Licença para o Serviço Militar**

Art. 88 - Ao servidor convocado para o serviço militar será concedida licença, na forma e condições previstas na legislação específica.

Parágrafo Único - Concluído o serviço militar, o servidor terá até 30 (trinta) dias sem remuneração para reassumir o exercício do cargo.

## **SEÇÃO VII**

### **Da Licença para Atividade Política**

Art. 89 - O servidor terá direito a licença, sem remuneração, durante o período que mediar entre a sua escolha em convenção partidária, como candidato a cargo eletivo, e a véspera do registro de sua candidatura perante a Justiça Eleitoral.

Parágrafo único - O servidor candidato a cargo eletivo na localidade onde desempenha suas funções e que exerça cargo de direção, chefia, assessoramento, arrecadação ou fiscalização, dele será afastado, a partir do dia imediato ao do registro de sua candidatura perante a Justiça Eleitoral, até o 15º (décimo quinto) dia seguinte ao do pleito.

Art. 90 - A partir do registro da candidatura e até o 15º (décimo quinto) dia seguinte ao da eleição, o servidor fará jus à licença remunerada como se em efetivo exercício estivesse.

## **SEÇÃO VIII**

### **Da Licença-Prêmio por Assiduidade**

Art. 91 - Após cada quinquênio ininterrupto de exercício, o servidor fará jus a 3 (três) meses de licença, que poderão ser acumuladas até o máximo de dois períodos, a título de prêmio por assiduidade, com a remuneração que percebia à data do seu afastamento.

§ 1º - Os períodos de licença-prêmio já adquiridos e não gozados pelo servidor que vier a falecer ou aposentar-se por invalidez serão convertidos em pecúnia, em favor de seus beneficiários da pensão, ou pago por ocasião da aposentadoria.

§ 2º - A autoridade deverá conceder a licença-prêmio dentro do prazo de até um ano, se requerida pelo servidor.

Art. 92 - Não se concederá licença-prêmio ao servidor que, no período aquisitivo:

I - sofrer penalidade disciplinar de suspensão;

II - afastar-se do cargo em virtude de:

a) licença por motivo de doença em pessoa da família, sem remuneração;

b) licença para tratar de interesses particulares;

c) condenação à pena privativa de liberdade por sentença definitiva;

d) afastamento para acompanhar cônjuge ou companheiro;

§ 1º - As faltas injustificadas ao serviço retardarão a concessão da licença-prêmio, na proporção de 1 (um) mês para cada falta.

§ 2º - VETADO.

Art. 93 - O número de servidores em gozo simultâneo de licença-prêmio não poderá ser superior a 1/3 (um terço) da lotação da respectiva unidade administrativa do órgão ou entidade.

## **SEÇÃO IX**

### **Da Licença para Tratar de Interesses Particulares**

Art. 94 - A critério da Administração, poderá ser concedida ao servidor estável licença para o trato de assuntos particulares, pelo prazo de até 2 (dois) anos consecutivos, sem remuneração.

§ 1º - A licença poderá ser interrompida, a qualquer tempo, a pedido do servidor ou do interesse do serviço.

§ 2º - Não se concederá nova licença antes de decorridos 2 (dois) anos do término da anterior.

§ 3º - Não se concederá licença a servidores nomeados, removidos, redistribuídos ou transferidos, antes de completarem 2 (dois) anos de exercício.

## **SEÇÃO X**

### **Da Licença para Desempenho de Mandato Classista**

Art. 95 – É assegurado ao servidor o direito à licença para desempenho de mandato em confederação, federação, associação de classe de âmbito estadual, sindicato representativo da categoria, central sindical ou entidade fiscalizadora da profissão, com remuneração do cargo efetivo.

§ 1º - Somente poderão ser licenciados servidores eleitos para cargos de direção ou representação nas referidas entidades, obedecendo os seguintes critérios:

I - Confederação, Federação, Central Sindical e Associação de Classe terão no máximo 03 (três) liberações por entidade, sendo que a Associação de Classe deverá ter no mínimo 200 associados.

II - Ao Sindicato de Classe ficam assegurados 03 (três) liberações por entidade, mais 01 (um) para cada 500 (quinhentos) servidores na base da categoria no limite máximo de 30 (trinta) liberados.

§ 2º - A licença terá duração igual a do mandato sendo automaticamente prorrogada em caso de reeleição.

## **SEÇÃO XI**

### **Da Licença à Gestante e à Paternidade**

Art. 96 - Será concedida licença à servidora gestante por 120 (cento e vinte) dias consecutivos, sem prejuízo da remuneração.

§ 1º - A licença, com início no primeiro dia do nono mês de gestação, poderá ser antecipada por prescrição médica.

§ 2º - No caso de nascimento prematuro, a licença terá início a partir do parto.

§ 3º - No caso de natimorto ou aborto, atestado por médico oficial, a servidora terá 30 (trinta) dias de licença remunerada a partir do evento.

Art. 97 - Pelo nascimento de filhos, o servidor terá direito à licença paternidade de 5 (cinco) dias úteis a partir do parto do cônjuge ou da companheira.

Art. 98 - À servidora que adotar ou obtiver guarda judicial de criança até 1 (um) ano de idade, serão concedidos 90 (noventa) dias de licença remunerada, e, se de mais de 1 (um) ano, a licença remunerada será de 30 (trinta) dias.

Art. 99 - Para amamentar o próprio filho, até a idade de seis meses, a servidora lactante terá direito, durante a jornada de trabalho, a duas horas de descanso que poderá ser parcelada em dois períodos de uma hora.

## **CAPÍTULO V**

### **Do Afastamento para servir a outro Órgão ou Entidade**

#### **SEÇÃO I**

#### **Das Disposições Gerais**

Art. 100 - O servidor poderá ser cedido para ter exercício em outro órgão ou entidade da Administração Direta, Indireta ou fundacional, mediante pedido fundamentado pela autoridade requisitante, sob pena de nulidade.

§ 1º - Excetua-se dos dispositivos deste artigo, as requisições para a Governadoria do Estado e as nomeações para cargos em comissão e dos dirigentes de entidades administrativas de nomeação pelo Governador do Estado ou de eleição pela assembléia geral.

§ 2º - As disposições de servidores, no âmbito da Administração Pública, far-se-ão sempre com ônus para o órgão requisitante, salvo nos casos de servidores nomeados para cargos de confiança ou de solicitação para ocupar cargo de Secretário de Município.

§ 3º - As disposições serão concedidas pelo prazo de 1 (um) ano, prorrogável, por necessidade do serviço, por igual período, exceto os casos previstos no parágrafo primeiro deste artigo e as requisições para os Poderes Executivo, Judiciário e Legislativo da União, dos Estados do Distrito Federal e Secretarias de Municípios, cujo prazo será o do tempo da serventia.

#### **SEÇÃO II**

#### **Dos Afastamentos do Servidor Público**

Art. 101 - A disposição de servidor entre órgãos da Administração Direta, Indireta e Fundacionais somente poderá ocorrer quando tenha por finalidade o exercício de cargo em comissão ou de direção superior das entidades administrativas e, excepcionalmente, o exercício de função técnica ou científica, recaindo, neste último caso o ônus para o órgão requisitante.

Art. 102 - No interesse do serviço, será permitido o afastamento do servidor para exercer função de chefia, direção e assessoramento intermediários, desde que compatível com sua formação técnica ou científica.

Parágrafo Único - O afastamento do servidor, no caso deste artigo, vigorará pelo tempo de sua serventia.

Art. 103 - Ao servidor em exercício de mandato eletivo aplicam-se as seguintes disposições:

I - tratando-se de mandato eletivo federal, estadual ou distrital, ficará afastado de seu cargo;

II - investido no mandato de Prefeito, será afastado do cargo, sendo-lhe facultado optar pela sua remuneração;

III - investido no mandato de Vereador:

a) - havendo compatibilidade de horário, perceberá as vantagens de seu cargo, sem prejuízo da remuneração do cargo eletivo;

b) - não havendo compatibilidade de horário, será afastado do cargo, sendo-lhe facultado optar pela sua remuneração.

IV - em qualquer caso que exija o afastamento para o exercício de mandato eletivo, seu tempo de serviço será contado para todos os efeitos legais, exceto para promoção por merecimento.

V - no caso de afastamento do cargo, o servidor contribuirá para a previdência social como se em exercício estivesse.

VI - investido em mandato eletivo ou classista, o servidor não poderá ser removido, transferido ou redistribuído, de ofício, para localidade diversa daquela onde exerce o mandato.

Art. 104 - O servidor não poderá ausentar-se do Estado para estudo ou missão especial, sem autorização do Chefe do Poder a que está vinculado.

§ 1º - A ausência não excederá a 4 (quatro) anos, e finda a missão ou estudo, somente decorrido igual período, será permitida nova ausência.

§ 2º - Ao servidor beneficiado pelo disposto neste artigo não será concedida exoneração ou licença para tratar de interesse particular antes de decorrido período igual ao do afastamento, ressalvada a hipótese de ressarcimento da despesa havida com seu afastamento.

Art. 105 - A critério da Administração, poderá ser concedida ao servidor estável bolsa-de-estudo, fora do Estado, para fins de cursos de pós-graduação, aperfeiçoamento, extensão e pesquisa; por prazo de até 2 (dois) anos prorrogável por igual período, conforme exigirem as circunstâncias, devidamente comprovadas.

§ 1º - É vedada a concessão de bolsa-de-estudo para a formação profissional e outros cursos existentes no Estado, inclusive os previstos neste artigo.

§ 2º - O valor da bolsa-de-estudo não poderá ultrapassar à remuneração do cargo do servidor.

## **CAPÍTULO VI** **Das Concessões**

Art. 106 - Sem qualquer prejuízo e considerado de efetivo exercício, poderá o servidor ausentar-se do serviço:

I - por 1 (um) dia, para doação de sangue;

II - por 2 (dois) dias, para se alistar como eleitor;

III - por 8 (oito) dias consecutivos em razão de:

a) casamento;

b) falecimento do cônjuge, companheiro, pais, madrasta ou padrasto, filhos, enteados, menor sob guarda ou tutela, irmãos ou pessoas que vivem sob sua dependência econômica.

Art. 107 - Será concedido horário especial ao servidor estudante, quando comprovada a incompatibilidade entre o horário escolar e o da repartição sem prejuízo do exercício do cargo.

§ 1º - Para efeito do disposto neste artigo, será exigida a compensação de horário na repartição, respeitada a duração semanal do trabalho.

§ 2º - Ao servidor estudante que mudar de sede no interesse da administração, é assegurada, na localidade da nova residência, matrícula em instituição de ensino congênere estadual, em qualquer época, independentemente de vaga.

§ 3º - O disposto no § 2º deste artigo é extensivo ao cônjuge ou companheiro, aos filhos e àqueles que vivam na sua dependência econômica.

## **CAPÍTULO VII** **Do Tempo de Serviço**

Art. 108 - A apuração do tempo de serviço será feita em dias, convertidos em anos, considerado o ano como de 365 (trezentos e sessenta e cinco) dias.

Parágrafo Único - Feita a conversão, os dias restantes, até 182 (cento e oitenta e dois), não serão computados, arredondando-se para um ano quando excederem este número, para efeito de aposentadoria.

Art. 109 São considerados como de efetivo exercício os afastamentos em virtude de:

I - férias;

II - exercício de cargo em comissão em qualquer dos Poderes do Estado e nos serviços da União, dos Estados, dos Municípios, e do Distrito Federal;

III - desempenho de mandato eletivo federal, estadual, municipal ou distrital e atividade política, na forma do art. 89, exceto para promoção por merecimento;

IV - júri, serviço militar e outros serviços obrigatórios por lei;

V – disposição, regularmente concedida, para prestar serviços nos órgãos e entidades da Administração Direta, Indireta e Fundações do Estado, e afastamento para bolsas-de-estudos;

VI - licença:

a) à gestante, à adotante e à paternidade;

b) para tratamento da própria saúde até 2 (dois) anos;

c) para o desempenho de mandato classista, exceto para efeito de promoção por merecimento;

d) por motivo de acidente em serviço ou doença profissional;

e) prêmio por assiduidade.

VII - deslocamento para a nova sede;

VIII - participação em competição desportiva, congressos e outras atividades culturais devidamente autorizada;

IX - disponibilidade;

X - prisão do servidor, quando absolvido por sentença definitiva;

Art. 110 - Contar-se-á apenas para efeito de aposentadoria e disponibilidade:

I - o tempo de serviço público prestado à União, Estados, Municípios e Distrito Federal;

II - licença para tratamento de saúde de pessoa da família do servidor, com remuneração;

III - o tempo correspondente ao desempenho de mandato eletivo, anterior ao ingresso no serviço público;

IV - o tempo de serviço prestado na atividade privada, condicionado à compensação financeira, na forma do art. 202, § 2º da Constituição Federal.

Art. 111 – É vedada a contagem cumulativa de tempo de serviço prestado concomitantemente em mais de um cargo ou função de órgãos ou entidades dos Poderes da



União, Estado, Município e Distrito Federal e suas entidades da administração indireta e fundacionais.

## **CAPÍTULO VIII**

### **Do Direito de Petição**

Art. 112 – É assegurado ao servidor o direito de requerer ao Poder Público em defesa de direito ou interesse legítimo.

Art. 113 - O requerimento será dirigido à autoridade competente para decidi-lo e encaminhado por intermédio daquela a que estiver imediatamente subordinado o requerente.

Art. 114 - Cabe pedido de reconsideração a autoridade que houver expedido o ato ou proferido a primeira decisão, não podendo ser renovado.

Parágrafo Único - O requerimento e o pedido de reconsideração de que tratam os artigos anteriores deverão ser despachados no prazo de 5 (cinco) dias e decididos dentro de 30 (trinta) dias.

Art. 115 - Caberá recurso:

I - do indeferimento do pedido de reconsideração;

II - das decisões sobre os recursos sucessivamente interpostos.

Parágrafo único - O recurso será dirigido à autoridade imediatamente superior à que tiver expedido o ato ou proferido a decisão, e, sucessivamente, em escala ascendente, às demais autoridades, sempre por intermédio da sua chefia imediata.

Art. 116 - O prazo para interposição do pedido de reconsideração ou de recurso é de 60 (sessenta) dias, a contar da publicação ou da ciência, pelo interessado, da decisão recorrida.

§ 1º - O recurso poderá ser recebido, com efeito suspensivo, a juízo da autoridade competente.

§ 2º - Em caso de provimento do pedido de reconsideração ou do recurso, os efeitos da decisão retroagirão à data do ato impugnado.

Art. 117 - Para o exercício do direito de petição, é assegurada vista do processo ou documento, na repartição, ao servidor ou a procurador por ele constituído.

Art. 118 - A administração deverá rever seus atos, a qualquer tempo, quando eivados de ilegalidades.

Art. 119 - São fatais e improrrogáveis os prazos estabelecidos neste Capítulo, salvo motivo de força maior.

Art. 120 - O direito de requerer prescreve:

I - em 5 (cinco) anos, quanto aos atos de demissão e de cassação de aposentadoria ou disponibilidade, ou que afetem interesse patrimonial e créditos resultantes das relações de trabalho;

II - em 180 (cento e oitenta) dias, nos demais casos, salvo quando outro prazo for fixado em lei.

§ 1º - O prazo da prescrição será contado da data da publicação do ato impugnado ou da data da ciência pelo interessado, quando o ato não for publicado.

§ 2º - O pedido de reconsideração e o recurso, quando cabíveis, interrompem a prescrição.

§ 3º - A prescrição é de ordem pública, não podendo ser relevada pela administração.

**CAPÍTULO IX**  
**Da Pensão e da Aposentadoria**  
**SEÇÃO I**  
**Da Pensão**

Art. 121 - Por morte do servidor, os dependentes fazem jus a uma pensão mensal de valor correspondente ao da respectiva remuneração ou provento, a partir da data do óbito, observadas as normas da entidade previdenciária.

Parágrafo Único - Em nenhuma hipótese, o valor da pensão será superior ou inferior ao da remuneração ou proventos do servidor e ao salário-de-contribuição previdenciário.

Art. 122 - As pensões distinguem-se, quanto à natureza, em vitalícias e temporárias.

§ 1º - A pensão vitalícia é composta de cota ou cotas permanentes, que somente se extinguem ou reverterem com a morte de seus beneficiários.

§ 2º - A pensão temporária é composta de cota ou cotas que podem se extinguir ou reverter por motivo de morte, cessação de invalidez ou maioridade do beneficiário.

Art. 123 - São beneficiários das pensões:

I - vitalícia:

a) o cônjuge;

b) a pessoa desquitada, separada judicialmente ou divorciada, com direito de perceber pensão alimentícia;

c) o companheiro ou companheira designado que comprove união estável como entidade familiar;

d) a mãe e o pai que comprovem dependência econômica do servidor;

e) a pessoa designada, maior de 60 (sessenta) anos e a pessoa portadora de deficiência, que vivam sob a dependência do servidor.

II - temporária:

a) os filhos, ou enteados, até 21 (vinte e um) anos de idade, ou, se inválido, enquanto perdurar a invalidez:

b) menor sob guarda ou tutela até 21 (vinte e um) anos de idade;

c) a irmã ou irmão órfão, até 21 (vinte e um) anos, e o inválido, enquanto durar a invalidez, que comprovem dependência econômica do servidor;

d) a pessoa designada que viva na dependência econômica do servidor, até 21 (vinte e um) anos, ou, se inválida, enquanto durar a invalidez.

§ 1º - A concessão de pensão vitalícia aos beneficiários de que tratam as alíneas "a" e "b" do inciso I deste artigo exclui desse direito os demais beneficiários referidos nas alíneas "d" e "e".

§ 2º - A concessão da pensão temporária aos beneficiários de que tratam as alíneas "a" e "b" do inciso II deste artigo exclui desse direito os beneficiários referidos nas alíneas "c" e "d".

§ 3º - O limite de idade estabelecido nas alíneas acima, deste artigo, será prorrogado até o limite de 24 (vinte e quatro) anos mediante comprovação de matrícula e frequência em instituição de ensino oficial ou reconhecida.

Art. 124 - A pensão será concedida integralmente ao titular da pensão vitalícia, exceto se existirem beneficiários da pensão temporária.

§ 1º - Ocorrendo habilitação de vários titulares à pensão vitalícia, o seu valor será distribuído em partes iguais entre os beneficiários habilitados.

§ 2º - Ocorrendo habilitação às pensões vitalícias e temporária, metade do valor caberá ao titular ou titulares da pensão vitalícia, sendo a outra metade rateada em partes iguais, entre os titulares da pensão temporária.

§ 3º - Ocorrendo habilitação somente à pensão temporária, o valor integral da pensão será rateado, em partes iguais, entre os que se habilitarem.

Art. 125 - A pensão poderá ser requerida a qualquer tempo, prescrevendo tão somente as prestações exigíveis há mais de 5 (cinco) anos.

Parágrafo Único - Concedida a pensão, qualquer prova posterior ou habilitação tardia, que implique exclusão de beneficiário ou redução da pensão, só produzirá efeitos a partir da data em que for oferecida e comprovada.

Art. 126 - Não faz jus à pensão o beneficiário condenado pela prática de crime doloso de que tenha resultado a morte do servidor.

Art. 127 - Será concedida pensão provisória por morte presumida do servidor, quando declarada a ausência pela autoridade judiciária competente.

Parágrafo Único - A pensão provisória será transformada em vitalícia ou temporária, conforme o caso, decorridos 5 (cinco) anos de sua vigência, ressalvado o eventual reaparecimento do servidor, hipótese em que o benefício será automaticamente cancelado.

Art. 128 - Acarreta perda da qualidade de beneficiário:

I - o seu falecimento;

II - a anulação do casamento, quando a decisão ocorrer após a concessão da pensão ao cônjuge;

III - a cessação de invalidez, em se tratando de beneficiário inválido;

IV - a maioridade de filho, irmã ou irmão órfão ou pessoa designada, aos 21 (vinte e um) anos de idade, ressalvada a hipótese prevista no parágrafo 3º do art. 123, desta Lei Complementar;

V - a acumulação indevida de pensão;

VI - a renúncia expressa.

Art. 129 - Por morte ou perda da qualidade de beneficiário, a respectiva cota reverterá:

I - da pensão vitalícia para os remanescentes desta pensão ou para os titulares da pensão temporária, se não houver pensionista remanescente da pensão vitalícia;

II - da pensão temporária para os co-beneficiários ou, na falta destes, para o beneficiário da pensão vitalícia.

Art. 130 - As pensões serão automaticamente atualizadas na mesma data e na mesma proporção dos reajustes dos servidores em atividade.

Art. 131 - Ressalvado o direito de opção, é vedada a percepção cumulativa de mais de duas pensões.

## **SEÇÃO II** **Da Aposentadoria**

Art. 132 - O servidor público será aposentado:

I - por invalidez permanente, com proventos integrais, quando decorrente de acidente em serviço, moléstia profissional ou doença grave, contagiosa ou incurável, especificadas em lei, e proporcionais nos demais casos;

II - compulsoriamente, aos 70 (setenta) anos de idade, com proventos proporcionais ao tempo de serviço;

III - voluntariamente:

a) aos 35 (trinta e cinco) anos de serviço, se homem, e aos 30 (trinta) anos se mulher, com proventos integrais;

b) aos 30 (trinta) anos de efetivo exercício em funções de magistério, se homem, e 25 (vinte e cinco) anos, se mulher, com proventos integrais;

c) aos 30 (trinta) anos de serviço, se homem, e aos 25 (vinte e cinco) anos, se mulher, com proventos proporcionais ao tempo de serviço;

d) aos 65 (sessenta e cinco) anos de idade, se homem, e aos 60 (sessenta) anos, se mulher, com proventos proporcionais ao tempo de serviço.

e) aos 30 (trinta) anos de serviço, se homem, e 25 (vinte e cinco) anos, se mulher, em exercício de atividades consideradas penosas, insalubres ou perigosas, com vencimentos integrais.

Art. 133 - A aposentadoria compulsória será automática, e declarada por ato, com vigência a partir do dia imediato àquele em que o servidor atingir a idade-limite de permanência no serviço ativo.

Art. 134 - A aposentadoria voluntariamente ou por invalidez vigorará a partir da data da publicação do respectivo ato.

§ 1º - A aposentadoria por invalidez será precedida de licença para tratamento de saúde, por período não excedente a 24 (vinte e quatro) meses.

§ 2º - Expirado o período de licença e não estando em condições de reassumir o cargo ou de ser readaptado, o servidor será aposentado.

§ 3º - O lapso de tempo compreendido entre o término da licença e a publicação do ato da aposentadoria será considerado como de prorrogação da licença.

Art. 135 - O provento da aposentadoria será calculado com base no vencimento do cargo efetivo, acrescido das vantagens de caráter permanente, previstos em lei, e revisto na mesma data e na mesma proporção, sempre que se modificar a remuneração dos servidores em atividade.

§ 1º - São estendidos aos inativos quaisquer benefícios ou vantagens posteriormente concedidos aos servidores em atividade, mesmo quando decorrentes de transformação, extinção ou reclassificação de cargo ou função em que se deu a aposentadoria.

§ 2º - O valor dos proventos da aposentadoria será calculado com rigorosa observância do limite estabelecido pelo § 1º, do art. 40, desta Lei Complementar.

Art. 136 - O servidor que tiver exercido função de direção, chefia, assessoramento, assistência, cargo em comissão ou função gratificada, por período de 5 (cinco) anos consecutivos, ou 10 (dez) anos interpolados, poderá aposentar-se com a gratificação da função ou da gratificação do cargo em comissão, de maior valor, desde que exercido por um período mínimo de 2 (dois) anos.

Parágrafo Único - Quando o exercício da função ou cargo em comissão de maior valor não corresponder ao período de 2 (dois) anos, será incorporada a gratificação ou remuneração da função ou cargo em comissão imediatamente inferior dentre os exercidos.

**TÍTULO IV**  
**Do Regime Disciplinar**  
**CAPÍTULO I**  
**Dos Deveres do Servidor**

Art. 137 - São deveres do servidor público:

I - exercer com dignidade, zelo e dedicação as atribuições de seu cargo;

II - ser leal às instituições a que servir;

III - observar as normas legais e regulamentares;

IV - cumprir, com presteza, as ordens superiores, exceto quando manifestamente ilegais;

V - atender com presteza:

a) ao público em geral, prestando as informações solicitadas, ressalvadas as protegidas por sigilo;

b) à expedição de certidões requeridas para defesa de direito ou esclarecimento de situações de interesse pessoal, no prazo máximo de 10 (dez) dias;

c) às requisições para a defesa da Fazenda Pública;

VI - levar ao conhecimento da autoridade imediatamente superior as irregularidades de que tiver ciência em razão do cargo;

VII - zelar pela economia do material e a conservação do patrimônio público;

VIII - guardar sigilo sobre assunto da repartição;

IX - manter conduta compatível com a moralidade pública;

X - ser assíduo e pontual ao serviço;

XI - tratar com urbanidade as pessoas;

XII - representar contra ilegalidade, omissão ou abuso de poder;

**CAPÍTULO II**  
**Das Proibições**

Art. 138 - Ao servidor é proibido:

- I - ausentar-se do serviço durante o expediente, sem prévia autorização do chefe imediato;
- II - retirar sem prévia anuência da autoridade competente, qualquer documento ou objeto da repartição;
- III - recusar fé a documentos públicos;
- IV - retardar andamento de documento e processo ou execução de serviço, deixar de praticar, indevidamente, ato de ofício, ou praticá-lo contra disposição expressa de lei, para satisfazer interesse pessoal;
- V - promover manifestação de apreço ou despreço no recinto da repartição;
- VI - cometer a pessoa estranha à repartição, fora dos casos previstos em lei, o desempenho de atribuição que seja de sua responsabilidade ou de seu subordinado, inclusive a outro servidor atribuições estranhas ao cargo que ocupa, exceto em situações de emergência e transitórias;
- VII - coagir ou aliciar subordinados no sentido de filiareem-se ou desfiliarem-se a associação profissional ou sindical, ou a partido político;
- VIII - manter sob sua chefia imediata, em cargo ou função de confiança, cônjuge, companheiro ou parente até o segundo grau civil;
- IX - valer-se do cargo para lograr proveito pessoal ou de outrem, em detrimento da dignidade da função pública;
- X - participar de gerência ou administração de empresa privada, sociedade comercial ou exercer o comércio, exceto na qualidade de acionista, cotista ou comanditário;
- XI - atuar como procurador ou intermediário, junto a repartição pública, salvo quando se tratar de benefícios previdenciários ou assistenciais de parentes até o segundo grau, e de cônjuge ou companheiro;
- XII - receber propina, comissão, presente ou vantagem de qualquer espécie, exigir vantagem indevida para si ou para outrem, em razão de suas atribuições;
- XIII - praticar usura sob qualquer de suas formas;
- XIV - proceder de forma desidiosa;
- XV - utilizar pessoal ou recursos materiais da repartição em serviços ou atividades particulares;
- XVI - referir-se de modo depreciativo em informação, parecer ou despacho às autoridades e a atos da administração pública, podendo, porém, em trabalho assinado, criticá-los do ponto de vista doutrinário ou da organização do serviço;
- XVII - exercer quaisquer atividades que sejam incompatíveis com o exercício do cargo ou função e com o horário de trabalho;

XVIII - aceitar comissão, emprego ou pensão de estado estrangeiro.

### **CAPÍTULO III**

#### **Da Acumulação**

Art. 139 – É vedada a acumulação remunerada de cargos públicos, ressalvados os casos previstos na Constituição Federal.

Art. 140 - A acumulação de cargos, ainda que lícita, fica condicionada à comprovação da compatibilidade de horários.

Art. 141 - O servidor não poderá exercer mais de um cargo em comissão ou função gratificada, nem participar, remunerado, de mais de um órgão de deliberação coletiva.

Parágrafo Único - O servidor que acumular licitamente 2 (dois) cargos efetivos, quando investido em cargo de provimento em comissão, ficará afastado de ambos os cargos efetivos.

### **CAPÍTULO IV**

#### **Das Responsabilidades**

Art. 142 - Pelo exercício irregular de suas atribuições, o servidor responde civil, penal e administrativamente.

Art. 143 - A responsabilidade civil decorre de ato omissivo ou comissivo, doloso ou culposo, que resulte em prejuízo ao erário ou a terceiros.

§ 1º - A indenização de prejuízo dolosamente causado ao erário somente será liquidado na forma prevista no art. 42, § 3º, na falta de outros bens que assegurem a execução do débito pela via judicial.

§ 2º - Tratando-se de dano causado a terceiros, responderá o servidor perante a Fazenda Pública, em ação regressiva.

Art. 144 - A responsabilidade penal abrange os crimes e contravenções imputados ao servidor, nessa qualidade.

Art. 145 - A responsabilidade administrativa resulta de atos ou omissões praticados no desempenho do cargo ou função.

Art. 146 - As cominações civis, penais e disciplinares poderão cumular-se, sendo independentes entre si.

Art. 147 - A responsabilidade administrativa do servidor será afastada no caso de absolvição criminal que negue a existência do fato ou sua autoria.



## **CAPÍTULO V**

### **Das Penalidades**

Art. 148 - São penalidades disciplinares:

I - advertência;

II - suspensão;

III - demissão;

IV - cassação de aposentadoria ou disponibilidade;

V - destituição de cargo em comissão;

VI - destituição de função gratificada.

Art. 149 - Na aplicação das penalidades serão consideradas a natureza e a gravidade da infração cometida, os danos que dela provierem para o serviço público, as circunstâncias agravantes ou atenuantes e os antecedentes funcionais.

Art. 150 - A advertência será aplicada por escrito, nos casos de violação de proibição constante do art. 138 incisos I, II, III, V, VI, VII e VIII e de inobservância de dever funcional previsto em lei, regulamentação ou norma interna, que não justifique imposição de penalidade mais grave.

Art. 151 - A suspensão será aplicada em caso de reincidência das faltas punidas com advertência e de violação das demais proibições que não tipifiquem infração sujeita a penalidade de demissão, não podendo exceder de 90 (noventa) dias.

§ 1º - Será punido com suspensão de até 15 (quinze) dias o servidor que, injustificadamente, recusar-se a ser submetido a inspeção médica determinada pela autoridade competente, cessando os efeitos da penalidade uma vez cumprida a determinação.

§ 2º - Quando houver conveniência para o serviço, a penalidade de suspensão poderá ser convertida em multa, na base de 50% (cinquenta por cento) por dia de vencimento ou remuneração, ficando o servidor obrigado a permanecer em serviço.

Art. 152 - As penalidades de advertência e de suspensão terão seus registros cancelados, após o decurso de 3 (três) e 5 (cinco) anos de efetivo exercício, respectivamente, se o servidor não houver, nesse período, praticado nova infração disciplinar.

Parágrafo único - O cancelamento da penalidade não surtirá efeitos retroativos.

Art. 153 - A demissão será aplicada nos seguintes casos:

I - crime contra a administração pública;

II - abandono de cargo;

III - inassiduidade habitual;

IV - improbidade administrativa;

V - incontinência pública e conduta escandalosa na repartição;

VI - insubordinação grave em serviço;

VII - ofensa física, em serviço, a servidor ou a particular, salvo em legítima defesa própria ou de outrem;

VIII - aplicação irregular de dinheiros públicos;

IX - revelação de informação sigilosa do qual se apropriou em razão do cargo;

X - lesão aos cofres públicos e dilapidação do patrimônio estadual;

XI - corrupção;

XII - acumulação ilegal de cargos, empregos ou funções públicas;

XIII - transgressão dos incisos IV, IX, X, XI, XII, XV e XVII do art. 138, desta Lei Complementar.

Art. 154 - Verificada em processo disciplinar acumulação proibida e provada a boa-fé, o servidor optará por um dos cargos.

§ 1º - Provada a má-fé, perderá também o cargo que exercia há mais tempo e restituirá o que tiver percebido indevidamente.

§ 2º - Na hipótese do parágrafo anterior, sendo um dos cargos, emprego ou função exercido em outro órgão ou entidade, a demissão lhe será comunicada.

Art. 155 - Será cassada a aposentadoria ou a disponibilidade do inativo que houver praticado, na atividade, falta punível com a demissão.

Art. 156 - A destituição de cargo em comissão exercido por não ocupante de cargo efetivo será aplicada nos casos de infração sujeita às penalidades de suspensão e de demissão.

Parágrafo único - Constatada a hipótese de que trata este artigo, a exoneração efetuada nos termos do art. 34 será convertida em destituição de cargo em comissão.

Art. 157 - A demissão ou a destituição de cargo em comissão, nos casos dos incisos IV, VIII, X e XI do art. 153 implica a indisponibilidade dos bens e o ressarcimento ao erário, sem prejuízo da ação penal cabível.

Art. 158 - A demissão, ou a destituição de cargo em comissão por infringência do art. 138, incisos IX a XII incompatibiliza o ex-servidor para nova investidura em cargo público estadual, pelo prazo de 5 (cinco) anos.

Parágrafo único - Não poderá retornar ao serviço público estadual o servidor que for demitido ou destituído do cargo em comissão por infringência do art. 153 incisos I, IV, VIII, X e XI.

Art. 159 - Configura abandono de cargo a ausência intencional do servidor ao serviço por mais de 30 (trinta) dias consecutivos.

Art. 160 - Entende-se por inassiduidade habitual a falta ao serviço, sem causa justificada, por 60 (sessenta) dias, interpoladamente, durante o período de 12 (doze) meses.

Art. 161 - O ato de imposição da penalidade mencionará sempre o fundamento legal e a causa da sanção disciplinar.

Art. 162 - As penalidades disciplinares serão aplicadas:

I - pelo Governador do Estado ou, conforme o caso, pela autoridade referida no parágrafo único do art. 7º quando se tratar de demissão, cassação de aposentadoria ou disponibilidade de servidor, inclusive das autarquias e fundações do Estado;

II - pelos Secretários de Estado, dirigentes de órgãos e das autarquias e fundações do Estado, quando se tratar de suspensão superior a 30 (trinta) dias e destituição de função;

III - pelo chefe da repartição e autoridades administrativas de hierarquias imediatamente inferior àquelas mencionadas no inciso anterior, nos casos de advertência ou de suspensão de até 30 (trinta) dias.

Art. 163 - A ação disciplinar prescreverá:

I - em 5 (cinco) anos, quanto às infrações puníveis com demissão, cassação de aposentadoria ou disponibilidade e destituição de cargos em comissão.

II - em 2 (dois) anos, quanto à suspensão;

III - em 180 (cento oitenta) dias, quanto à advertência.

§ 1º - O prazo de prescrição começa a correr da data em que o fato se tornou conhecido.

§ 2º - Os prazos de prescrição previstos na lei penal aplicam-se às infrações disciplinares capituladas também como crime.

§ 3º - A abertura de sindicância ou a instauração de processo disciplinar interrompe a prescrição, até a decisão final proferida por autoridade competente.

**TÍTULO V**  
**Do Processo Administrativo Disciplinar**  
**CAPÍTULO I**  
**Das Disposições Gerais**

Art. 164 - A autoridade que tiver ciência de irregularidade no serviço público é obrigado a promover a sua apuração imediata, mediante sindicância ou processo administrativo disciplinar, assegurada ao acusado ampla defesa.

Art. 165 - As denúncias sobre irregularidades serão objetos de apuração, desde que contenham a identificação e o endereço do denunciante e sejam formuladas por escrito, confirmada a autenticidade.

Parágrafo Único - Quando o fato narrado não configurar evidente infração disciplinar ou ilícito penal, a denúncia será arquivada, por falta de objeto.

Art. 166 - Da sindicância poderá resultar:

I - arquivamento do processo;

II - aplicação de penalidade de advertência ou suspensão até 30 (trinta) dias;

III - instauração de processo disciplinar;

Parágrafo único - O prazo para conclusão da sindicância não excederá 30 (trinta) dias, podendo ser prorrogado por igual período, a critério da autoridade superior.

Art. 167 - Sempre que o ilícito praticado pelo servidor ensejar a imposição de penalidade de suspensão por mais de 30 (trinta) dias, será obrigatória a instauração de processo disciplinar.

## **CAPÍTULO II**

### **Do Afastamento Preventivo**

Art. 168 - Como medida cautelar e a fim de que o servidor não venha a influir na apuração da irregularidade, a autoridade instauradora do processo disciplinar poderá determinar o afastamento do exercício do cargo, pelo prazo de até 60 (sessenta) dias, será prejuízo da remuneração.

Parágrafo Único - O afastamento poderá ser prorrogado por igual prazo, findo o qual cessarão os seus efeitos, ainda que não concluído o processo.

## **CAPÍTULO III**

### **Do Processo Disciplinar**

Art. 169 - O processo disciplinar é o instrumento destinado a apurar responsabilidade de servidor por infração praticada no exercício de suas atribuições, ou que tenha relação com as atribuições do cargo em que se encontre investido.

Art. 170 - O processo disciplinar será conduzido por comissão composta de 3 (três) servidores estáveis, de cargo igual, equivalente ou superior ao do indiciado, designados pela autoridade competente, que indicará, dentre eles, o seu presidente.

§ 1º - A Comissão terá como secretário servidor designado pelo seu presidente, podendo a indicação recair em um de seus membros.

§ 2º - Não poderá participar de comissão de sindicância ou de inquérito, cônjuge, companheiro ou parente do acusado, consanguíneo ou afim, em linha reta ou colateral, até o terceiro grau.

Art. 171 - A Comissão exercerá suas atividades com independência e imparcialidade, assegurado o sigilo necessário à elucidação do fato ou exigido pelo interesse da administração.

Parágrafo único - As reuniões e as audiências das comissões terão caráter reservado.

Art. 172 - O processo disciplinar se desenvolve nas seguintes fases:

I - instauração, com a publicação do ato que constituir a comissão;

II - inquérito administrativo, que compreende instrução, defesa e relatório;

III - julgamento.

Art. 173 - O prazo para a conclusão do processo disciplinar não excederá 60 (sessenta) dias, contados da data da publicação do ato que constituir a comissão, admitida a sua prorrogação por igual prazo, quando as circunstâncias o exigirem.

§ 1º - Sempre que necessário, a comissão dedicará tempo integral aos seus trabalhos, ficando seus membros dispensados do ponto, até a entrega do relatório final.

§ 2º - As reuniões da comissão serão registradas em atas, que deverão detalhar as ocorrências e as deliberações adotadas.

## **SEÇÃO I** **Do Inquérito**

Art. 174 - O inquérito administrativo obedecerá ao princípio do contraditório, assegurada ao acusado ampla defesa, com a utilização dos meios e recursos admitidos em direito.

Art. 175 - Os autos da sindicância integrarão o processo disciplinar, como peça informativa da instrução.

Parágrafo único - Apurada na sindicância que a infração está capitulada como ilícito penal, a autoridade competente encaminhará cópia dos autos ao Ministério Público, independentemente da imediata instauração do processo administrativo.

Art. 176 - Na fase do inquérito, a comissão promoverá a tomada de depoimentos, acareações, investigações e diligências cabíveis, objetivando a coleta de prova, recorrendo, quando necessário, a técnicos e peritos, de modo a permitir a completa elucidação dos fatos.

Art. 177 - É assegurado ao servidor o direito de acompanhar o processo, pessoalmente ou por intermédio de procurador constituído, arrolar e reinquirir testemunhas, produzir provas e contraprovas e formular quesitos, quando se tratar de prova pericial.

§ 1º - O presidente da comissão poderá denegar, motivadamente, pedidos considerados impertinentes, meramente protelatórios, ou de nenhum interesse para o esclarecimento dos fatos.

§ 2º - Será indeferido o pedido de prova pericial, quando a comprovação do fato depender de conhecimento especial de perito.

§ 3º - Em qualquer fase do processo será permitida a intervenção do defensor constituído pelo indiciado.

Art. 178 - As testemunhas serão intimadas a depor mediante mandado expedido pelo presidente da comissão, devendo a segunda via, com o ciente do interessado, ser anexada aos autos.

Parágrafo Único - Se a testemunha for servidor público, a expedição do mandato será imediatamente comunicada ao chefe da repartição onde serve, com a indicação do dia e hora marcados para inquirição.

Art. 179 - O depoimento será prestado oralmente e reduzido a termo, não sendo lícito à testemunha trazê-lo por escrito.

§ 1º - As testemunhas serão inquiridas separadamente.

§ 2º - Na hipótese de depoimentos contraditórios ou que se infirmem, proceder-se-á a acareação entre os depoentes.

Art. 180 - Concluída a inquirição das testemunhas, a comissão promoverá o interrogatório do acusado, observadas as formalidades legais.

§ 1º - No caso de mais de um acusado, cada um deles será ouvido separadamente, e sempre que divergirem em suas declarações sobre fatos ou circunstâncias, será promovida acareação entre eles.

§ 2º - O procurador do acusado poderá assistir ao interrogatório, bem como à inquirição das testemunhas, sendo-lhe vedado interferir nas perguntas e respostas, facultando-se-lhe, porém, reinquiri-las, por intermédio do presidente da comissão.

Art. 181 - Quando houver dúvida sobre a sanidade mental do acusado, a comissão proporá à autoridade competente que seja submetido a exame por junta médica oficial, da qual participe, pelo menos, um médico psiquiatra.

Parágrafo único - O incidente de sanidade mental será processado em auto apartado e apenso ao processo principal, após a expedição do laudo pericial.

Art. 182 - Tipificada a infração disciplinar, será formulada a indicição do servidor, com a especificação dos fatos a ele imputados e das respectivas provas.

§ 1º - O indiciado será citado por mandado expedido pelo presidente da comissão para apresentar defesa escrita, no prazo de 10 (dez) dias, assegurando-se-lhe vista do processo na repartição.

§ 2º - Havendo 2 (dois) ou mais indiciados, o prazo será comum e de 20 (vinte) dias.

§ 3º - O prazo de defesa poderá ser prorrogado pelo dobro, para diligências consideradas indispensáveis.

§ 4º - No caso de recusa do indiciado em apor o ciente na cópia da citação, o prazo para defesa contar-se-á da data declarada, em termo próprio, pelo membro da comissão que fez a citação, com a assinatura de 2 (duas) testemunhas.

Art. 183 - O indiciado que mudar de residência fica obrigado a comunicar à comissão o lugar onde poderá ser encontrado.

Art. 184 - Achando-se o indiciado em lugar incerto e não sabido, será citado por edital, publicado no Diário Oficial do Estado e em jornal de grande circulação na localidade do último domicílio conhecido, para apresentar defesa.

Parágrafo único - Na hipótese deste artigo, o prazo para defesa será de 15 (quinze) dias a partir da última publicação do edital.

Art. 185 - Considerar-se-á revel o indiciado que, regulamente citado, não apresentar defesa no prazo legal.

§ 1º - A revelia será declarada, por termo, nos autos do processo e devolverá o prazo para a defesa.

§ 2º - Para defender o indiciado revel, a autoridade instauradora do processo designará um servidor como defensor dativo, ocupante de cargo de nível igual ou superior ao do indiciado.

Art. 186 - Apreciada a defesa, a comissão elaborará relatório minucioso, onde resumirá as peças principais dos autos e mencionará as provas em que se baseou para formar a sua convicção.

§ 1º - O relatório será sempre conclusivo quanto à inocência ou à responsabilidade do servidor.

§ 2º - Reconhecida a responsabilidade do servidor, a comissão indicará o dispositivo legal ou regulamentar transgredido, bem como as circunstâncias agravantes ou atenuantes.

Art. 187 - O processo disciplinar, com o relatório da comissão, será remetido à autoridade que determinou a sua instauração, para julgamento.

## **SEÇÃO II** **Do Julgamento**

Art. 188 - No prazo de 20 (vinte) dias, contados do recebimento do processo disciplinar, a autoridade julgadora proferirá, motivadamente, a sua decisão.

§ 1º - Se a penalidade a ser aplicada exceder a alçada da autoridade instauradora do processo, este será encaminhado à autoridade competente, que decidirá em igual prazo.

§ 2º - Não decidido o processo no prazo deste artigo, o indiciado, se afastado, reassumirá o exercício do cargo ou função, aí aguardando o julgamento final.

§ 3º - Havendo mais de um indiciado e diversidade de sanções, o julgamento caberá à autoridade competente para a imposição da pena mais grave.

§ 4º - Se a penalidade prevista for a demissão ou cassação de aposentadoria ou disponibilidade, o julgamento caberá à autoridade competente para aplicá-la.

Art. 189 - O julgamento acatará o relatório da comissão, salvo quando, manifestamente, contrário às provas dos autos.

Parágrafo único - Quando o relatório da comissão contrariar as provas dos autos, a autoridade julgadora poderá, motivadamente, agravar a penalidade proposta, abrandá-la ou isentar o servidor de responsabilidade.

Art. 190 - Verificada a existência de vício insanável, a autoridade julgadora declarará a nulidade total ou parcial do processo e ordenará a constituição de outra comissão, para instauração de novo processo.

§ 1º - O julgamento fora do prazo legal não implica em nulidade do processo.

§ 2º - A autoridade julgadora que der causa à prescrição será responsabilizada, na forma da lei.

§ 3º - Extinta a punibilidade pela prescrição, a autoridade julgadora determinará o registro do fato no assentamento individual do servidor.

Art. 191 - Quando a infração estiver capitulada como crime, o processo disciplinar será remetido ao Ministério Público para a instauração da ação penal, ficando traslado na repartição.

Art. 192 - O servidor que responder a processo disciplinar só poderá ser exonerado a pedido, ou aposentado voluntariamente, após a conclusão do processo e o cumprimento da penalidade, acaso aplicada.

Art. 193 - Serão assegurados transportes e diárias:

I - ao servidor convocado para prestar depoimento fora da sede de sua repartição, na condição de testemunha, denunciado ou indiciado;

II - aos membros da comissão e ao secretário quando obrigados a se deslocarem da sede dos trabalhos para realização de diligências necessárias ao esclarecimentos dos fatos.

### **SEÇÃO III** **Da Revisão do processo**



Art. 194 - O processo disciplinar poderá ser revisto, a qualquer tempo, a pedido ou de ofício, quando se aduzirem fatos novos ou circunstâncias suscetíveis de justificar a inocência do punido ou a inadequação da penalidade aplicada.

§ 1º - Em caso de falecimento, ausência ou desaparecimento do servidor, qualquer pessoa da família poderá requerer a revisão do processo.

§ 2º - No caso de incapacidade mental do servidor, a revisão será requerida pelo curador.

Art. 195 - A simples alegação de injustiça da penalidade não constitui fundamento para a revisão, que requer elementos novos ainda não apreciados no processo originário, cabendo o ônus da prova ao requerente.

Art. 196 - O requerimento de revisão do processo será dirigido ao Secretário de Estado, dirigentes de órgãos ou entidades administrativas que, se autorizar a revisão, encaminhará o pedido à repartição onde se originou o processo disciplinar.

Art. 197 - A autoridade que determinou a instauração do processo originário providenciará a constituição de comissão revisora, observando, no que couber, as normas e procedimentos do processo disciplinar.

Parágrafo único - A revisão correrá em apenso ao processo originário.

Art. 198 - Na petição inicial, o requerente pedirá dia e hora para a produção de provas e inquirição das testemunhas que arrolar.

Art. 199 - A comissão revisora terá o prazo de 60 (sessenta) dias para a conclusão dos trabalhos e o prazo para julgamento será de 20 (vinte) dias, contados do recebimento do processo, no curso do qual a autoridade julgadora poderá determinar diligências.

Parágrafo único - O julgamento caberá à mesma autoridade que aplicou a penalidade.

Art. 200 - Julgada procedente a revisão, será declarada sem efeito a penalidade aplicada, restabelecendo-se todos os direitos do servidor, exceto em relação à destituição de cargo em comissão, que será convertida em exoneração.

Parágrafo único - Da revisão do processo não poderá resultar agravamento de penalidade.

**TÍTULO VI**  
**CAPÍTULO ÚNICO**  
**Das Disposições Gerais e Transitórias**  
**SEÇÃO I**  
**Das Disposições Gerais**

Art. 201 - O Dia do Servidor Público será comemorado a 28 (vinte e oito) de outubro.

Art. 202 - Os prazos previstos nesta Lei serão contados em dias corridos, excluindo-se o dia do começo e incluindo-se o último, ficando prorrogado, para o primeiro dia útil seguinte, o prazo vencido em dia que não haja expediente.

Art. 203 - Por motivo de crença religiosa ou de convicção filosófica ou política, o servidor não poderá ser privado de quaisquer dos seus direitos, sofrer discriminação em sua vida funcional, nem eximir-se do cumprimento de seus deveres funcionais.

Art. 204 - Ao servidor público civil é assegurado o direito à livre associação sindical e o direito de greve, na forma da legislação federal.

Art. 205 - Consideram-se da família do servidor, além do cônjuge e filhos, quaisquer pessoas que vivam às suas expensas e constem de seu assentamento individual.

Parágrafo único - Equipara-se ao cônjuge a companheira ou companheiro que comprove união estável como entidade familiar.

## **SEÇÃO II**

### **Das Disposições Finais e Transitórias**

Art. 206 - Os adicionais por tempo de serviço, já concedidos, ficam transformados em triênio e a licença especial em licença-prêmio.

Parágrafo único - É mantida a Progressão Horizontal, como adicional por tempo de serviço, aos servidores que a percebem na data da vigência desta lei e cujo limite não poderá exceder de 80% (oitenta por cento) do vencimento, bem como a Gratificação de Representação percebida pelos ocupantes do cargo de Procurador do Estado.<sup>33</sup>

Art. 207 - O regime jurídico desta Lei Complementar é extensivo aos servidores públicos do Tribunal de Contas do Estado, Ministério Público, Advocacia Geral do Estado, Defensoria Pública e Serventuários da Justiça remunerados com recursos do Estado.

Art. 208 - Os Poderes e órgãos do Estado adotarão as medidas necessárias para adequação de seus procedimentos administrativos às normas contidas nesta Lei Complementar, ressalvados os direitos adquiridos, inclusive quanto a aplicação do art. 164, inciso I, da Lei nº 2.854, de 09 de março de 1968.

Art. 209 - Haverá em cada órgão da administração estadual uma Comissão integrada por servidores, de carreira incumbida de reduzir os riscos inerentes ao trabalho, por meio de normas de saúde, higiene e segurança.

---

<sup>33</sup> Vide ADIN 1331-9 – Rel. Min. Francisco Resek. O STF, ao julgar a medida liminar, decidiu: “Desse modo, DEFIRO em parte o pedido de liminar para, ad referendum do Plenário, suspender, até decisão desta ação direta, a vigência, no art. 40, das expressões “previstas no parágrafo único do art. 206”, assim como da menção, nele contida, aos incisos 00I, VII e XII, do art. 055, todos da Lei Complementar nº 013, de 03.01.94, do Estado do Piauí. Por MAIORIA de votos, o Tribunal REFERENDOU a decisão do presidente (Ministro Sepúlveda Pertence), que deferira, em parte, a medida liminar que suspendera, até a decisão final da ação, a eficácia da expressão “previstas no parágrafo único do art. 206”, contida no art. 040, assim como da menção, nele contida, aos incisos 00I, VII e XII, do art. 055, todos da Lei Complementar nº 013, de 03.01.94, do Estado do Piauí. Vencidos, em parte, os Ministros Marco Aurélio e Octavio Gallotti, que deferiam integralmente a medida liminar. Votou o presidente.” Plenário, 16.08.95. Aguardando julgamento do mérito.

Art. 210 - Esta Lei entrará em vigor na data de sua publicação, revogadas a Lei nº 2.854, de 09 de março de 1968 e demais disposições em contrário.

Palácio Pirajá, em 3 de janeiro de 1994

Governador do Estado  
Secretário de Governo

## Lei nº 5.078, de 26 de julho de 1999

Dispõe sobre a contribuição para o custeio da previdência social dos servidores públicos, ativos, dos três Poderes do Estado, e dá outras providências.

O Governador do Estado do Piauí  
FAÇO saber que o Poder Legislativo decreta e eu sanciono a seguinte Lei:

Art. 1º - A contribuição social do servidor público ativo dos três Poderes do Estado, para a manutenção do regime de previdência social dos seus servidores, incidente sobre a totalidade da remuneração de contribuição e do subsídio, será de:

I – oito pontos percentuais incidentes sobre a parcela da remuneração e do subsídio até o limite de R\$ 280,00 (duzentos e itenta reais);

II – dez pontos percentuais incidentes sobre a parcela da remuneração e do subsídio que exceder a R\$ 280,00 (duzentos e oitenta reais), até o limite de R\$1.200,00 (mil e duzentos reais);

III – doze pontos percentuais incidentes sobre a parcela da remuneração e do subsídio que exceder a R\$1.200,00 (mil e duzentos reais).

Parágrafo único - Entende-se como remuneração de contribuição o vencimento do cargo efetivo, acrescido das vantagens pecuniárias permanentes estabelecidas em lei, os adicionais de caráter individual, ou quaisquer vantagens, inclusive as relativas à natureza ou ao local de trabalho, ou outra paga sob o mesmo fundamento, excluídas:

I – as diárias para viagens, desde que não excedam a cinquenta por cento da remuneração;

II – a ajuda de custo em razão de mudança de sede;

III – a indenização de transporte;

IV – o salário-família.

Art. 2º - O Estado, as autarquias e as fundações públicas estaduais contribuirão para o custeio do regime próprio de previdência social dos seus servidores públicos, observados os critérios estabelecidos no inciso II do art. 50 da Lei nº 4.051, de 21 de maio de 1986.

Art. 3º - O servidor público ativo que permanecer em atividade após completar as exigências para a aposentadoria voluntária integral nas condições previstas no art. 40 da Constituição Federal, na redação dada pela Emenda Constitucional nº 20, de 15 de dezembro de 1998, ou nas condições previstas no art. 8º da referida emenda, fará jus à isenção da contribuição previdenciária até a data da publicação da concessão de sua aposentadoria voluntária ou compulsória.

Art. 4º - As contribuições previstas nesta Lei serão exigidas a partir da sua vigência e, até tal data, fica mantida a contribuição de que trata o art. 50 e seus incisos da Lei

nº 4.051, de 21 de maio de 1986.

Art. 5º - Esta Lei entra em vigor na data da sua publicação.

Art. 6º - Revogam-se as disposições em contrário, em especial o inciso I do art. 50, da Lei nº 4.051, de 21 de maio de 1986.

PALÁCIO DE KARNAK, em Teresina(PI), 26 de julho de 1999

Francisco de Assis de Moraes Souza  
Governador do Estado

Juarez Piauhyense de Freitas Tapety  
Secretário de Governo



PODER JUDICIÁRIO  
TRIBUNAL DE JUSTIÇA DO ESTADO DO PIAUÍ  
**SECRETARIA JUDICIÁRIA**  
**COORDENADORIA JUDICIÁRIA DO PLENO**  
Praça Des. Edgard Nogueira, S/N, Centro Cívico  
TERESINA-PI – CEP: 64.000-830  
Fone: (86) 3226 2613 / 86 98876 1487 – E-mail: secretaria.pleno@tjpi.jus.br

---

## **LEI COMPLEMENTAR Nº 230/2017, DE 29 DE NOVEMBRO DE 2017**

---

*Dispõe sobre o Plano de Carreiras e Remuneração  
dos Servidores do Poder Judiciário do Estado do Piauí*  
(Atualizada com as Leis Complementares Estaduais nºs 232/2018, 235/2018 e 237/2018)



## LEI COMPLEMENTAR Nº 230, DE 29 DE NOVEMBRO DE 2017<sup>1</sup>

*Dispõe sobre o Plano de Carreiras e Remuneração dos Servidores do Poder Judiciário do Estado do Piauí, revoga a LC nº 115, de 25 de agosto de 2008, disposições em contrário e dá outras providências.*

**O GOVERNADOR DO ESTADO DO PIAUÍ**, Faço saber que o Poder Legislativo decreta e eu sanciono a seguinte Lei Complementar:

### **TÍTULO I** **DAS DISPOSIÇÕES PRELIMINARES**

**Art. 1º.** Fica instituído o Plano de Carreiras e Remuneração dos servidores do Poder Judiciário do Estado do Piauí, incluídos os servidores do Tribunal de Justiça, da Justiça Militar, da Corregedoria Geral da Justiça, da Escola Judiciária, do FERMOJUPI, das Comarcas, dos Postos Avançados de Atendimento, dos Termos Judiciários, das Varas, dos Juizados Especiais e seus Anexos, e das Turmas Recursais.

**Parágrafo único.** As carreiras e cargos de provimento efetivo, os cargos em comissão (CC), as funções de confiança (FC) e as unidades do Poder Judiciário Estadual são os previstos nesta Lei e seus anexos.

**Art. 2º.** Aplica-se subsidiariamente a esta Lei o Estatuto dos Servidores Públicos Cíveis do Estado (Lei Complementar nº 13, de 03 de janeiro de 1994 e alterações), a Lei nº 3.716/79 (LOJEPI) e a Lei 9.784/99 (Lei do Processo Administrativo Federal).

**Art. 3º.** O Plano de Carreiras e Remuneração objetiva fundamentalmente a valorização e profissionalização do servidor do Poder Judiciário, a maior eficiência nos serviços jurisdicionais e administrativos da Justiça, mediante:

- I** – adoção de princípios de mérito para ingresso e desenvolvimento na carreira;
- II** – estabelecimento, em caráter sistemático e permanente, de programas de capacitação e aperfeiçoamento dos servidores, através da Escola Judiciária do Piauí (EJUD/TJ-PI) e outros órgãos públicos ou privados.

**Art. 4º.** Os setores que compõem o Poder Judiciário do Estado do Piauí, são classificados em duas áreas, sem prejuízos de outros a serem instituídos:

- I** – área de apoio direto à atividade judicante: setores com competência para impulsionar diretamente a tramitação de processo judicial, tais como, unidades judiciárias de primeiro e de segundo grau, protocolo, distribuição, secretarias judiciárias, gabinetes, contadoria, centrais de mandados, centros judiciários de solução de conflitos, setores de admissibilidade de recursos, setores de processamento de autos, precatórios, arquivo;
- II** – área de apoio indireto à atividade judicante (apoio administrativo): setores sem competência para impulsionar diretamente a tramitação do processo judicial e, por isso, não

<sup>1</sup> Publicado no Diário Oficial do Estado nº 222, de 29 de novembro de 2018, Ano LXXXVI – 129º da República, p. 01/39



definidos como de apoio direto à atividade judicante.

**§1º.** Consideram-se unidades judiciárias de primeiro grau as varas, a justiça itinerante, os juizados especiais e as turmas recursais, compostos por seus gabinetes, secretarias e postos avançados, quando houver.

**§2º.** Consideram-se unidades judiciárias de segundo grau os gabinetes de desembargadores e as secretarias de órgãos fracionários (câmaras de direito público, câmaras especializadas, câmaras reunidas e tribunal pleno), excluídas a Presidência, a Vice-Presidência, a Corregedoria Geral da Justiça e a Vice-Corregedoria Geral da Justiça.

**Art. 5º.** A distribuição de servidores, dos cargos em comissão e das funções de confiança entre as áreas de apoio direto e indireto obedecerá aos seguintes critérios:

**I** – a quantidade total de servidores lotados nas áreas de apoio indireto à atividade judicante (apoio administrativo) deve corresponder a, no máximo, 30% (trinta por cento) do total de servidores;

**II** – a quantidade total de servidores lotados nas áreas de apoio direto à atividade judicante deve corresponder a, no mínimo, 70% (setenta por cento) do total de servidores;

**III** – a quantidade total de servidores das áreas de apoio direto à atividade judicante de primeiro e de segundo graus deve ser proporcional à quantidade média de processos (casos novos) distribuídos a cada grau de jurisdição no último triênio;

**IV** – as unidades judiciárias de primeiro e de segundo graus serão agrupadas por critérios de semelhança, relacionados à competência material, base territorial, entrância ou outro parâmetro objetivo a ser definido pelo Tribunal de Justiça do Estado do Piauí (TJ/PI), para fins de definição da lotação paradigma de seus servidores.

**§1º.** Na apuração do percentual descrito no inciso I, deste artigo, serão excluídos da base de cálculo os servidores lotados na EJUD/TJ-PI e nas áreas de tecnologia da informação.

**§2º.** Serão definidos e publicados pela Presidência do Tribunal, mediante portaria, o agrupamento das unidades judiciárias de primeiro e segundo graus e sua lotação paradigma.

**§3º.** A distribuição de servidores efetivos, de cargos em comissão e de funções de confiança de apoio direto da atividade judicante, deverá ser revista pelo Tribunal de Justiça, no máximo, a cada 2 (dois) anos, a fim de promover as devidas adequações.

**§4º.** Os critérios para a distribuição de servidores, dos cargos em comissão e das funções de confiança entre as áreas de apoio direto e indireto poderão ser adaptados para atender às circunstâncias locais.

## TÍTULO II DAS CARREIRAS

### CAPÍTULO I DA ESTRUTURA E ATRIBUIÇÕES DAS CARREIRAS

**Art. 6º.** O quadro de pessoal efetivo do Poder Judiciário do Piauí é composto pelas seguintes carreiras:

**I** – analista judiciário: atividades de planejamento, organização, coordenação, supervisão técnica, assessoramento, direção de unidade, estudo, pesquisa, elaboração de laudo, parecer, prática de ato processual, cumprimento de decisão judicial e administrativa, prestação de informação de relevante complexidade;

**II** – técnico judiciário: execução de suporte técnico em áreas específicas, de acordo





com a sua formação, ou de suporte administrativo em geral como cumprimento de decisão judicial e administrativa, prestação de informação;

III - auxiliar judiciário: atividades básicas de apoio operacional ou de suporte administrativo.

Parágrafo único. Os atuais ocupantes dos cargos que compõem a carreira de técnico judiciário e auxiliar judiciário passarão a compor quadro em extinção, devendo os cargos providos serem extintos quando ocorrerem suas vacâncias.

**Art. 7º.** O quadro de pessoal efetivo do Poder Judiciário do Piauí é composto pelas seguintes áreas de atuação:

**I** – judiciária: compreende os serviços realizados privativamente por bacharéis em direito, abrangendo processamento de feitos, análise e pesquisa de legislação, doutrina e jurisprudência nos vários ramos do direito, assessoramento, elaboração de minutas, prática de ato processual, cumprimento de decisão judicial e administrativa e execução de mandados;

**II** – apoio especializado: compreende os serviços cuja execução exija dos titulares o devido registro no órgão fiscalizador do exercício da profissão ou o domínio de habilidades específicas a critério da administração;

**III** – administrativa: compreende os serviços relacionados com recursos humanos, material e patrimônio, licitações, contratos, orçamento, finanças, segurança e transporte, dentre outras atividades complementares a critério da administração.

§1º. O quadro geral dos cargos de provimento efetivo e respectivas carreiras e áreas de atuação são os constantes do Anexo II, desta Lei.

§2º. As atribuições dos cargos de provimento efetivo são os constantes do Anexo III, desta Lei.

§3º. O nível 6A, referências I, II e III, constante do Anexo V, desta Lei, somente será implantado no exercício financeiro de 2019.

## CAPÍTULO II DO INGRESSO NA CARREIRA

**Art. 8º.** O ingresso em qualquer das carreiras de provimento efetivo do Poder Judiciário dar-se-á na primeira referência do nível inicial, após aprovação em concurso público de provas ou de provas e títulos.

§1º. O concurso público constará de exames de conhecimento, com caráter eliminatório e classificatório, compreendendo testes objetivos e/ou dissertativos e, conforme o caso, a realização de testes práticos.

§2º. O curso de inicialização deverá preceder a nomeação e terá caráter classificatório.

**Art. 9º.** Além dos requisitos previstos no Estatuto dos Servidores Cíveis do Estado, são requisitos de escolaridade para ingresso nas carreiras judiciárias os constantes do Anexo IV, desta Lei.

§1º. Havendo exigência legal, em especial para os cargos de Arquiteto, Assistente Social, Bibliotecário, Contador, Enfermeiro, Engenheiro, Estatístico, Fisioterapeuta, Médico, Nutricionista, Odontólogo, Psicólogo, é obrigatória a inscrição no Conselho Profissional respectivo.

§2º. A comprovação do atendimento dos requisitos previstos neste artigo será exigida no momento da posse.



### CAPÍTULO III DO DESENVOLVIMENTO NA CARREIRA

**Art. 10.** O desenvolvimento do servidor nos cargos de provimento efetivo das carreiras dos quadros de pessoal do Poder Judiciário dar-se-á mediante progressão funcional e promoção, observados os critérios e as normas a serem definidos em resolução.

**Art. 11.** A progressão funcional consiste na movimentação do servidor de uma referência para a seguinte, dentro do mesmo nível, anualmente, na data em que o servidor completar o interstício de um ano, na referência em que estiver posicionado.

**Parágrafo único.** Somente terá direito à progressão funcional o servidor que apresentar o desempenho satisfatório em avaliação de desempenho.

**Art. 12.** A promoção consiste na movimentação do servidor da última referência de um nível para a primeira referência do nível seguinte, na data em que o servidor completar o interstício de um ano, da progressão funcional imediatamente anterior.

**Parágrafo único.** Somente terá direito à promoção o servidor que apresentar desempenho satisfatório em avaliação de desempenho e participar, durante o período de permanência no nível, de conjunto de ações de educação corporativa a serem definidas em resolução.

### CAPÍTULO IV DA AVALIAÇÃO DE DESEMPENHO

**Art. 13.** Entende-se por avaliação de desempenho a verificação sistemática e formal da atuação do servidor no exercício das atribuições do cargo, no âmbito de sua área e/ou especialidade, mediante critérios objetivos, a serem definidos em resolução.

**Art. 14.** As avaliações serão públicas e realizadas na forma prevista em resolução, que deverá conter, dentre outros, os seguintes regramentos:

**I** – critérios de avaliação;

**II** – ficha de avaliação, na qual são atribuídos pontos para cada fator avaliativo;

**III** – plano de ação.

**Parágrafo único.** A avaliação de desempenho ocorrerá a cada ano e seus procedimentos serão coordenados e acompanhados pela Comissão de Avaliação, segundo critérios estabelecidos nesta Lei e em resolução.

**Art. 15.** São finalidades da Avaliação de Desempenho:

**I** – verificar a aptidão do servidor para o exercício das atribuições do cargo efetivo, no período do estágio probatório;

**II** – subsidiar a concessão de progressão funcional e promoção;

**III** – detectar necessidades de capacitação e desenvolvimento;

**IV** – identificar necessidades de adequação na lotação do servidor.

**Art. 16.** Os processos de avaliação de desempenho compõem-se, obrigatoriamente, da auto-avaliação do servidor e da avaliação da chefia imediata e de relatório final da Comissão de



#### Avaliação.

§1º. O servidor que, no período de avaliação, houver trabalhado sob a direção de mais de uma chefia, será avaliado por aquela a qual esteve subordinado por mais tempo.

§2º. O servidor cedido será avaliado pelo órgão no qual estiver em exercício, observados os critérios estabelecidos em resolução.

§3º. O servidor licenciado ou afastado do cargo de provimento efetivo não será submetido à avaliação de desempenho, sendo retomada a contagem do seu tempo para efeito de avaliação após o término do afastamento ou licença.

**Art. 17.** O processo da avaliação de desempenho do servidor é contínuo e permanente, devendo representar o desempenho predominante no período avaliativo, incluindo o mês de sua formalização.

**Art. 18.** Os servidores de cargo efetivo em estágio probatório serão avaliados a cada 12 (doze) meses pela Comissão de Avaliação, que apresentará relatório de desempenho para verificação da aptidão ou inaptidão para fim de concessão de estabilidade.

### TÍTULO III DOS DIREITOS E VANTAGENS

#### CAPÍTULO I DO SUBSÍDIO

**Art. 19.** Os titulares dos cargos das Carreiras enumeradas no artigo 7º serão remunerados por subsídio, na forma do Anexo V desta Lei, fixado em parcela única, vedado o acréscimo de qualquer gratificação, adicional, abono, prêmio, verba de representação ou outra espécie remuneratória, conforme o disposto no artigo 37, X e XI da Constituição Federal (CF), ressalvadas as verbas de caráter indenizatório, as funções comissionadas (FC) e os cargos em comissão (CC), ou oriundas de condições especiais, devendo ser reajustado no mês de janeiro de cada ano, mediante lei específica, condicionado à disponibilidade orçamentária e financeira.

**Art. 20.** A percepção de subsídio não exclui o direito ao recebimento, nos termos da legislação e regulamentação específica, das seguintes espécies remuneratórias:

**I** – gratificação natalina;

**II** – adicional de férias;

**III** – abono de permanência, de que tratam o §19, do artigo 40, da CF, o §5º, do artigo 2º e o §1º, do artigo 3º da Emenda Constitucional (EC) nº 41/2003;

**IV** – retribuição pelo exercício de função de direção, chefia e assessoramento;

**V** – parcelas indenizatórias previstas em lei.

**Art. 21.** É vedada qualquer diferença remuneratória entre os servidores de uma mesma carreira, independente da comarca em que estejam em exercício.

**Art. 22.** Fica assegurada revisão geral anual, da remuneração e subsídios, sem prejuízo de eventuais reajustes, sempre na mesma data e sem distinção de índices, condicionada à disponibilidade orçamentária e financeira.



## CAPÍTULO II DAS VANTAGENS

**Art. 23.** Além do subsídio, poderão ser pagas ao servidor as seguintes vantagens:

**I** – indenizações;

**II** – gratificações;

**III** – adicionais.

**§1º.** As indenizações não se incorporam ao subsídio ou provento de aposentadoria para qualquer efeito.

**§2º.** As vantagens e a regulamentação contidas nesta Lei não excluem outras decorrentes da aplicação do Estatuto dos Servidores Públicos Cíveis do Estado (Lei Complementar nº 13, de 03 de janeiro de 1994) e leis específicas.

**§3º** Ficam proibidos a concessão e o pagamento de qualquer vantagem remuneratória não prevista em lei ou em valores superiores aos nela previstos, assim como de vantagem absorvida por esta Lei.

### Seção I Das Indenizações

**Art. 24.** Constituem indenizações ao servidor:

**I** – indenização de transporte;

**II** – auxílio-alimentação;

**III** – auxílio-saúde.

**Parágrafo único.** As indenizações previstas neste artigo e no Anexo VI não serão devidas aos servidores afastados do Poder Judiciário, exceto os afastamentos autorizados para capacitação do servidor, não se incorporam aos proventos de inatividade e terão seus valores fixados por ato da Presidência do Tribunal, a quem também caberá a sua revisão anual.

#### Subseção I Da indenização de transporte

**Art. 25.** Aos ocupantes do cargo de Oficial de Justiça e Avaliador, no efetivo exercício de suas atribuições, é devida indenização para o custeio das despesas com transporte, conforme disposições contidas no Anexo VI, desta Lei.

#### Subseção II Do auxílio alimentação

**Art. 26.** Aos servidores efetivos e comissionados no exercício das atribuições das suas carreiras é devido auxílio alimentação, de natureza indenizatória, conforme disposto no Anexo VI, desta Lei.

#### Subseção III Do auxílio saúde

**Art. 27.** Aos servidores efetivos e comissionados no exercício das atribuições das suas



carreiras ou cargos é devido auxílio saúde, de natureza indenizatória, conforme disposto no Anexo VI, desta Lei.

## Seção II Das Retribuições, Gratificações e Adicionais

**Art. 28.** Poderão ser deferidos aos servidores as seguintes retribuições, gratificações e adicionais:

- I** – retribuição pelo exercício de cargo ou função de direção, chefia e assessoramento;
- II** – retribuição por condição especial de trabalho;
- III** – gratificação natalina;
- IV** – gratificação por incremento de produtividade;
- V** – adicional de férias;
- VI** – adicional de insalubridade;
- VII** – adicional de periculosidade.

§1º O direito aos adicionais previstos nos incisos VI e VII cessa com o afastamento do servidor da atividade que exercia ou com a eliminação dos riscos que deram causa à sua concessão.

§2º Os adicionais previstos nos incisos VI e VII são inacumuláveis, devendo o servidor optar por um deles.

§ 3º Os valores das indenizações previstas nos incisos IV, VI e VII serão revisados por ato da Presidência do Tribunal de Justiça.

### Subseção I Da retribuição pelo exercício de cargo ou função de direção, chefia e assessoramento

**Art. 29.** Ao servidor ocupante de cargo efetivo investido em função de direção, chefia ou assessoramento é devida retribuição pelo seu exercício.

**Parágrafo único.** A retribuição de que trata este artigo consta no Quadro II, do Anexo VII, desta Lei.

### Subseção II Da gratificação por condições especiais de trabalho

**Art. 30.** A gratificação por condições especiais de trabalho será concedida com vista ao interesse público de fixar o servidor em determinadas regiões, incentivá-lo no exercício de determinadas funções ou quando estas se realizarem em locais de difícil provimento e serviços de natureza especial com dedicação exclusiva.

**Parágrafo único.** O modo, a forma e as circunstâncias para pagamento desta gratificação serão definidos em resolução do Tribunal de Justiça, cabendo à Presidência a concessão, a fixação e revisão.

### Subseção III Da gratificação natalina

**Art. 31.** A gratificação natalina corresponde a 1/12 (um doze avos) do subsídio a que o servidor fizer jus no mês de dezembro, por mês de exercício no respectivo ano, podendo ser paga em duas parcelas, sendo a primeira até junho e a segunda até dezembro.



**Art. 32.** O servidor exonerado perceberá sua gratificação natalina, proporcionalmente aos meses de exercício, calculada sobre a remuneração do mês da exoneração.

**Parágrafo único.** No caso de pagamento proporcional da gratificação natalina, as frações inferiores a 01 (um) mês serão contadas por dia efetivamente trabalhado.

#### Subseção IV Da gratificação por incremento de produtividade

**Art. 33.** As disposições relativas à percepção da gratificação por incremento de produtividade serão dispostas em resolução do Tribunal de Justiça do Estado do Piauí, que estabelecerá critérios de aferimento, pontuação, quantificação e revisão, dentre outros.

#### Subseção V Do adicional de férias

**Art. 34.** Será pago ao servidor por ocasião das férias um adicional correspondente a 1/3 (um terço) do subsídio do período de férias, independente de solicitação do servidor.

**Parágrafo único.** No caso do servidor exercer função de confiança ou cargo em comissão de direção, chefia ou assessoramento a respectiva vantagem será considerada no cálculo do adicional de que trata este artigo.

#### Subseção VI Do adicional de insalubridade

**Art. 35.** Os servidores do Poder Judiciário que desempenham atividades com habitualidade em locais insalubres ou em contato permanente com substâncias tóxicas e/ou radioativas fazem jus a adicional de insalubridade, conforme disposto no Anexo VI, desta Lei.

#### Subseção VII Do adicional de periculosidade

**Art. 36.** Aos ocupantes da carreira de Oficial de Justiça e Avaliador no efetivo exercício de suas atribuições é devido adicional de periculosidade, conforme disposto no Anexo VI, desta Lei.

## TÍTULO IV DOS CARGOS EM COMISSÃO, DAS FUNÇÕES DE CONFIANÇA E DAS SUBSTITUIÇÕES

### CAPÍTULO I DAS DISPOSIÇÕES GERAIS

**Art. 37.** Os cargos de provimento em comissão, classificados pelo símbolo CC, escalonados em níveis de 1 a 6, e as funções de confiança, classificadas pelo símbolo FC/FC-PM, escalonadas em níveis de 1 a 3, são apenas os previstos no Quadro I, do Anexo VII, e Anexo XI, desta Lei.



**§1º.** As funções de confiança e os cargos em comissão somente se destinam às atribuições de direção, chefia e assessoramento.

**§2º.** Os cargos em comissão, num percentual de 30% (trinta por cento), serão ocupados por servidores de carreira.

**§3º.** O percentual descrito no parágrafo 2º, do art. 37, desta Lei, será exigido a partir de 1º de janeiro de 2019.

**Art. 38.** O preenchimento dos cargos em comissão e funções de confiança, da Vice-Presidência, da Corregedoria-Geral da Justiça, da Vice-Corregedoria da Justiça, dos Gabinetes dos Desembargadores, da Escola Judiciária do Piauí, da Ouvidoria Judiciária e dos Juízos de Direito e demais unidades judiciárias, será feito mediante indicação de seus titulares e nomeação pelo Presidente do Tribunal de Justiça, desde que atendidos os requisitos previstos em Lei.

**Art. 39.** Aos Magistrados, em qualquer grau de jurisdição, competem as indicações para os cargos em comissão e funções de confiança de seus gabinetes.

**Art. 40.** Os cargos e funções de que trata este título serão providos levando-se em conta a formação profissional, que deverá ser compatível com a natureza das atribuições e responsabilidades.

**Parágrafo único.** Quando exigida qualificação ou habilitação específica para os cargos em comissão ou funções de confiança, o substituto legal ou eventual deverá possuir igual ou superior qualificação ou habilitação.

**Art. 41.** No âmbito da jurisdição do Tribunal ou Juízo de 1º grau, e dos detentores de cargos de chefia, direção e assessoramento, é vedada a nomeação ou designação, para os cargos em comissão e funções de confiança, de cônjuge, companheiro, parente ou afim, em linha reta ou colateral, até o terceiro grau, inclusive, dos respectivos membros e juízes vinculados, salvo a de ocupante de cargo público de provimento efetivo, caso em que a vedação é restrita à nomeação ou designação para servir perante o magistrado ou servidor determinante da incompatibilidade.

**Parágrafo único.** As vedações acima aplicam-se aos auxiliares da justiça, estagiários, interinos ou interventores dos cartórios extrajudiciais e demais colaboradores.

## CAPÍTULO II DOS CARGOS EM COMISSÃO

**Art. 42.** Do valor da gratificação pelo exercício de cargo em comissão, correspondente ao vencimento e à representação, será na forma do Anexo VII, Quadro II.

**Parágrafo único.** O servidor público efetivo designado para cargo em comissão poderá optar entre o vencimento ou subsídio do cargo efetivo que ocupa e o vencimento da gratificação do cargo em comissão.

**Art. 43.** São requisitos para o exercício dos cargos em comissão as qualificações ou níveis de escolaridade referidos no Anexo X, desta Lei.

**§1º.** Os requisitos de escolaridade para investidura dos cargos de Assessor de Magistrado e Oficial de Gabinete, constantes no Anexo X, desta Lei, passarão a ser exigidos a partir de 1º de janeiro de 2020;

**§2º.** O requisito de 02 (dois) anos de prática jurídica para investidura do cargo de



Consultor Jurídico, constante no Anexo X, desta Lei, passará a ser exigido a partir de 1º de janeiro de 2020.

### CAPÍTULO III DAS FUNÇÕES DE CONFIANÇA

**Art. 44.** As funções de confiança somente podem ser exercidas por servidores efetivos do Poder Judiciário do Estado do Piauí.

**Art. 45.** São requisitos para o exercício das funções de confiança as qualificações ou níveis de escolaridade descritos no Anexo X, desta Lei.

**Parágrafo único.** O requisito de escolaridade para investidura do cargo de Secretário de Vara, constante no Anexo X, desta Lei, passará a ser exigido a partir de 1º de janeiro de 2020.

### CAPÍTULO IV DAS SUBSTITUIÇÕES

**Art. 46.** Nos casos de impedimentos ou afastamentos superiores a 15 (quinze) dias, o servidor ocupante de cargo de direção ou chefia será substituído com prazo determinado e não superior a 180 (cento e oitenta) dias por servidor, observando o artigo 37, desta Lei.

**§1º.** O substituto perceberá, além de seu subsídio, a diferença proporcional ao tempo de substituição, calculada como se fosse titular do cargo em comissão ou da função de confiança.

**§2º.** A substituição prevista no *caput* deste artigo depende de ato da administração, após a indicação do gestor.

### TÍTULO V DAS DISPOSIÇÕES DISCIPLINARES

#### CAPÍTULO I DAS DISPOSIÇÕES GERAIS

**Art. 47.** Aos servidores do Poder Judiciário, *lato sensu*, aplicam-se, no que couber, as disposições previstas no Título IV – Do Regime Disciplinar, e no Título V – Do Processo Administrativo Disciplinar, da Lei Complementar nº 13, de 03 de janeiro de 1994 – Estatuto dos Servidores Públicos Civis do Estado e subsidiariamente a Lei Federal nº 9.784/1999.

**Art. 48.** A responsabilização administrativa de servidor do Poder Judiciário dar-se-á sempre através de processo administrativo ou sindicância, em que lhe seja assegurada ampla defesa.

**§1º.** A instauração do Processo Administrativo Disciplinar caberá:

**I** – Ao Presidente do Tribunal de Justiça, quanto aos ilícitos administrativos atribuídos a servidores ou colaboradores do Poder Judiciário em exercício no 2º grau de jurisdição;

**II** – Ao Corregedor Geral da Justiça, quanto aos ilícitos administrativos imputados a servidores ou colaboradores do Poder Judiciário em exercício no 1º grau de jurisdição.

**§2º.** O Corregedor Geral poderá delegar a instauração e condução de processo administrativo disciplinar ou sindicância a Juiz de Direito ou, onde houver mais de um Juiz, ao





Juiz de Direito Diretor do Fórum.

§3º. Os auxiliares da justiça e demais colaboradores, delegatários do serviço extrajudicial e seus substitutos, nos atos ligados a arrecadação de tributos, submetem-se a regra do inciso I, deste artigo.

~~§4º. Os auxiliares da justiça e demais colaboradores, delegatários do serviço extrajudicial e seus substitutos, nos atos ligados à prática registral e notarial, submetem-se a regra do inciso II, deste artigo.~~

§4º. Os auxiliares da justiça e demais colaboradores submetem-se a regra do inciso II deste artigo. (Redação dada pela Lei Complementar nº 237/2018, de 17.07.2018).

§5º. Compete à Presidência do Tribunal de Justiça, de maneira privativa, processar e julgar em caráter administrativo, a incursão dos delegatários e substitutos, auxiliares da justiça e demais colaboradores, nas regras contidas na Súmula Vinculante nº 13.

## CAPÍTULO II DOS DEVERES E PROIBIÇÕES

**Art. 49.** Os servidores do Poder Judiciário devem ter irrepreensível procedimento na vida pública, pugnando pelo prestígio da Administração da Justiça, velando pela dignidade de suas funções e respeitando a integridade das funções dos Magistrados, membros do Ministério Público, da Defensoria Pública e da Advocacia pública.

**Parágrafo único.** São deveres dos servidores do Poder Judiciário, além dos inerentes aos demais servidores públicos civis do Estado do Piauí:

- I – desincumbir-se diariamente de seus encargos funcionais, no foro ou repartição;
- II – desempenhar com zelo e presteza, dentro dos prazos, os serviços, os seus encargos e os que, na forma da lei, lhes forem atribuídos;
- III – observar sigilo funcional quanto à matéria dos procedimentos em que atuar e, especialmente, nos que tramitam em segredo de Justiça ou sigilo fiscal;
- IV – transmitir, em caso de impedimento, afastamento, substituição e exoneração, as informações e dados da unidade ou cargo ocupado.

**Art. 50.** Além das proibições previstas no Estatuto dos Servidores Civis, aos servidores do Poder Judiciário é vedado especialmente:

- I – dificultar ou deixar de levar ao conhecimento de autoridade competente, por via hierárquica e em 24 (vinte e quatro) horas, representação, petição, recurso judicial ou administrativo ou documento que houver recebido, se não estiver na sua alçada resolvê-lo;
- II – negligenciar a guarda de bens ou valores pertencentes à repartição judicial ou de terceiros que estejam sob sua responsabilidade, possibilitando assim que eles se danifiquem ou se extraviem;
- III – indicar ou insinuar nome de advogado para assistir pessoa que seja parte em processo judicial;
- IV – advogar, assistir ou intervir, ainda que informalmente, nos processos judiciais ou administrativos, exceto no último caso, quando nomeado como defensor dativo, na forma da lei;
- V – cobrar custas, emolumentos ou qualquer outra quantia ou vantagem não prevista em lei ou em valor superior ao previsto legalmente;
- VI – utilizar, ceder ou permitir que outrem use objetos e valores apreendidos ou depositados no interesse da Justiça, salvo nos casos previstos em lei;
- VII – participar de correntes financeiras, consórcios, pirâmides e qualquer ato que



evidencie usura.

**Parágrafo único.** Conforme a natureza das suas atribuições, ao servidor do Poder Judiciário é também proibido:

**I** – faltar com a lisura na feitura de cálculos, atualizações e perícias ou aplicar índices de correção ou de juros superiores aos permitidos por lei;

**II** – manipular, por qualquer modo ou expediente, o caráter aleatório da distribuição.

### CAPÍTULO III DAS SANÇÕES DISCIPLINARES

**Art. 51.** Aos servidores do Poder Judiciário serão aplicadas as mesmas sanções previstas no Estatuto dos Servidores Cívicos do Estado do Piauí.

**Art. 52.** As sanções disciplinares de advertência e suspensão são aplicadas pelo Presidente do Tribunal de Justiça e Corregedor Geral da Justiça, no âmbito de suas competências, e, as de demissão, de cassação de aposentadoria, de destituição de cargo em comissão e de destituição de função gratificada são aplicadas pelo Presidente do Tribunal de Justiça.

**Art. 53.** A demissão será aplicada nos casos de violação às proibições previstas no art. 50, IV a VI, e parágrafo único desta Lei e nos casos previstos no Estatuto dos Servidores Cívicos do Estado do Piauí

**Art. 54.** A suspensão será aplicada por infração ao disposto no art. 50, II, III e VII, e nos casos previstos no Estatuto dos Servidores Públicos Cívicos do Estado.

**Art. 55.** A advertência será aplicada no caso de violação ao art. 50, I, e também nas hipóteses previstas no Estatuto dos Servidores Cívicos do Estado do Piauí.

## TÍTULO VI DAS UNIDADES

### CAPÍTULO I DAS UNIDADES DO TRIBUNAL DE JUSTIÇA

#### Seção I Das Disposições Gerais

**Art. 56.** Os serviços do Poder Judiciário são realizados pelos Gabinetes de Desembargadores, Turmas Recursais, Gabinetes de Magistrados, Varas, Juizados Especiais Cíveis e Criminais, Tribunal Pleno, Presidência, Vice-Presidência, Corregedoria Geral da Justiça, Vice-Corregedoria Geral da Justiça, Ouvidoria Judicial, FERMOJUPI e Escola Judiciária (EJUD/PI), dentre outras unidades de apoio direto e indireto à atividade jurisdicional constantes da presente Lei e cujas atribuições serão previstas no Regimento Interno do Tribunal de Justiça do Estado do Piauí, sem prejuízo de outros diplomas reguladores.

#### Seção II Das Unidades de Segundo Grau



**Art. 57.** Integrarão a estrutura da Presidência do Tribunal de Justiça, como órgãos de assessoramento direto:

- I** – o Gabinete da Presidência (GABPRE);
- II** – o Gabinete dos Juízes Auxiliares (GABJAPRE);
- III** – a Coordenadoria de Precatórios (CPREC);
- IV** – a Assessoria de Comunicação (ASCOM);
- V** – o Conselho de Segurança Institucional (CSI);
- VI** – a Secretaria Geral (SECGER);
- VII** – a Secretaria da Presidência (SECPRE);
- VIII** – a Secretaria de Assuntos Jurídicos (SAJ);
- IX** – a Secretaria de Administração e Gestão de Pessoas (SEAD);
- X** – a Secretaria de Orçamento e Finanças (SOF);
- XI** – a Secretaria de Tecnologia da Informação e Comunicação (STIC);
- XII** – a Secretaria de Gestão Estratégica (SEGES);
- XIII** – a Secretaria Judiciária (SEJU);
- XIV** – o Fundo de Reparamento e Modernização do Poder Judiciário do Estado do Piauí (FERMOJUPI);
- XV** – a Superintendência de Segurança (SUSEG);
- XVI** – a Superintendência de Controle Interno (SCI);
- XVII** – a Superintendência Administrativa da EJUD (SUPEJUD);
- XVIII** – o Grupo de Monitoramento e Fiscalização do Sistema Carcerário (GMF);
- XIX** – a Coordenadoria Estadual Judiciária da Infância e da Juventude (CEJIJ);
- XX** – a Coordenadoria Estadual da Mulher em Situação de Violência Doméstica e Familiar (CEM);
- XXI** – o Núcleo Socioambiental (NUSA);
- XXII** – o Núcleo Permanente de Métodos Consensuais de Solução de Conflitos (NUPEMEC);
- XXIII** – o Centro Judiciário de Resolução de Conflitos e Cidadania de 2º Grau (CEJUSC - 2º Grau).
- XXIV** – a Comissão Permanente de Processo Administrativo Disciplinar (CPPAD – 2º grau);
- XXV** – a Comissão Permanente de Processo Administrativo Contratual (CPPAD – CONTRATUAL);
- XXVI** – o Núcleo de Memória Judicial (NMJ);
- XXVII** – a Supervisão dos Juizados Especiais (SUJECCs).

**§1º.** Integrarão a estrutura da Presidência do Tribunal de Justiça, como órgãos operacionais vinculados à Secretaria Geral:

- I** – Superintendência de Engenharia e Arquitetura (SENA);
  - II** – Superintendência de Licitações e Contratos (SLC);
  - III** – Superintendência de Gestão de Contratos e Convênios (SGC);
- §2º.** Integrará a estrutura da Presidência do Tribunal de Justiça, como órgão operacional vinculado à Secretaria de Administração e Pessoal (SEAD) a Superintendência de Gestão da Saúde e Qualidade de Vida (SUGESQ).

**Art. 58.** Integrarão a estrutura da Vice-Presidência do Tribunal de Justiça:

- I** – Núcleo de Gerenciamento de Precedentes (NUGEP), incluídas as análises dos



recursos extraordinários, especiais e ordinários, dentre outros;

**II** – Núcleo de Apoio às Unidades Judiciárias de 1º e 2º graus (NAUJ), composto de equipes multidisciplinares destinadas a dar suporte aos juízos de primeira e segunda instância.

**Art. 59.** A Escola Judiciária do Piauí (EJUD-PI) integra o Poder Judiciário, como órgão executor das políticas de educação, seleção e formação institucionais.

**Art. 60.** A Ouvidoria Judiciária (OUV) integra o Poder Judiciário como órgão auxiliar da Presidência.

**Art. 61.** São unidades jurisdicionais de segundo grau do Tribunal de Justiça os seguintes órgãos:

**I** – o Tribunal Pleno;

**II** – as Câmaras Reunidas Cíveis (CRCiv);

**III** – as Câmaras Reunidas Criminais (CRCrim);

**IV** – as Câmaras de Direito Público (CDP);

**V** – as Câmaras Cíveis (CCiv);

**VI** – as Câmaras Criminais (CCrim);

**VII** – os Gabinetes dos Desembargadores (GABDES).

§1º. Os cargos de consultor jurídico destinados aos órgãos julgadores fracionários constantes dos incisos IV, V e VI, deste artigo, serão distribuídos igualmente entre seus membros.

§2º. A força de trabalho diretamente vinculado aos membros do Tribunal investidos em cargo de direção será lotado nas áreas de apoio direto ou indireto à atividade judicante, a depender da atribuição para impulsionar ou não a tramitação do processo judicial.

**Art. 62.** Integram a estrutura da Corregedoria Geral da Justiça do Tribunal de Justiça (CGJ/PI), como órgãos de assessoramento direto as seguintes unidades:

**I** – o Gabinete do Corregedor Geral da Justiça (GABCOR);

**II** – o Gabinete dos Juízes Auxiliares (GABJACOR);

**III** – a Secretaria da Corregedoria (SECCOR);

**IV** – a Assistência de Imprensa (ASI);

**V** – a Comissão Permanente de Processo Disciplinar de 1º grau (CPPAD – 1º grau);

**VI** – a Comissão Estadual Judiciária de Adoção Internacional (CEJAD);

**VII** – a Coordenadoria de Planejamento e Modernização (COPM);

**VIII** – a Coordenadoria de Cadastro e Tramitação Processual (COCTP);

**IX** – os Centros Judiciários de Resolução de Conflitos e Cidadania de 1º Grau (CEJUSCs – 1º Grau);

**Parágrafo único.** Integra a estrutura da Vice-Corregedoria Geral da Justiça do Piauí, como órgão de assessoramento direto, o Gabinete do Vice-Corregedor Geral da Justiça (GABVICOR).

~~**Art. 63.** Compete ao Vice-Corregedor Geral da Justiça substituir o Corregedor Geral da Justiça nas suas ausências, impedimentos e afastamentos, supervisionar a Justiça Itinerante e as atividades extrajudiciais.~~

**Art. 63.** Compete ao Vice-Corregedor Geral da Justiça: (redação dada pela Lei Complementar nº 237/2018, de 17.07.2018)

**I** – substituir o Corregedor Geral da Justiça nas suas ausências, impedimentos, suspeições e afastamentos; (inciso acrescido pela LCE nº 237/2018, de 17.07.2018)



**II** – supervisionar a Justiça Itinerante; **(inciso acrescido pela LCE nº 237/2018, de 17.07.2018)**

**III** – exercer a fiscalização disciplinar, o controle, a normatização e a orientação dos serviços extrajudiciais. **(inciso acrescido pela LCE nº 237/2018, de 17.07.2018)**

**§1º.** A instauração de processo administrativo disciplinar em face de delegatários do serviço extrajudicial, sem prejuízo da competência do Juiz Corregedor Permanente, caberá ao Vice-Corregedor Geral da Justiça, sendo de sua competência exclusiva a aplicação das penalidades constantes no art. 33 da Lei Federal nº 8.935, de 18 de novembro de 1994, cumprindo ao Presidente do Tribunal de Justiça decretar a pena de perda de delegação, quando for o caso. **(inciso acrescido pela LCE nº 237/2018, de 17.07.2018)**

**§2º.** A designação e a cessação de interinidade para as serventias extrajudiciais declaradas vagas é de competência exclusiva do Vice-Corregedor Geral da Justiça. **(parágrafo acrescido pela LCE nº 237/2018, de 17.07.2018)**

**§3º.** Para o exercício de suas atribuições, relativamente às atividades extrajudiciais, o Vice-Corregedor Geral da Justiça utilizará a estrutura administrativa da Corregedoria Geral da Justiça e os cargos que compõem a estrutura do Gabinete da Vice-Corregedoria, elencados no Anexo VIII, Quadro XXXII, desta Lei. **(parágrafo acrescido pela LCE nº 237/2018, de 17.07.2018)**

**Art. 64.** Caberá ao Tribunal de Justiça, através de resolução, criar e extinguir as suas unidades administrativas internas e dispor sobre as suas respectivas atribuições, desde que não importe na criação ou na extinção de cargos, ou aumento de despesa.

### **Seção III**

#### **Das Unidades Judiciárias de Primeiro Grau**

**Art. 65.** São unidades judiciárias de primeiro grau as turmas recursais, as varas e os juizados especiais, compostos por secretarias, diretorias de juizados especiais, gabinetes e postos avançados, onde houver, com as atribuições previstas no Anexo IX, desta Lei, sem prejuízo de outras atividades pertinentes a sua área de atuação.

**Art. 66.** As Turmas Recursais contam com uma secretaria unificada, e um cargo em comissão de Diretor de Secretaria, indicado pelo Juiz Presidente da Primeira Turma Recursal, nomeado pela Presidência do TJ/PI.

**Parágrafo único.** Os membros das Turmas Recursais farão jus a um Assessor de Magistrado, nomeado na forma do art. 38, desta Lei.

**Art. 67.** Consideram-se de apoio direto e indireto aos órgãos judiciários de 1º grau da Capital as seguintes unidades:

**I** – a Diretoria de Fórum;

**II** – os Centros Judiciários de Soluções de Conflitos e Cidadania de 1º Grau (CEJUSCs - 1º Grau);

**III** – a Central de Inquéritos e Audiência de Custódia;

**IV** – a Central de Mandados;

**V** – a Seção de Contadoria Judicial;

**VI** – a Seção de Apoio Psicossocial;

**VII** – a Seção de Atendimento e Certidões de 1º grau;



- 
- VIII** – a Distribuição de 1º grau;
  - IX** – os Gabinetes de juízes de 1º grau;
  - X** – as Secretarias das varas de 1º grau;
  - XI** – as Diretorias de juizados especiais;
  - XII** – as Turmas Recursais;
  - XIII** – a Justiça Itinerante;
  - XIV** – A Comissão Permanente de Processo Administrativo Disciplinar de 1º Grau (CPPAD - 1º Grau);

**§1º.** A criação e instalação das unidades enumeradas nos incisos deste artigo nas demais Comarcas do Estado dependerá da conveniência e oportunidade do Tribunal, mediante análise de parâmetros objetivos a serem definidos por resolução, e disponibilidade orçamentário-financeira.

**§2º.** Os cargos em comissão e as funções de confiança das unidades acima referidas são apenas os previstos no Quadro I, do Anexo VII, desta Lei.

## **TÍTULO VII** **DOS CARGOS CRIADOS E EXTINTOS**

### **CAPÍTULO I** **DOS CARGOS CRIADOS**

**Art. 68.** Ficam criados, no quadro de pessoal do Poder Judiciário do Estado do Piauí, os cargos de provimento efetivo constantes do Anexo II, desta Lei e os cargos em comissão e as funções de confiança constantes do Quadro I, do Anexo VII, desta Lei.

### **CAPÍTULO II** **DOS CARGOS EXTINTOS E EM QUADRO DE EXTINÇÃO**

**Art. 69.** Ficam extintos os seguintes cargos efetivos em razão de vacância e de sua desnecessidade na estrutura administrativa do Poder Judiciário Estadual:

- I** – Taquígrafo e Oficial de Imprensa;
- II** – Auxiliar Administrativo.

**Art. 70.** Compõem quadro em extinção os atuais ocupantes dos seguintes cargos:

- I** – Oficial de Transporte, Operador de Som, Técnico em Eletricidade, Técnico em Enfermagem, Técnico em Informática, Técnico Gráfico, Telefonista, Técnico Administrativo e Técnico em Contabilidade;
- II** – Bombeiro Hidráulico, Marceneiro e Pedreiro;
- III** – Atendente judiciário e oficial judiciário

## **TÍTULO VIII** **DAS DISPOSIÇÕES FINAIS E TRANSITÓRIAS**

**Art. 71.** Fica vedada qualquer modalidade de provimento que propicie ao servidor investir-se, sem prévia aprovação em concurso público destinado ao seu provimento, em cargo que não integra a carreira na qual anteriormente investido.



**Art. 72.** Aos policiais militares em efetivo exercício nas repartições do Poder Judiciário estadual permanece a gratificação prevista no Anexo XI, desta Lei, que não se incorpora aos proventos de inatividade.

**Art. 73.** A implantação do Plano de Carreiras e Remuneração previsto nesta Lei, com enquadramento dos atuais servidores abrangidos, será feita por ato da Presidência do TJ/PI.

**Art. 74.** Os efeitos financeiros desta Lei ficam condicionados ao atendimento dos requisitos previstos na Lei de Responsabilidade Fiscal – Lei Complementar nº 101, de 04 de maio de 2000 e a disponibilidade orçamentária.

**Art. 75.** VETADO.

**Art. 76.** Esta Lei entra em vigor na data da sua publicação, observadas as regras de transição.

PALÁCIO DE KARNAK, em Teresina (PI), 29 de novembro de 2017.

***JOSÉ WELLINGTON BARROSO DE ARAÚJO DIAS***  
GOVERNADOR DO ESTADO

***MERLONG SOLANO NOGUEIRA***  
SECRETÁRIO DE GOVERNO



PODER JUDICIÁRIO  
TRIBUNAL DE JUSTIÇA DO ESTADO DO PIAUÍ  
**SECRETARIA JUDICIÁRIA**  
**COORDENADORIA JUDICIÁRIA DO PLENO**

Praça Des. Edgard Nogueira, S/N, Centro Cívico  
TERESINA-PI – CEP: 64.000-830  
Fone: (86) 3226 2613 / 86 98876 1487 – E-mail: secretaria.pleno@tjpi.jus.br

---

**ANEXOS**



**ANEXO I**  
**Tabela de transformação dos cargos efetivos do Poder Judiciário**

CARGOS	NÍVEIS	Nº CARGOS	CARREIRA	ÁREA	CARGOS	NÍVEIS	REFERÊNCIAS	Nº CARGOS
Analista Judicial	11 A 15	1276	ANALISTA JUDICIÁRIO	JUDICIÁRIA	ANALISTA JUDICIAL	1A a 6A	I, II, III	1294
Oficial de Justiça e Avaliador	11 A 15	388			OFICIAL DE JUSTIÇA E AVALIADOR	1A a 6A	I, II, III	428
Analista de Sistemas/Banco de Dados	11 A 15	3		APOIO ESPECIALIZADO	ANALISTA DE SISTEMAS/ BANCO DE DADOS	1A a 6A	I, II, III	5
Analista de Sistemas/ Desenvolvimento	11 A 15	50			ANALISTA DE SISTEMAS/ DESENVOLVIMENTO	1A a 6A	I, II, III	50
Analista de Sistemas/ Infraestrutura	11 A 15	3			ANALISTA DE SISTEMAS/ INFRAESTRUTURA	1A a 6A	I, II, III	3
Arquiteto	11 A 15	4*			ARQUITETO	1A a 6A	I, II, III	5
Assistente Social	11 A 15	24			ASSISTENTE SOCIAL	1A a 6A	I, II, III	34
Auditor	11 A 15	14			AUDITOR	1A a 6A	I, II, III	14
Contador	11 A 15	15			CONTADOR	1A a 6A	I, II, III	15
Enfermeiro	11 A 15	4			ENFERMEIRO	1A a 6A	I, II, III	3
Engenheiro Civil	11 A 15	8			ENGENHEIRO CIVIL	1A a 6A	I, II, III	8
Engenheiro Eletricista	11 A 15	2			ENGENHEIRO ELETRICISTA	1A a 6A	I, II, III	2
Médico	11 A 15	12			MÉDICO	1A a 6A	I, II, III	4
Nutricionista	11 A 15	2			NUTRICIONISTA	1A a 6A	I, II, III	2
Odontólogo	11 A 15	8			ODONTÓLOGO	1A a 6A	I, II, III	4
Psicólogo	11 A 15	24		PSICÓLOGO	1A a 6A	I, II, III	34	
Psiquiatra	11 A 15	8		PSIQUIATRA	1A a 6A	I, II, III	7	
					FISIOTERAPEUTA	1A a 6A	I, II, III	2
Analista administrativo	11 A 15	210		ADMINISTRATIVA	ANALISTA ADMINISTRATIVO	1A a 6A	I, II, III	230
Arquivologista	11 A 15	1			ARQUIVOLOGISTA	1A a 6A	I, II, III	1
Bibliotecário	11 A 15	1	BIBLIOTECÁRIO		1A a 6A	I, II, III	1	
Estatístico	11 A 15	2	ESTATÍSTICO		1A a 6A	I, II, III	2	
Atendente Judiciário	11 A 15	9	ATENDENTE JUDICIÁRIO		1A a 6A	I, II, III	9	
Oficial Judiciário	11 A 15	130	OFICIAL JUDICIÁRIO		1A a 6A	I, II, III	130	
Operador de Som	06 A 10	6	OPERADOR DE SOM		1B a 5B	I, II, III	0	
Técnico em Contabilidade	06 A 10	5	TÉCNICO JUDICIÁRIO	APOIO ESPECIALIZADO	TÉCNICO EM CONTABILIDADE	1B a 5B	I, II, III	0
Técnico em Eletricidade	06 A 10	3			TÉCNICO EM ELETRICIDADE	1B a 5B	I, II, III	0
Técnico em Informática	06 A 10	15			TÉCNICO EM INFORMÁTICA	1B a 5B	I, II, III	0
Técnico Gráfico	06 A 10	4		TÉCNICO GRÁFICO	1B a 5B	I, II, III	0	
Oficial de Transporte	06 A 10	17		ADMINISTRATIVA	OFICIAL DE TRANSPORTE	1B a 5B	I, II, III	0
Telefonista	06 A 10	3	TELEFONISTA		1B a 5B	I, II, III	0	
Técnico Administrativo	06 A 10	126	ADMINISTRATIVA	TÉCNICO ADMINISTRATIVO	1B a 5B	I, II, III	0	
Bombeiro Hidráulico	01 A 05	2	AUXILIAR JUDICIÁRIO	ADMINISTRATIVA	BOMBEIRO HIDRÁULICO	1C a 5C	I, II, III	0
Marceneiro	01 A 05	3			MARCENEIRO	1C a 5C	I, II, III	0
Pedreiro	01 A 05	2			PEDREIRO	1C a 5C	I, II, III	0
<b>TOTAL</b>		<b>2270</b>	<b>TOTAL</b>					<b>2.287</b>

**ANEXO II**  
**Quadro Geral de Cargos de Provedimento Efetivo do Poder Judiciário**

CARREIRAS	ÁREAS	CARGOS	NÍVEIS	REFERÊNCIAS	Nº DE CARGOS
ANALISTA JUDICIÁRIO	JUDICIÁRIA	ANALISTA JUDICIAL	1A a 6A	I, II, III	1.294
		OFICIAL DE JUSTIÇA E AVALIADOR	1A a 6A	I, II, III	428
	APOIO ESPECIALIZADO	ANALISTA DE SISTEMAS/BANCO DE DADOS	1A a 6A	I, II, III	5
		ANALISTA DE SISTEMAS/ DESENVOLVIMENTO	1A a 6A	I, II, III	50
		ANALISTA DE SISTEMAS/ INFRAESTRUTURA	1A a 6A	I, II, III	3
		ARQUITETO	1A a 6A	I, II, III	5
		ASSISTENTE SOCIAL	1A a 6A	I, II, III	34
		AUDITOR	1A a 6A	I, II, III	14
		CONTADOR	1A a 6A	I, II, III	15
		ENFERMEIRO	1A a 6A	I, II, III	3
		ENGENHEIRO CIVIL	1A a 6A	I, II, III	8
		ENGENHEIRO ELETRICISTA	1A a 6A	I, II, III	2
		FISIOTERAPEUTA	1A a 6A	I,II, III	2
		MÉDICO	1A a 6A	I, II, III	4
		NUTRICIONISTA	1A a 6A	I, II, III	2
		ODONTÓLOGO	1A a 6A	I, II, III	4
		PSICÓLOGO	1A a 6A	I, II, III	34
		PSQUIATRA	1A a 6A	I, II, III	7
		ADMINISTRATIVA	ANALISTA ADMINISTRATIVO	1A a 6A	I, II, III
	ARQUIVOLOGISTA		1A a 6A	I, II, III	1
	BIBLIOTECÁRIO		1A a 6A	I, II, III	1
	ESTATÍSTICO		1A a 6A	I, II, III	2
	ATENDENTE JUDICIARIO		1A a 6A	I, II, III	9
OFICIAL JUDICIÁRIO	1A a 6A		I, II, III	130	
TÉCNICO JUDICIÁRIO	APOIO ESPECIALIZADO	OPERADOR DE SOM	1B a 5B	I, II, III	0
		TÉCNICO EM CONTABILIDADE	1B a 5B	I, II, III	0
		TÉCNICO EM ELETRICIDADE	1B a 5B	I, II, III	0
		TÉCNICO EM INFORMÁTICA	1 B a 5B	I, II, III	0
		TÉCNICO GRÁFICO	1B a 5B	I, II, III	0

	ADMINISTRATIVA	OFICIAL DE TRANSPORTE	1B a 5B	I, II, III	0
		TELEFONISTA	1B a 5B	I, II, III	0
		TÉCNICO ADMINISTRATIVO	1B a 5B	I, II, III	0
AUXILIAR JUDICIÁRIO	ADMINISTRATIVA	BOMBEIRO HIDRÁULICO	1C a 5C	I, II, III	0
		MARCENEIRO	1C a 5C	I, II, III	0
		PEDREIRO	1C a 5C	I, II, III	0

**ANEXO III**  
**Quadro de Atribuições dos Cargos de Provisão Efetivo do Poder Judiciário**

CARGO EFETIVO	ATRIBUIÇÕES
ANALISTA JUDICIAL	<p>a) confeccionar minutas, atos ordinatórios, emitir informações e manifestações;</p> <p>b) verificar a regularidade formal e legal de petições e processos;</p> <p>c) realizar estudos e pesquisas na legislação, na jurisprudência e na doutrina pertinente para fundamentar a análise de processo;</p> <p>d) fornecer suporte técnico jurídico aos magistrados, órgãos julgadores e unidades do Tribunal, auxiliando na elaboração de minutas e decisões;</p> <p>e) realizar serviços de natureza judiciária na respectiva área de atuação, envolvendo matéria que exija conhecimentos jurídicos;</p> <p>f) inserir, atualizar e consultar informações em base de dados;</p> <p>g) verificar e certificar decurso de prazos processuais;</p> <p>h) atender ao público interno e externo;</p> <p>i) redigir, digitar e conferir expedientes diversos, dar fiel cumprimento às determinações judiciais e executar outras atividades de mesma natureza e grau de complexidade.</p>
ANALISTA ADMINISTRATIVO	<p>a) realizar tarefas relacionadas à administração de recursos humanos, materiais, patrimoniais, orçamentários e financeiros, de desenvolvimento organizacional, licitações e contratos, contabilidade e auditoria;</p> <p>b) emitir informações e pareceres; elaborar, analisar e interpretar dados e demonstrativos, implementar, acompanhar e avaliar projetos pertinentes à área de atuação;</p> <p>c) elaborar e aplicar instrumentos de acompanhamento, avaliação, pesquisa, controle e divulgação referentes aos projetos desenvolvidos;</p> <p>d) atender ao público interno e externo;</p> <p>e) redigir, digitar e conferir expedientes diversos;</p> <p>f) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
OFICIAL DE JUSTIÇA E AVALIADOR	<p>a) cumprir, pessoalmente e na forma da lei, as ordens e os atos judiciais;</p> <p>b) lavrar certidões e autos dos mandados na forma da lei;</p> <p>c) certificar, expressamente, a data, a hora e o local do deslocamento para realização completa do ato judicial, mencionando as circunstâncias essenciais relacionadas à sua execução, e as situações adversas que dificultaram ou impediram o cumprimento do ato;</p> <p>d) retirar, diariamente, os mandados;</p> <p>e) cumprir os mandados dentro do prazo legal, salvo se outro não for estipulado pela autoridade ou por norma;</p> <p>f) devolver os mandados de intimação para audiência, previamente antes da data designada para a audiência;</p> <p>g) cumprir a escala de plantão;</p> <p>h) avaliar os bens penhorados e/ou arrestados nos atos processuais, sem prejuízo de outras avaliações e atribuições compatíveis com sua função determinadas pelo superior hierárquico.</p>

<p style="text-align: center;">ANALISTA DE SISTEMAS/BANCO DE DADOS</p>	<ul style="list-style-type: none"> <li>a) realizar atividades de nível superior que envolvam a elaboração de projetos para criação e manutenção de banco de dados corporativo, planejando seu <i>layout</i> físico e lógico;</li> <li>b) instalar, configurar, gerenciar, monitorar e adequar o funcionamento de sistemas gerenciadores de bancos de dados;</li> <li>c) criar estratégias de auditoria e melhoria da performance do banco de dados, realizando a instalação de <i>upgrades</i>, <i>downgrades</i>, <i>patches</i> e <i>releases</i>, incluindo a realização de atividades de <i>backup</i> e <i>restore</i>;</li> <li>d) planejar, coordenar e executar as migrações de dados de sistemas, bem como efetuar replicação e atualização de bases de dados em produção para desenvolvimento por meio de importações/exportações de bancos de dados;</li> <li>e) monitorar as aplicações, efetuando ajustes de desempenho (<i>tunning</i>) de aplicação e banco de dados, propondo ajustes de melhorias nos programas e aplicações;</li> <li>f) monitorar a utilização de memória, processador, acesso a discos, volume de dados dos bancos de dados;</li> <li>g) prestar suporte técnico a usuários e desenvolvedores;</li> <li>h) emitir pareceres técnicos, relatórios, informações e outros documentos oficiais;</li> <li>i) elaborar documentação técnica relativa aos procedimentos e controles;</li> <li>j) elaborar especificações técnicas de bens e serviços de tecnologia da informação relacionados a sua área de atuação e termo de referência para contratações de T.I.C.;</li> <li>k) gerir contratos com fornecedores de bens e serviços de tecnologia da informação;</li> <li>l) realizar atividades que exijam conhecimentos específicos e aprofundados em informática.</li> </ul>
<p style="text-align: center;">ANALISTA DE SISTEMAS/ DESENVOLVIMENTO</p>	<ul style="list-style-type: none"> <li>a) realizar atividades de nível superior que envolvam a gestão de informação, análise e diagnóstico das necessidades dos usuários;</li> <li>b) coordenar e gerir processos de desenvolvimento de sistemas;</li> <li>c) estabelecer e monitorar a utilização de normas e padrões para o desenvolvimento de sistemas;</li> <li>d) elaborar projetos de sistemas de informação de acordo com a metodologia de desenvolvimento de sistemas vigente;</li> <li>e) levantar e especificar os casos de uso, utilizando artefatos definidos na metodologia;</li> <li>f) construir protótipos de telas e sistemas;</li> <li>g) elaborar, implementar e testar os códigos de programas de acordo com o plano de testes dos sistemas;</li> <li>h) produzir documentação necessária para os usuários de sistemas de informação;</li> <li>i) prestar o assessoramento técnico no que se refere a prazos, recursos e alternativas de desenvolvimento de sistemas, efetuando a prospecção, análise e implementação de novas ferramentas de desenvolvimento;</li> <li>j) efetivar treinamentos relativos à utilização dos sistemas de informação, ferramentas de acesso e manipulação de dados;</li> <li>k) realizar alterações, manutenções e adequações necessárias ao bom funcionamento dos sistemas;</li> <li>l) acompanhar e avaliar o desempenho dos sistemas implantados, identificando e providenciando as medidas corretivas competentes;</li> <li>m) planejar estrategicamente e analisar os sistemas de informações;</li> <li>n) administrar os componentes reutilizáveis e repositórios;</li> <li>o) monitorar a certificação e inspecionar os modelos e códigos de sistemas;</li> <li>p) elaborar e manter o modelo corporativo de dados;</li> <li>q) administrar os dados;</li> <li>r) elaborar especificações técnicas de bens e serviços de tecnologia da informação relacionados a sua área de atuação e termo de referência para contratações de T.I.C.;</li> <li>s) gerir os contratos com fornecedores de bens e serviços de tecnologia da informação;</li> <li>t) realizar atividades que exijam conhecimentos específicos e aprofundados de tecnologia da informação;</li> <li>u) desenvolver outras atribuições de mesma natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</li> </ul>

<p>ANALISTA DE SISTEMAS/ INFRAESTRUTURA</p>	<p>a) realizar atividades de nível superior que envolvam o projeto de redes de computadores, definindo a topologia e a configuração necessária;</p> <p>b) instalar, customizar e manter os recursos de rede;</p> <p>c) analisar a utilização e o desempenho das redes de computadores, identificando os problemas e promovendo as correções no ambiente operacional;</p> <p>d) planejar a evolução da rede, visando à melhoria na qualidade dos serviços;</p> <p>e) prestar o suporte técnico e consultoria relativamente à aquisição, à implantação e ao uso adequados dos recursos de rede;</p> <p>f) avaliar e especificar as necessidades de <i>hardware</i> e <i>software</i> básico e de apoio;</p> <p>g) configurar ambientes operacionais,</p> <p>h) instalar, customizar e manter <i>software</i> básico e de apoio;</p> <p>i) analisar o desempenho do ambiente operacional, efetuando as adequações necessárias;</p> <p>j) analisar a utilizar os recursos de <i>software</i> e <i>hardware</i> e o planejamento da evolução do ambiente, visando à melhoria na qualidade do serviço;</p> <p>k) prestar consultoria e suporte técnico relativamente à aquisição, implantação e uso adequado dos recursos de <i>hardware</i> e <i>software</i>;</p> <p>l) prospectar, analisar e implementar novos recursos de <i>hardware</i>, <i>software</i> e rede, visando à sua utilização na organização;</p> <p>m) analisar a viabilidade de instalação de novas aplicações no ambiente operacional da organização, objetivando manter o padrão de desempenho de serviços implantados;</p> <p>n) desenvolver sistemáticas, estudos, normas, procedimentos e padronização das características técnicas, visando à melhoria da segurança e dos serviços prestados;</p> <p>o) elaborar especificações técnicas de bens e serviços de tecnologia da informação relacionados a sua área de atuação e termo de referência para contratações de T.I.C.;</p> <p>p) gerir contratos com fornecedores de bens e de serviços de tecnologia da informação;</p> <p>q) realizar atividades que exijam conhecimentos específicos e aprofundados de tecnologia da informação, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
<p>ARQUITETO</p>	<p>a) realizar estudo e análise de <i>interfaceamento</i> de projetos;</p> <p>b) elaborar e acompanhar a execução de projetos arquitetônicos dentro do complexo arquitetônico dos prédios do Poder Judiciário;</p> <p>c) elaborar relatórios referentes ao andamento, execução e finalização das obras executadas pelo Poder Judiciário;</p> <p>d) providenciar e manter atualizado os levantamentos das condições físicas dos imóveis pertencentes ao Poder Judiciário;</p> <p>e) fiscalizar e acompanhar as atividades das obras de construção, manutenção, ampliação, reforma e executar o projeto e operacionalização referente aos serviços de arquitetura;</p> <p>f) propor a elaboração e a aplicação de normas e de procedimentos técnicos na sua área de atuação;</p> <p>g) exercer outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
<p>ASSISTENTE SOCIAL</p>	<p>a) avaliar os casos, elaborando estudo ou perícia social, com a finalidade de subsidiar ou assessorar a autoridade judiciária no conhecimento dos aspectos socioeconômicos, culturais, interpessoais, familiares, institucionais, comunitários e outros;</p>

	<p>b) emitir laudos técnicos, pareceres e resposta a quesitos, por escrito ou verbalmente, em audiências e ainda realizar acompanhamento e reavaliação de casos;</p> <p>c) desenvolver ações de aconselhamento, orientação, encaminhamento, prevenção e outros, no que se refere às questões sociojurídicas;</p> <p>d) desenvolver atividades específicas em quaisquer unidades jurisdicionais;</p> <p>e) estabelecer e aplicar procedimentos técnicos de mediação junto ao grupo familiar em situação de conflito;</p> <p>f) contribuir e/ou participar de trabalhos que visem à integração do Poder Judiciário com as instituições que desenvolvam ações na área social, buscando a articulação com a rede de atendimento à infância, juventude e família, para o melhor encaminhamento;</p> <p>g) acompanhar visitas de pais às crianças, em casos excepcionais, quando determinado judicialmente;</p> <p>h) fiscalizar instituições e/ou programas que atendam criança e adolescente sob medida protetiva e/ou em cumprimento de medida socioeducativa, quando da determinação judicial, em conformidade com a legislação vigente;</p> <p>i) realizar trabalhos junto à equipe multiprofissional com objetivo de atender à solicitação de estudo psicossocial;</p> <p>j) elaborar mensal e anualmente relatório estatístico, quantitativo e qualitativo sobre as atividades desenvolvidas, bem como pesquisas e estudos, com vistas a manter e melhorar a qualidade do trabalho;</p> <p>k) supervisionar estágio de alunos do curso de serviço social, mediante prévia autorização do Tribunal de Justiça;</p> <p>l) planejar e coordenar as atividades técnicas e administrativas específicas do setor social;</p> <p>m) elaborar e manter atualizado cadastro de recursos da comunidade;</p> <p>n) elaborar, implementar, coordenar, executar e avaliar, controlando e fiscalizando se necessário, planos, programas e projetos que sejam do âmbito de atuação do serviço social, de acordo com as diretrizes fixadas pelo Tribunal, nos serviços de atendimento a magistrados e servidores;</p> <p>o) assessorar a alta Administração, sempre que necessário, nas questões relativas à matéria do serviço social.</p> <p>p) atender a todas as determinações judiciais relativas à prática do serviço social.</p>
AUDITOR	<p>a) desenvolver atividades de controle interno;</p> <p>b) realizar auditoria, prestações de contas e outros trabalhos correlatos nas diversas unidades, departamentos, coordenadorias e seções do Poder Judiciário;</p> <p>c) elaborar relatórios, manuais, pareceres, certificados, notas técnicas e estudos, no exercício das atividades de controle interno relacionadas à fiscalização e à avaliação;</p> <p>d) assessorar a Presidência do Tribunal de Justiça junto ao Tribunal de Contas do Estado;</p> <p>e) realizar ações segregadas de auditoria e de controle interno, com atuação direta nos processos e procedimentos, <i>a priori</i> e <i>a posteriori</i>;</p> <p>f) analisar demonstrações financeiras, planilhas de formação de preço relativos aos processos de Licitações e Contratos;</p> <p>g) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
CONTADOR	<p>a) examinar planos de contas do Poder Judiciário;</p> <p>b) realizar cálculos necessários à liquidação de julgados e à atualização de valores de títulos, guias e depósitos judiciais;</p> <p>c) realizar conciliação bancária e analisar demonstrações financeiras dos órgãos do Poder Judiciário;</p> <p>d) realizar escrituração, demonstrações e análises contábeis conforme definido pelo Conselho Federal de Contabilidade</p> <p>e) realizar atividades que exijam conhecimentos específicos e aprofundados de contabilidade, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
ENFERMEIRO	<p>a) realizar atividades a fim de promover e de preservar a saúde de magistrados, servidores e seus dependentes;</p>

	<p>b) planejar, organizar, supervisionar e/ou executar serviços de enfermagem, tais como a realização de serviços emergenciais, o acompanhamento a pacientes, a manutenção dos prontuários atualizados, o controle do estoque e das condições de uso dos materiais, equipamentos, medicamentos, soluções, aparelhos e instrumentos utilizados no atendimento;</p> <p>c) planejar e/ou participar de programas de saúde e da elaboração de relatórios;</p> <p>d) realizar atividades que exijam conhecimentos específicos e aprofundados de enfermagem, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
ENGENHEIRO CIVIL	<p>a) realizar atividades de nível superior a fim de garantir os padrões de qualidade técnica e de segurança das obras e reparos de edificações, bem como a adequada manutenção de instalações;</p> <p>b) planejar e elaborar orçamentos, projetos e especificações nas obras e serviços de engenharia do Poder Judiciário;</p> <p>c) elaborar laudos e termos de referências na área de engenharia necessários à contratação de obras e serviços respectivos;</p> <p>d) realizar a avaliação de imóveis para fins de aquisição, alienação e locação;</p> <p>e) assessorar a comissão de licitação na contratação de obras e serviços de engenharia, bem como acompanhar e fiscalizar a execução dos serviços contratados;</p> <p>f) realizar atividades que exijam conhecimentos específicos e aprofundados de engenharia, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
ENGENHEIRO ELETRICISTA	<p>a) realizar atividades a fim de garantir os padrões de qualidade técnica na geração, transmissão, distribuição e utilização de energia elétrica, bem como a adequada manutenção e reparo das instalações;</p> <p>b) planejar e elaborar orçamentos, especificações, projetos elétricos e termos de referência;</p> <p>c) assessorar a comissão de licitação na contratação de obras e serviços de engenharia elétrica, bem como acompanhamento e fiscalização da execução dos serviços contratados;</p> <p>d) realizar atividades que exijam conhecimentos específicos e aprofundados de engenharia elétrica, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
FISIOTERAPEUTA	<p>a) elaborar diagnóstico fisioterapêutico;</p> <p>b) realizar atividades destinadas à promoção, ao tratamento e à recuperação da saúde de pacientes mediante a aplicação de técnicas fisioterapêuticas;</p> <p>c) realizar atividades que exijam conhecimentos específicos e aprofundados de fisioterapia;</p> <p>d) redigir, digitar e conferir expedientes diversos e executar outras atividades de mesma natureza e grau de complexidade que venham a ser determinadas pela autoridade superior.</p>
MÉDICO	<p>a) prestar assistência médica aos magistrados, servidores e seus dependentes;</p> <p>b) coordenar campanhas preventivas de saúde pública;</p> <p>c) elaborar laudos, pareceres técnicos, executar perícias em juntas médicas, em especial para fim de aposentadoria e licença;</p> <p>d) controlar as condições de uso dos equipamentos e aparelhos utilizados no atendimento médico;</p> <p>e) realizar atividades que exijam conhecimentos específicos e aprofundados de medicina, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
NUTRICIONISTA	<p>a) prestar assistência especializada a magistrados e a servidores, ativos e inativos, bem como aos respectivos dependentes;</p> <p>b) coordenar campanhas de reeducação alimentar;</p> <p>c) realizar atividades que exijam conhecimentos específicos e aprofundados de nutrição, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior;</p> <p>d) acompanhar e fiscalizar a execução dos serviços contratados relacionados a sua área de atuação.</p>
ODONTÓLOGO	<p>a) realizar atividades com o fim de promover e preservar a saúde bucal de magistrados, servidores e dependentes;</p>



	<p>b) diagnosticar e tratar afecções da cavidade oral;</p> <p>c) executar perícias, além de elaborar e aplicar medidas preventivas relativas à saúde bucal e geral;</p> <p>d) controlar o estoque e uso de equipamentos, aparelhos, materiais, instrumentos, medicamentos e soluções utilizados para atendimento odontológico;</p> <p>e) realizar atividades que exijam conhecimentos específicos e aprofundados de odontologia, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
PSICÓLOGO	<p>a) proceder à avaliação de crianças, adolescentes, adultos e idosos, elaborando o estudo psicológico, com a finalidade de subsidiar ou assessorar a autoridade judiciária no conhecimento dos aspectos psicológicos de sua vida familiar, institucional e comunitária, para que o magistrado decida e ordene as medidas cabíveis;</p> <p>b) exercer atividades no campo da psicologia jurídica, numa abordagem clínica, realizando entrevistas psicológicas, individuais, grupais, de casal e família, além de devolutivas;</p> <p>c) aplicar técnicas psicométricas e projetivas, observação lúdica de crianças, crianças/pais, para compreender e analisar a problemática apresentada elaborando um prognóstico;</p> <p>d) propor procedimentos a serem aplicados;</p> <p>e) realizar estudos de campo, através de visitas domiciliares, em abrigos, internatos, escolas e outras instituições, buscando uma discussão multiprofissional, intraequipe e extraequipe, para realizar o diagnóstico situacional e a compreensão da psicodinâmica das pessoas implicadas na problemática judicial em estudo;</p> <p>f) proceder ao encaminhamento para psicodiagnóstico, terapia e atendimento especializado (escolar, fonoaudiológico, etc);</p> <p>g) realizar o acompanhamento de casos objetivando a clareza para definição da medida, avaliando a adaptação criança/família, reavaliando e constatando a efetivação de mudanças, verificando se os encaminhamentos a recursos sociais e psicológicos oferecidos na comunidade, e a aplicação das medidas de proteção e socioeducativas foram efetivados;</p> <p>h) aplicar técnicas de orientação, aconselhamento individual, casal e de família;</p> <p>i) fornecer subsídios por escrito (em processo judicial) ou verbalmente (em audiência), emitir laudos, pareceres e responder a quesitos;</p> <p>j) executar o cadastramento de casais interessados em adoção, de crianças adotáveis, crianças e adolescentes acolhidos, de recursos e programas comunitários psicossociais e de áreas afins (educação, saúde, cultura e lazer), além de treinamento de famílias de apoio, visando a reinserção à família biológica ou substituta;</p> <p>k) promover a prevenção e controle da violência intrafamiliar e extrafamiliar, institucional contra crianças e adolescentes e de condutas infracionais e violência doméstica e familiar contra a mulher e contra o idoso;</p> <p>l) ministrar supervisão de estagiários de psicologia;</p> <p>m) elaborar pesquisas e estudos, ampliando o conhecimento psicológico na área do direito e da psicologia judiciária, levantando o perfil dos atendidos e dos psicólogos e assistentes sociais;</p> <p>n) fornecer indicadores para formulação de programas de atendimento, relacionados a medidas de proteção socioeducativas, na área da Justiça da Infância e Juventude, auxiliando na elaboração de políticas públicas, relativas à família, à infância e à juventude;</p> <p>o) realizar atividades que exijam conhecimentos específicos e aprofundados de psicologia, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>

MÉDICO PSIQUIATRA	<p>a) participar de equipe multidisciplinar e executar atividades relativas ao planejamento operacional e execução, acompanhamento e avaliação de projetos, planos, programas, campanhas, estudos, encontros, cursos e eventos em geral;</p> <p>b) instruir expedientes administrativos, elaborar relatórios, pareceres médicos, informações, pareceres técnicos e outros instrumentos que forneçam dados para decisões superiores;</p> <p>c) anotar em prontuários os atos médicos realizados, registrando inclusive impressão diagnóstica e tratando, quando for o caso;</p> <p>d) preencher e visar mapas de atendimento, com fins de elaborar relatórios mensais para consolidação de dados estatísticos;</p> <p>e) realizar exames criminológicos, bem como desempenhar outras atividades correlatas ou outras atribuições que possam vir a surgir, da mesma natureza e nível de complexidade, conforme as necessidades da área;</p> <p>f) realizar atividades que exijam conhecimentos específicos e aprofundados de psiquiatria, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
ARQUIVOLOGISTA	<p>a) organizar arquivos de documentos, livros, fotografias e tudo o que diz respeito à memória do Poder Judiciário;</p> <p>b) promover a adequada preservação de documentos e gestão de arquivos;</p> <p>c) planejar, organizar, coordenar e controlar os procedimentos e operações técnicas para produção, tramitação, utilização, avaliação e arquivamento de documentos;</p> <p>d) realizar estudos, atendimento a usuários e divulgação do acervo;</p> <p>e) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
BIBLIOTECÁRIO	<p>a) documentar, catalogar, classificar, indexar livros, teses, bibliografias, jurisprudências e outros documentos;</p> <p>b) orientar consultas em pesquisas bibliográficas e escolhas de publicações;</p> <p>c) treinar pessoal para catalogação e elaborar estudos e projetos para geração e manutenção de bases de dados, conservação e desenvolvimento do acervo e modernização dos serviços;</p> <p>d) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
ESTATÍSTICO	<p>a) planejar e desenvolver investigações estatísticas;</p> <p>b) coordenar os trabalhos de coleta, análise e interpretação de dados; elaborar parecer, instrumentais técnicos, pesquisas organizacionais, pesquisas de satisfação, laudos e relatórios;</p> <p>c) fornecer informações que favoreçam a tomada de decisões e o acompanhamento da execução de atividades;</p> <p>d) acompanhar e analisar, sistematicamente, a legislação relacionada com a sua área de atuação;</p> <p>e) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior;</p> <p>f) consolidação dos dados estatísticos para o Conselho Nacional de Justiça.</p>
OPERADOR DE SOM	<p>a) operar os serviços de som nas sessões do Tribunal de Justiça;</p> <p>b) instalar e testar os equipamentos de som a serem utilizados;</p> <p>c) realizar a manutenção preventiva dos equipamentos de som, bem como detectar e resolver os problemas de som que venham a ocorrer;</p> <p>d) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>

TÉCNICO EM CONTABILIDADE	<p>a) realizar práticas contábeis auxiliares;</p> <p>b) participar de trabalhos de tomadas de contas;</p> <p>c) orientar as áreas do Tribunal quanto aos procedimentos de registros contábeis e de classificação da despesa e receita, em observância ao plano de contas vigente;</p> <p>d) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
TÉCNICO EM ELETRICIDADE	<p>a) realizar instalação elétrica nos prédios do Poder Judiciário;</p> <p>b) realizar manutenção preventiva e corretiva em instalações e aparelhos elétricos;</p> <p>c) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
TÉCNICO EM INFORMÁTICA	<p>a) realizar atividades de nível intermediário que envolvam a adequada automatização de rotinas, por intermédio do desenvolvimento, codificação, teste, implantação, documentação e manutenção dos programas e sistemas;</p> <p>b) verificar, preparar e operar os equipamentos de informática, com a transferência de dados para sistemas automatizados;</p> <p>c) atender aos usuários;</p> <p>d) realizar trabalhos que exijam conhecimentos básicos e/ou específicos de informática;</p> <p>e) auxiliar o analista de sistemas.</p> <p>f) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
TÉCNICO GRÁFICO	<p>a) realizar atividades de nível intermediário com a finalidade de zelar pela qualidade da impressão gráfica de livros, folhetos, revistas, cartazes, jornais e folders;</p> <p>b) realizar, verificar e controlar as operações de montagem, encadernação, arte final e impressão;</p> <p>c) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
OFICIAL DE TRANSPORTE	<p>a) conduzir os veículos pertencentes ao Poder Judiciário com zelo e segurança, com habilitação profissional;</p> <p>b) comunicar à chefia imediata sobre a ocorrência de acidentes ou defeitos, além de outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
TÉCNICO ADMINISTRATIVO	<p>a) prestar apoio técnico, administrativo e processual, pertinente às atribuições das unidades organizacionais;</p> <p>b) executar tarefas de apoio à atividade judiciária; arquivar processos e documentos;</p> <p>c) efetuar tarefas relacionadas à movimentação de processos nos sistemas processuais judiciais e administrativos, zelar pela guarda de processos e documentos; atender ao público interno e externo;</p> <p>d) classificar e autuar processos; redigir, digitar e conferir expedientes diversos, mandados, cartas, certidões, ofícios;</p> <p>e) realizar a publicação de despachos, decisões e sentenças; juntadas de documentos, petições, mandados, postagens de correspondências;</p> <p>f) atender ao público em geral e atendimento aos oficiais de justiça, protocolizando os mandados entregues e devolvidos;</p> <p>g) controlar os registros de protocolos em geral;</p> <p>h) exercer outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
TELEFONISTA	<p>a) garantir a adequada transmissão e recebimento de mensagens via telefone;</p> <p>b) verificar a manutenção e a utilização correta dos equipamentos e zelar por sua limpeza e conservação;</p> <p>c) exercer outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>

BOMBEIRO HIDRÁULICO	<p>a) efetuar serviços preventivos e corretivos no sistema hidráulico dos prédios do Poder Judiciário;</p> <p>b) fazer a manutenção dos equipamentos hidráulicos;</p> <p>c) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
MARCENEIRO	<p>a) garantir a adequada confecção e reparo de móveis e peças de madeira ou efetuar reparos em portas e móveis e peças de madeira;</p> <p>b) realizar a verificação e o controle da qualidade dos serviços de construção e montagem de quadros de aviso, tablados de madeira, estantes e divisórias e de conserto de móveis e peças de madeira, tais como portas, estantes, mesas, balcões, lambris, revestimentos em fórmica;</p> <p>c) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>
PEDREIRO	<p>a) executar reparos e trabalhos de alvenaria, concreto e outros materiais de construção em edifícios do Poder Judiciário;</p> <p>b) colocar telhas, azulejos e ladrilhos e executar outras tarefas, seguindo instruções de superiores;</p> <p>c) realizar outras atribuições de natureza e grau de complexidade correlatos que venham a ser determinadas pela autoridade superior.</p>

**ANEXO IV**  
**Níveis de escolaridade exigidos para provimento de cargos efetivos**

NÍVEIS DE ESCOLARIDADE	CURSOS	CARGO EFETIVO
NÍVEL SUPERIOR	Bacharelado em Direito	Analista Judicial Oficial de Justiça e Avaliador
	Bacharelado ou Licenciatura em geral	Analista Administrativo
	Curso de Nível Superior em Tecnologia da Informação	Analista de Sistemas/ Desenvolvimento
		Analista de Sistemas/Banco de Dados
		Analista de Sistemas/ Infraestrutura
	Arquitetura	Arquiteto
	Arquivologia	Arquivologista
	Biblioteconomia	Bibliotecário
	Administração, Ciências Atuariais, Ciências Contábeis, Direito, Economia, Engenharia Civil e Informática	Auditor
	Ciências Contábeis	Contador
	Enfermagem	Enfermeiro
	Engenharia Civil	Engenheiro Civil
	Engenharia Elétrica	Engenheiro Eletricista
	Estatística	Estatístico
	Fisioterapia	Fisioterapeuta
	Medicina, podendo ser exigida habilitação específica	Médico
	Nutrição	Nutricionista
	Odontologia	Odontólogo
	Psicologia	Psicólogo
	Serviço Social	Assistente Social
Bacharelado ou licenciatura em geral	Atendente judiciário (quadro em extinção)	
Bacharelado ou licenciatura em geral	Oficial judiciário (quadro em extinção)	
NÍVEL MÉDIO	Curso de ensino médio	Técnico Administrativo (quadro em extinção)
		Técnico Gráfico (quadro em extinção)
		Técnico em Operador de Som (quadro em extinção)
		Telefonista (quadro em extinção)
		Oficial de Transporte (quadro em extinção)
	Curso de nível médio em Técnico em Contabilidade	Técnico em Contabilidade (quadro em extinção)
Curso de ensino médio profissionalizante na área de tecnologia da informação ou Curso de ensino médio mais curso de Técnico em Informática	Técnico em Informática (quadro em extinção)	
Curso de nível médio em Técnico em Eletricidade	Técnico em Eletricidade (quadro em extinção)	
NÍVEL FUNDAMENTAL	Curso de ensino fundamental	Bombeiro Hidráulico (quadro em extinção)
		Marceneiro (quadro em extinção)
		Pedreiro (quadro em extinção)

**ANEXO V**  
**Subsídio de servidores ocupantes de cargos de provimento efetivo**

CARREIRA	NÍVEL	REFERÊNCIA	SUBSÍDIO
AUXILIAR JUDICIÁRIO	1C	I	1.468,04
		II	1.538,50
		III	1.612,35
	2C	I	1.689,74
		II	1.770,85
		III	1.855,85
	3C	I	1.944,94
		II	2.038,30
		III	2.136,13
	4C	I	2.238,67
		II	2.346,11
		III	2.458,73
	5C	I	2.576,74
		II	2.700,43
		III	2.830,04
TÉCNICO JUDICIÁRIO	1B	I	2.965,90
		II	3.108,25
		III	3.257,46
	2B	I	3.413,81
		II	3.577,67
		III	3.749,40
	3B	I	3.929,37
		II	4.117,99
		III	4.315,65
	4B	I	4.522,81
		II	4.739,89
		III	4.967,41
	5B	I	5.205,84
		II	5.455,73
		III	5.717,59
ANALISTA JUDICIÁRIO	1A	I	5.992,05
		II	6.279,67
		III	6.581,09
	2A	I	6.896,98
		II	7.228,04
		III	7.574,98
	3A	I	7.938,58
		II	8.319,64
		III	8.718,97
	4A	I	9.137,49
		II	9.576,08
		III	10.035,73

	5A	I	10.517,46
		II	11.022,29
		III	11.551,37
	6A	I	12.105,83
		II	12.686,90
		III	13.295,87

**ANEXO VI**  
**Vantagens devidas aos servidores do Poder Judiciário**

<b>VANTAGENS</b>		<b>VALOR (R\$)</b>
INDENIZAÇÕES	Indenização de Transporte	1.728,00
	Auxílio Alimentação	1.440,00
	Auxílio Saúde	432,00
ADICIONAIS	Adicional de insalubridade	432,00
	Adicional de periculosidade	432,00



**ANEXO VII**  
**Disposições sobre cargos em comissão e funções de confiança**

**Quadro I**  
**Quadro Geral de cargos em comissão e funções de confiança do Poder Judiciário**

<b>Símbolo</b>	<b>CARGOS EM COMISSÃO</b>
CC/01	SECRETÁRIOS
CC/02	CONSULTOR JURÍDICO
	SUPERINTENDENTE DO FERMOJUPI
	SUPERINTENDENTE DE ENGENHARIA E ARQUITETURA
	SUPERINTENDENTE DE LICITAÇÕES E CONTRATOS
	SUPERINTENDENTE DE GESTÃO DE CONTRATOS E CONVÊNIOS
	SUPERINTENDENTE DA JUSTIÇA ITINERANTE
	SUPERINTENDENTE DE SAÚDE E QUALIDADE DE VIDA
	SUPERINTENDENTE ADMINISTRATIVO DA EJUD
	SUPERINTENDENTE DE CONTROLE INTERNO
SUPERINTENDENTE DE SEGURANÇA	
CC/03	ASSESSOR DE MAGISTRADO
	ASSESSOR JUDICIÁRIO
	ASSESSOR ADMINISTRATIVO
	ASSESSOR DE COMUNICAÇÃO SOCIAL
CC/04	ASSISTENTE DE CERIMONIAL
	ASSISTENTE DE IMPRENSA E DIVULGAÇÃO
	ASSISTENTE DE SEGURANÇA
	AUXILIAR DE APOIO JUDICIÁRIO
	COORDENADORES
CC/05	DIRETOR DE SECRETARIA
	SECRETÁRIO DE SESSÃO
	SECRETÁRIO EXECUTIVO
CC/06	AJUDANTE DE ORDEM
	CHEFE DE SEÇÃO
	OFICIAL DE GABINETE DE MAGISTRADO
	OFICIAL DA AUDIÊNCIA DE CUSTÓDIA
	OFICIAL DA CORREGEDORIA DE PRESÍDIOS
<b>Símbolo</b>	<b>FUNÇÃO DE CONFIANÇA</b>
FC/01	DISTRIBUIDOR JUDICIÁRIO DE 1º GRAU – TERESINA
	DISTRIBUIDOR JUDICIÁRIO DE 2º GRAU
FC/02	PRESIDENTE DE COMISSÃO DE LICITAÇÃO
	PRESIDENTE DE COMISSÃO DE PROCESSO ADMINISTRATIVO
	SECRETÁRIO DE VARA
	SECRETÁRIO DA CENTRAL DE INQUÉRITOS
	SECRETÁRIO DO NUPEMEC
	SECRETÁRIO DO CEJUSC
FC/03	PREGOEIRO

	MEMBROS CPPAD
	OFICIAL DE TECNOLOGIA DA INFORMAÇÃO
	OFICIAL DA CEJAI
	SECRETÁRIO ASSISTENTE DE DIRETORIA DE FÓRUM
FC-PM/01	MILITAR I – CORONEL
	MILITAR I – TENENTE CORONEL
	MILITAR I – MAJOR
	MILITAR I – CAPITÃO
	MILITAR I – TENENTE
FC-PM/02	MILITAR II – SUBTENENTE
	MILITAR II – SARGENTO
FC-PM/03	MILITAR III – CABO
	MILITAR III – SOLDADO

**Quadro II**  
**Retribuição pelo exercício de cargo de direção, chefia e assessoramento**

<b>CARGO EM COMISSÃO</b>			
<b>SÍMBOLO</b>	<b>VENCIMENTO (10%)</b>	<b>REPRESENTAÇÃO (90%)</b>	<b>VALOR (Cargo/Função)</b>
CC/01	1.341,17	12.070,51	13.411,68
CC/02	849,41	7.644,60	8.494,01
CC/03	433,49	3.901,40	4.334,89
CC/04	257,78	2.319,94	2.577,72
CC/05	168,39	1.515,52	1.683,91
CC/06	114,06	1.026,52	1.140,58

<b>FUNÇÃO DE CONFIANÇA</b>			
FC/01	-	5.243,06	5.243,06
FC/02	-	2.577,72	2.577,72
FC/03	-	1.026,52	1.026,52
FC-PM/01	-	1.300,00	1.300,00
FC-PM/02	-	1.100,00	1.100,00
FC-PM/03	-	900,00	900,00

**ANEXO VIII**  
**Cargos em Comissão e Funções de Confiança por Unidade**

**Quadro I**

<b>GABINETE DE JUIZ AUXILIAR DA PRESIDÊNCIA</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor Judiciário	CC/03	2
<b>TOTAL</b>		<b>2</b>

**Quadro II**

**(Quadro alterado pela Lei Complementar nº 235/2018, de 12 de junho de 2018)**

<b>SUPERINTENDÊNCIA DE SEGURANÇA (Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Superintendente de Segurança	CC/02	1
Assistente de Segurança	CC/04	<del>25</del> <b>26</b>
Militar I (Coronel, Ten. Coronel, Major, Capitão, Tenentes)	FC-PM/01	9
Militar II (Subtenente e Sargentos)	FC-PM/02	35
Militar III (Cabo e Soldado)	FC-PM/03	109
Ajudante de Ordem	CC/05	2
<b>TOTAL</b>		<del>181</del> <b>182</b>

**Quadro III**

<b>ASSESSORIA DE COMUNICAÇÃO SOCIAL (Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor de Comunicação Social	CC/03	1
Assistente de Imprensa e Divulgação	CC/04	1
Assistente de Cerimonial	CC/04	1
<b>TOTAL</b>		<b>3</b>

**Quadro IV**

<b>GRUPO DE MONITORAMENTO E FISCALIZAÇÃO DO SISTEMA CARCERÁRIO - GMF (Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Coordenador do GMF	CC/04	1
Secretário Executivo	CC/05	1
<b>TOTAL</b>		<b>2</b>

**Quadro V**

<b>COORDENADORIA ESTADUAL JUDICIÁRIA DA INFÂNCIA E DA JUVENTUDE (Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário Executivo	CC/05	1
<b>TOTAL</b>		<b>1</b>

**Quadro VI**

<b>COORDENADORIA ESTADUAL DA MULHER (Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário Executivo	CC/05	1
<b>TOTAL</b>		<b>1</b>

**Quadro VII**

<b>CPPAD – 2º GRAU</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Presidente de Comissão de Processo Administrativo	FC/02	2
Membros da CPPAD – SERVIDORES E CONTRATOS	FC/03	4
<b>TOTAL</b>		<b>6</b>

**Quadro VIII**

<b>NÚCLEO DE MEMÓRIA JUDICIAL (Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Coordenador	CC/04	1
<b>TOTAL</b>		<b>1</b>

**Quadro IX**

<b>NÚCLEO SOCIOAMBIENTAL (Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Coordenador do Núcleo Socioambiental	CC/04	1
<b>TOTAL</b>		<b>1</b>

**Quadro X**

<b>NUPEMEC (Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Coordenador de Políticas Judiciárias de Cidadania	CC/04	1
Coordenador de Modalidades Autocompositivas	CC/04	1
Assessor de Magistrado	CC/03	1
Oficial de Gabinete de Magistrado	CC/06	1
Secretário do NUPEMEC	FC/02	1
<b>TOTAL</b>		<b>5</b>

**Quadro XI**

<b>Supervisão dos Juizados Especiais (Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor Administrativo	CC/03	1
<b>TOTAL</b>		<b>01</b>

**Quadro XII**

<b>NAUJ – Núcleo de Apoio as Unidades Judiciárias de 1º e 2º graus (Vice-Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor Administrativo	CC/03	6
Auxiliar de Apoio Judiciário	CC/04	20
<b>TOTAL</b>		<b>26</b>

**Quadro XIII**

<b>CONSELHO DE SEGURANÇA INSTITUCIONAL</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor Administrativo	CC/03	1
Coordenador	CC/04	1
<b>TOTAL</b>		<b>2</b>

**Quadro XIV**

<b>OUIDORIA JUDICIÁRIA (Ouvidoria)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor Administrativo	CC/03	1
Coordenador	CC/04	1
<b>TOTAL</b>		<b>2</b>

**Quadro XV**

<b>VICE-PRESIDÊNCIA DO TRIBUNAL DE JUSTIÇA</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor Judiciário	CC/03	2
Assessor Administrativo	CC/03	1
Consultor Jurídico	CC/02	4
Assessor de Magistrado	CC/03	4
Oficial de Gabinete de Magistrado	CC/06	1
<b>TOTAL</b>		<b>12</b>

**Quadro XVI**

<b>NUGEP (Vice-Presidência)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Coordenador do NUGEP	CC/04	1
<b>TOTAL</b>		<b>1</b>

**Quadro XVII**

<b>SECRETARIA DE GESTÃO ESTRATÉGICA (SEGES)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário Gestão Estratégica	CC/01	1
Assessor Administrativo	CC/03	1
Assessor Judiciário	CC/03	1
Coordenador do Escritório de Projetos	CC/04	1
Chefe da Seção de Governança	CC/06	1
Chefe da Seção de Análise Estatística	CC/06	1
<b>TOTAL</b>		<b>6</b>

**Quadro XVIII**

<b>SECRETARIA DA PRESIDÊNCIA</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário da Presidência	CC/01	1
Assessor Administrativo	CC/03	2
Consultor Jurídico	CC/02	4
Assessor de Magistrado	CC/03	4
Oficial de Gabinete de Magistrado	CC/06	1
<b>TOTAL</b>		<b>12</b>



**Quadro XIX**

<b>SECRETARIA DE ASSUNTOS JURÍDICOS (SAJ)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário de Assuntos Jurídicos	CC/01	1
Assessor Administrativo	CC/03	2
Assessor Judiciário	CC/03	1
Coordenador Administrativo	CC/04	1
Coordenador Judiciário	CC/04	1
<b>TOTAL</b>		<b>6</b>

**Quadro XX**

<b>SUPERINTENDENCIA DE CONTROLE INTERNO</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Superintendente de Controle Interno	CC/02	1
Assessor Administrativo	CC/03	1
Coordenador de Acompanhamento da Gestão	CC/04	1
Coordenador de Auditoria	CC/04	1
<b>TOTAL</b>		<b>4</b>

**Quadro XXI**

<b>SECRETARIA GERAL</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário Geral	CC/01	1
Assessor Judiciário	CC/03	3
Assessor Administrativo	CC/03	4
<b>TOTAL</b>		<b>8</b>

Quadro XXII

<b>FERMOJUPI - Fundo Especial de Reparcelamento e Modernização do Poder Judiciário do Estado do Piauí</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Superintendente do FERMOJUPI	CC/02	1
Assessor Judiciário	CC/03	1
Assessor Administrativo	CC/03	1
Coordenador de Fiscalizações	CC/04	1
Coordenador de Controle de Receitas	CC/04	1
Coordenador de Controle de Processos Fiscais	CC/04	1
Chefe da Seção de Controle Processual	CC/06	1
Chefe da Seção de Controle de Contas	CC/06	1
Chefe da Seção de Administração do Selo	CC/06	1
<b>TOTAL</b>		<b>9</b>

Quadro XXIII

<b>SUPERINTENDÊNCIA DE LICITAÇÕES E CONTRATOS</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Superintendente de Licitações e Contratos	CC/02	1
Assessor Administrativo	CC/03	1
Assessor Judiciário	CC/03	1
Chefe da Seção de Apoio	CC/06	1
Presidente de Comissão de Licitação	FC/02	2
Pregoeiro	FC/03	2
<b>TOTAL</b>		<b>8</b>

**Quadro XXIV**

<b>SUPERINTENDÊNCIA DE GESTÃO DE CONTRATOS E CONVÊNIOS</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Superintendente de Gestão de Contratos e Convênios	CC/02	1
Assessor Administrativo	CC/03	2
Chefe da Seção de Fiscalização e Contratos e Convênios	CC/06	1
Chefe da Seção de Acompanhamento e Controle de Contratos e Convênios	CC/06	1
<b>TOTAL</b>		<b>5</b>

**Quadro XXV**

<b>SUPERINTENDÊNCIA DE SAÚDE E QUALIDADE DE VIDA</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Superintendente de Saúde e Qualidade de Vida	CC/02	1
Chefe da Seção de Gestão de Saúde e Qualidade de Vida no Trabalho	CC/06	1
Chefe da Seção Administrativa	CC/06	1
<b>TOTAL</b>		<b>3</b>

**Quadro XXVI**

<b>SUPERINTENDÊNCIA DE ENGENHARIA E ARQUITETURA</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Superintendente de Engenharia e Arquitetura	CC/02	1
Chefe da Seção de Fiscalização de Obras e Projetos	CC/06	1
Chefe da Seção de Manutenção Predial	CC/06	1
<b>TOTAL</b>		<b>3</b>

Quadro XXVII

<b>SECRETARIA DE ADMINISTRAÇÃO E GESTÃO DE PESSOAS (SEAD)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário de Administração	CC/01	1
Assessor Administrativo	CC/03	3
Assessor Judiciário	CC/03	2
Coordenador de Patrimônio, Materiais e Documentação	CC/04	1
Coordenador de Serviços Gerais	CC/04	1
Coordenador de Gestão de Pessoal	CC/04	1
Coordenador de Pagamento de Pessoal	CC/04	1
Coordenador de Transportes	CC/04	1
Chefe de Seção de Arquivo Administrativo	CC/06	1
Chefe de Seção da Logística	CC/06	1
Chefe de Seção de Protocolo Geral	CC/06	1
Chefe de Seção de Serviços Gráficos	CC/06	1
Chefe de Seção de Patrimônio	CC/06	1
Chefe de Seção de Registro e Cadastro Funcional	CC/06	1
Chefe de Seção de Apoio aos Magistrados	CC/06	1
Chefe da Seção de Compras	CC/06	1
Chefe de Seção de Acompanhamento e Avaliação de Desempenho	CC/06	1
Chefe de Seção de Análise e Cálculos	CC/06	1
<b>TOTAL</b>		<b>21</b>

**Quadro XXVIII**

<b>SECRETARIA DE ORÇAMENTO E FINANÇAS (SOF)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário de Orçamento e Finanças	CC/01	1
Assessor Administrativo	CC/03	3
Coordenador de Execução Orçamentária	CC/04	1
Coordenador de Tesouraria	CC/04	1
Coordenador de Contabilidade e Controle	CC/04	1
Coordenador de Execução Financeira	CC/04	1
Chefe de Seção de Execução Orçamentária TJ e FERMOJUPI	CC/06	1
Chefe de Seção de Execução Orçamentária da EJUD	CC/06	1
<b>TOTAL</b>		<b>10</b>

**Quadro XXIX**

<b>GABINETE DO CORREGEDOR</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor Judiciário	CC/03	1
Consultor Jurídico	CC/02	4
Assessor de Magistrado	CC/03	4
Oficial de Gabinete de Magistrado	CC/06	1
<b>TOTAL</b>		<b>10</b>

**Quadro XXX**

<b>CPPAD – 1º GRAU (Corregedoria)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Presidente de Comissão de Processo Administrativo	FC/02	1
Membros da CPPAD	FC/03	2
<b>TOTAL</b>		<b>3</b>

**Quadro XXXI**

**(Quadro alterado pela Lei Complementar nº 237/2018, de 17 de julho de 2018)**

<b>Gabinete dos Juizes Auxiliares da Corregedoria</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor Judiciário	CC/03	<del>2</del> 1
Coordenador Disciplinar	CC/04	1
Coordenador Judicial	CC/04	1
Chefe de Seção de Metas e Indicadores	CC/06	1
Chefe de Seção de Correição	CC/06	1
<b>TOTAL</b>		<del>6</del> <b>5</b>

**Quadro XXXII**

**(Quadro alterado pela Lei Complementar nº 237/2018, de 17 de julho de 2018)**

<b>Gabinete do Vice-Corregedor</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Consultor Jurídico	CC/02	4
Assessor Judiciário	CC/03	<del>1</del> 2
Assessor de Magistrado	CC/03	4
<b>Coordenador Administrativo (acrescido)</b>	<b>CC/04 (acrescido)</b>	<b>1</b>
<b>Chefe de seção de autuação e organização processual (acrescido)</b>	<b>CC/06 (acrescido)</b>	<b>1</b>
Oficial de Gabinete	CC/06	1
<b>TOTAL</b>		<del>10</del> <b>13</b>

Quadro XXXIII

(Quadro alterado pela Lei Complementar nº 237/2018, de 17 de julho de 2018)

SECRETARIA DA CORREGEDORIA		
Cargo/Função	Símbolo	Quantidade
Secretário da Corregedoria	CC/01	1
Assessor Administrativo	CC/03	1
Assistente de Imprensa e Divulgação	CC/04	1
<del>Coordenador Administrativo</del>	<del>CC/04</del>	<del>4</del>
Coordenador de Cadastro e Tramitação Processual	CC/04	1
Coordenador de Planejamento e Modernização	CC/04	1
Coordenador de Orçamento e Finanças da CGJ	CC/04	1
Chefe de Seção de Arquivo e Depósito Judicial	CC/06	1
Chefe de Seção de Transportes	CC/06	1
Chefe de Seção de Expedientes	CC/06	1
<del>Chefe de Seção de Autuação e Organização Processual</del>	<del>CC/06</del>	<del>4</del>
Chefe de Seção de Cadastro de Serviços Judiciais e Cartorários	CC/06	1
Chefe de Seção de Contabilidade e Controle da CGJ	CC/06	1
Oficial da CEJAI	FC/03	1
<b>TOTAL</b>		<del>14</del> 12

Quadro XXXIV

Escola Judiciária do Estado do Piauí (EJUD-PI)		
Cargo/Função	Símbolo	Quantidade
Superintendente Administrativo da EJUD	CC/02	1
Coordenador Pedagógico	CC/04	1
Chefe da Seção da Biblioteca	CC/06	1
Chefe da Seção de Ensino à Distância	CC/06	1
Chefe da Seção de Formação e Aperfeiçoamento	CC/06	1
Chefe da Seção de Registro e Controle Acadêmico	CC/06	1
<b>TOTAL</b>		<b>6</b>

**Quadro XXXV**

<b>Secretaria de Tecnologia da Informação e Comunicação (STIC)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário de Tecnologia da Informação e Comunicação	CC/01	1
Assessor Administrativo	CC/03	2
Coordenador de Governança de TI	CC/04	1
Coordenador de Software	CC/04	1
Coordenador de Infraestrutura de TIC	CC/04	1
Chefe de Seção de Segurança da Informação	CC/06	1
Chefe de Seção de Suporte e Manutenção	CC/06	1
Chefe de Seção de Planejamento Estratégico e Gestão de Projetos	CC/06	1
Chefe de Seção de Aquisições e Contratações de Soluções de TIC	CC/06	1
Chefe de Seção de Sistemas Judiciais	CC/06	1
Chefe de Seção de Sistemas Administrativos	CC/06	1
Chefe de Seção de Sistemas Extrajudiciais	CC/06	1
Chefe de Seção de Banco de Dados	CC/06	1
Chefe de Seção de Redes de Comunicação	CC/06	1
Chefe de Seção de Apoio Tecnológico	CC/06	1
Oficial de TI	FC/03	5
<b>TOTAL</b>		<b>21</b>



**Quadro XXXVI****(Quadro alterado pela Lei Complementar nº 232/2018, de 23 de abril de 2018)**

<b>GABINETES DOS DESEMBARGADORES</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor de Magistrado	CC/03	<del>60</del> <b>64</b>
Oficial de Gabinete de Magistrado	CC/06	<del>15</del> <b>16</b>
<b>TOTAL</b>		<del>75</del> <b>80</b>

**Quadro XXXVII****(Quadro alterado pela Lei Complementar nº 232/2018, de 23 de abril de 2018)**

<b>SECRETARIA JUDICIÁRIA (SEJU)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário Judiciário	CC/01	1
Consultor Jurídico	CC/02	<del>64</del> <b>65</b>
Assessor Judiciário	CC/03	1
Coordenador Judiciário do Pleno	CC/04	1
Coordenador Judiciário Cível	CC/04	1
Coordenador Judiciário Criminal	CC/04	1
Secretário de Sessões do Pleno	CC/05	1
Secretário de Sessões de Câmara Cível e de Direito Público	CC/05	4
Secretário de Sessões de Câmara Criminal e de Direito Público	CC/05	2
Distribuidor do 2º grau	FC/01	1
<b>TOTAL</b>		<b>74</b>

Quadro XXXVIII

<b>COORDENADORIA DE PRECATÓRIOS</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Coordenador de Precatórios	CC/04	1
Chefe de Seção de Controle Processual	CC/06	1
Chefe de Seção de Controle de Contas	CC/06	1
<b>TOTAL</b>		<b>3</b>

Quadro XXXIX

<b>JUSTIÇA ITINERANTE (Vice-Corregedoria)</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Superintendência da Justiça Itinerante	CC/02	1
Diretor de Secretaria	CC/04	2
Assessor Judiciário	CC/03	1
<b>Total</b>		<b>4</b>

Quadro XL

<b>CEJUSC 2º Grau</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Secretário do Centro Judiciário	FC/02	1
<b>TOTAL</b>		<b>1</b>

Quadro XLI

TURMAS RECURSAIS		
Cargo/Função	Símbolo	Quantidade
Assessor de Magistrado	CC/03	12
Diretor de Secretaria das Turmas Recursais	CC/04	1
<b>TOTAL</b>		<b>13</b>

Quadro XLII

DIRETORIA DO FÓRUM CENTRAL DE TERESINA		
Cargo/Função	Símbolo	Quantidade
Secretário Assistente de Diretoria de Fórum	FC/03	1
Chefe da Central de Mandados	CC/06	1
Chefe de Seção de Contadoria Judicial	CC/06	1
Chefe da Seção de Apoio Psicossocial	CC/06	1
Chefe de Seção de Atendimento e Certidões	CC/06	1
Distribuidor 1º grau	FC/01	1
<b>TOTAL</b>		<b>6</b>

Quadro XLIII

CENTRAL DE INQUÉRITOS		
Cargo/Função	Símbolo	Quantidade
Assessor de Magistrado	CC/03	1
Oficial da Audiência de Custódia	CC/06	1
Secretário da Central de Inquéritos	FC/02	1
<b>TOTAL</b>		<b>3</b>

Quadro XLIV

CEJUSCs Teresina (Corregedoria)		
Cargo/Função	Símbolo	Quantidade
Secretário do Centro Judiciário	FC/02	3
<b>TOTAL</b>		<b>3</b>

Quadro XLV

CEJUSCs Interior (Corregedoria)		
Cargo/Função	Símbolo	Quantidade
Secretário do Centro Judiciário – Parnaíba	FC/02	1
Secretário do Centro Judiciário – Picos	FC/02	1
Secretário do Centro Judiciário – Piriipiri	FC/02	1
Secretário do Centro Judiciário – Floriano	FC/02	1
Secretário do Centro Judiciário – Oeiras	FC/02	1
Secretário do Centro Judiciário – Corrente	FC/02	1
<b>TOTAL</b>		<b>6</b>

Quadro XLVI

DIRETORIA DOS FÓRUNS DE CAMPO MAIOR / FLORIANO / PARNAÍBA / PICOS / PIRIPIRI		
Cargo/Função	Símbolo	Quantidade
Chefe de Seção de Protocolo e Distribuição	CC/06	5
Chefe de Central de Mandados	CC/06	5
Secretário Assistente de Diretoria de Fórum	FC/03	5
<b>TOTAL</b>		<b>15</b>

**Quadro XLVII**  
**(Quadro alterado pela Lei Complementar nº 235/2018, de 12 de junho de 2018)**

<b>VARAS DE 1ª INSTÂNCIA</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor de Magistrado	CC/03	<del>183</del> <b>196</b>
Oficial de Gabinete de Magistrado	CC/06	152
Oficial da Corregedoria de Presídios – VEP Teresina / Floriano / Parnaíba / Picos / Oeiras / São Raimundo Nonato / Bom Jesus / Esperantina / <b>Campo Maior (AC)</b>	CC/06	<del>8</del> <b>9</b>
Secretário Assistente de Diretoria de Fórum	FC/03	46
Secretário de Vara	FC/02	152
<b>TOTAL</b>		<del>541</del> <b>555</b>

**Quadro XLVIII**

<b>JUIZADOS ESPECIAIS</b>		
<b>Cargo/Função</b>	<b>Símbolo</b>	<b>Quantidade</b>
Assessor de Magistrado	CC/03	5
Oficial de Gabinete de Magistrado	CC/06	32
Diretor de Secretaria	CC/04	68
Secretário Assistente de Diretoria de Fórum	FC/03	32
<b>TOTAL</b>		<b>137</b>

**Anexo IX**  
**Atribuições dos Cargos em Comissão e Funções Comissionadas**

<b>CARGO</b>	<b>SÍMBOLO</b>	<b>ATRIBUIÇÕES</b>
SECRETÁRIOS	CC/01	<p>a) participar da formulação de políticas e estratégias que visam ao aperfeiçoamento do Poder Judiciário, colaborando com seus conhecimentos para a implantação destas;</p> <p>b) avaliar o desempenho das atividades da unidade, com vistas à diminuição dos custos operacionais e ao aumento da produtividade;</p> <p>c) planejar e organizar as atividades da unidade como um todo, em conjunto com os seus colaboradores, estabelecendo metas com vistas à consecução dos objetivos;</p> <p>d) promover e facilitar a sinergia entre as unidades administrativas do TJPI, por meio do gerenciamento das informações;</p> <p>e) controlar e orientar as ações das unidades administrativas sob sua direção, determinando objetivos em consonância com a política e as diretrizes do Poder Judiciário, para possibilitar o desempenho correto das funções pertinentes à unidade que dirige;</p> <p>f) validar as normas de serviço e os procedimentos de ação, examinando e determinando as rotinas de</p>

		<p>trabalho e as formas de execução, para obter melhor produtividade dos recursos disponíveis;</p> <p>g) fazer cumprir decisões tomadas em assuntos de sua competência legal ou regimental, baixando instruções de serviço, expedindo ordens e controlando o cumprimento das mesmas, para possibilitar a plena realização dos objetivos previstos;</p> <p>h) propor às autoridades superiores soluções para assuntos de sua área de competência, elaborando pareceres, formulando consultas e apresentando soluções, a fim de contribuir para a resolução de questões dependentes de deliberação superior;</p> <p>i) examinar e encaminhar relatórios de exposição dos andamentos dos trabalhos e apresentação de sugestões, se for o caso, para informar à administração superior sobre os assuntos que dizem respeito às unidades sob sua responsabilidade;</p> <p>j) desempenhar funções delegadas por ordem superior, como articular-se com órgãos da administração pública ou com entidades sobre problemas de interesse comum;</p> <p>k) realizar outras atividades pertinentes à sua área de atuação.</p>
SUPERINTEDENTES	CC/02	<p>a) participar da formulação de políticas e estratégias que visam ao aperfeiçoamento contínuo do Poder Judiciário, colaborando com seus conhecimentos para a implementação destas;</p> <p>b) avaliar o desempenho das atividades da unidade com vistas à diminuição dos custos operacionais e aumento da produtividade;</p> <p>c) planejar e organizar as atividades da unidade como um todo, em conjunto com os seus colaboradores, estabelecendo metas com vistas à consecução de objetivos;</p> <p>d) participar da formulação das normas a serem seguidas, colaborando com informes, sugestões e experiências para a definição de objetivos e metas, além da articulação de sua área gerencial com as demais;</p> <p>e) estabelecer ou aprovar métodos e procedimentos, horários e percursos, baseando-se nos objetivos a serem alcançados e na disponibilidade de recursos materiais, financeiros e humanos, para definir prioridades, sistemas e rotinas relacionadas a esses serviços;</p> <p>f) organizar as atividades programadas, distribuindo-as pelos setores da unidade que supervisiona, através do estabelecimento de normas e processos a serem seguidos, para assegurar o fluxo normal dessas atividades, os resultados previstos e padrões regulamentares uniformes na área sob sua responsabilidade;</p> <p>g) representar o TJPI nas negociações com outros poderes públicos e/ou instituições, formulando propostas de acordos, contratos e outras formas de cooperação, dentro dos poderes que lhe foram conferidos, para possibilitar soluções que atendam às necessidades e objetivos deste Poder, no que se relaciona com os serviços prestados, para verificar possíveis falhas e determinar ou propor modificações necessárias;</p> <p>h) informar à administração superior sobre as atividades executadas e os resultados alcançados, elaborando relatórios ou memoriais, ou através de reuniões e outros meios, para possibilitar a avaliação das diretrizes adotadas e sua conjugação com a política geral do TJPI;</p> <p>i) distribuir o serviço, fornecendo informações e implantando rotinas de trabalho, para assegurar e orientar a sua execução;</p> <p>j) informar seu superior imediato sobre o processamento dos trabalhos e resultados, através de relatórios, reuniões ou outros meios, para possibilitar avaliação geral;</p> <p>k) cumprir e fazer cumprir os despachos exarados e execução das ordens judiciais emanadas dos senhores Desembargadores, zelando pelo cumprimento dos prazos legais e judiciais;</p> <p>l) acompanhar a elaboração e a emissão de certidões pertinentes solicitadas;</p> <p>m) orientar e controlar a elaboração da pauta de julgamento dos processos em sua área de atuação;</p> <p>n) realizar outras atividades pertinentes à sua área de atuação.</p>
CONSULTOR JURÍDICO	CC/02	<p>a) despachar, diariamente, com os integrantes dos órgãos fracionários ou de gestão superior do Poder Judiciário;</p> <p>b) elaborar minutas de atos judiciais e administrativos de ordem monocrática nos processos distribuídos para relatoria dos integrantes dos órgãos fracionários ou de gestão superior do Poder Judiciário, após receber a devida orientação deste;</p>

		<p>c) elaborar minutas de atos judiciais e administrativos relacionados à competência dos órgãos fracionários e plenário, tais como: votos, revisões, acórdãos, portarias, resoluções, provimentos e instruções normativas;</p> <p>d) exercer, durante as sessões colegiadas e plenária, atividades de apoio aos seus integrantes;</p> <p>e) executar outras atribuições pertinentes ao cargo, conforme determinação superior.</p>
ASSESSOR DE MAGISTRADO	CC/03	<p>a) pesquisar textos jurídicos em doutrinas e jurisprudências;</p> <p>b) analisar os fundamentos dos recursos ou das ações originárias, analisando seus conteúdos, com base em textos legais, de modo a oferecer subsídios para a elaboração do voto pelo Desembargador;</p> <p>c) emitir relatórios dos processos para submetê-los a julgamento;</p> <p>d) supervisionar as atividades do Gabinete pertinentes ao controle de processos, visando manter o órgão julgador informado sobre as fases dos feitos conclusos;</p> <p>e) supervisionar e conferir a digitação dos votos elaborados;</p> <p>f) elaborar minuta de despachos nos autos para dar andamento aos processos, com vistas ao controle eficiente e eficaz das informações registradas;</p> <p>g) realizar outras atividades pertinentes à sua área de atuação.</p>
ASSESSOR JUDICIÁRIO	CC/03	<p>a) assessorar o gestor nas ações que visam à regularidade procedimental dos feitos pertinentes ao setor;</p> <p>b) realizar a pesquisa e a seleção de textos jurídicos e comunicações de interesse, consultando livros, diários oficiais e outras fontes, para inteirar-se do processo e apurar informações pertinentes ao caso que está sendo considerado, para melhor subsidiar o gestor nas suas decisões;</p> <p>c) redigir ou elaborar documentos jurídicos, pronunciamentos, minutas e informações que compõem o processo, de natureza administrativa, fiscal, cível, comercial, trabalhista, penal ou outras, aplicando legislação, forma e terminologia adequadas ao assunto em questão, garantindo autenticidade dos mesmos;</p> <p>d) desenvolver análises e auxiliar o gestor na supervisão e conferência das ações exaradas, velando pelo cumprimento dos prazos estabelecidos em leis e regulamentos;</p> <p>e) analisar processos enviados à unidade, averiguando o conteúdo, partes interessadas, propósitos e demais condicionantes para proceder os devidos encaminhamentos;</p> <p>f) zelar pelo cumprimento dos regulamentos e ordens de serviço, aplicando as medidas e providências cabíveis, para assegurar a consecução dos objetivos propostos;</p> <p>g) realizar outras atividades pertinentes à sua área de atuação.</p>
ASSESSOR ADMINISTRATIVO	CC/03	<p>a) proceder estudos específicos, coletando e analisando dados e examinando atividades especializadas em sua área de atuação, para colaborar nos trabalhos técnicos relativos e projetos básicos de ação, quando necessário, atualizando-se em questões relativas à aplicação de leis e regulamentos;</p> <p>b) participar da elaboração da Política Administrativa do TJPI, colaborando com informações, sugestões e experiências, a fim de contribuir para a definição de objetivos e para a articulação da área administrativa com as demais;</p> <p>c) elaborar relatórios periódicos, fazendo as exposições, para informar sobre o andamento dos trabalhos;</p> <p>d) acompanhar o desenvolvimento dos programas administrativos, orientando os gestores na solução de dúvidas e problemas, sugerindo estudos para possibilitar melhor desempenho dos trabalhos e avaliação dos efeitos;</p> <p>e) elaborar documentos diversos, consultando fontes de informação disponíveis, para possibilitar a apresentação dos dados solicitados;</p> <p>f) coletar informações e preparar projetos com propostas de criação de novos serviços ou modificações dos já existentes;</p> <p>g) redigir correspondências e documentos de rotina, observando os padrões estabelecidos de forma e estilo, para assegurar o funcionamento do sistema de comunicação interna e externa;</p> <p>h) realizar outras atividades pertinentes à sua área de atuação.</p>
ASSESSOR DE COMUNICAÇÃO SOCIAL	CC/03	<p>a) realizar a alimentação diária do portal de notícias (<i>homepage</i>) do Poder Judiciário;</p> <p>b) executar serviços de administração de informações jornalísticas (coleta, avaliação, sistematização</p>

		<p>e divulgação), bem como o seu fluxo para os veículos de comunicação;</p> <p>c) organizar e manter atualizada uma relação dos veículos de comunicação, acompanhados dos respectivos endereços, telefones, fax, e-mails, nomes de diretores e editores, etc</p> <p>d) elaborar produtos jornalísticos diversos, tais como fotografias, vídeos, programas de rádio ou de televisão, para fins de divulgação das ações do Poder Judiciário;</p> <p>e) participar da definição das estratégias de comunicação;</p> <p>f) realizar outras atividades pertinentes à sua área de atuação.</p>
AUXILIAR DE APOIO JUDICIÁRIO	CC/04	<p>a) organizar autos, papéis e sistemas;</p> <p>b) proceder a digitalização e alimentação de sistemas judiciais e administrativos;</p> <p>c) promover a logística de mutirões, esforços concentrados e semanas de conciliação;</p> <p>d) auxiliar nas jornadas de enfrentamento de demandas reprimidas;</p> <p>e) realizar outras atividades pertinentes à sua área de atuação.</p>
DIRETOR DE SECRETARIA	CC/04	<p>a) analisar o funcionamento das diversas rotinas, observando o desenvolvimento e efetuando estudos e ponderações a respeito, para propor medidas de simplificação e melhoria dos trabalhos;</p> <p>b) distribuir o serviço, fornecendo informações e implantando as rotinas de trabalho para assegurar e orientar a sua execução;</p> <p>c) informar papéis e processos, instruindo sobre o andamento dos mesmos para dar encaminhamento aos assuntos tratados;</p> <p>d) fazer cumprir as normas e ordens de serviço, organizando, distribuindo e orientando os trabalhos a serem executados, para assegurar a produtividade da unidade a que está vinculado;</p> <p>e) relatar o andamento dos trabalhos, apresentando periodicamente relatórios e justificativas, para informar sobre a execução das atividades que lhe competem;</p> <p>f) lavrar os termos de abertura e encerramento dos livros referentes as atividades cartorárias, empregando redação específica para liberá-lo à escrituração das atividades processadas;</p> <p>g) atender advogados e partes interessadas no processo, prestando informações verbais do estado e andamento dos feitos, salvo em assunto tratado em segredo de justiça, para cientificá-los da situação do mesmo;</p> <p>h) controlar todas as fases dos processos judiciais, mantendo atualizadas as informações pertinentes ao mesmo;</p> <p>i) monitorar os processos através de lançamentos e baixa dos mesmos aos advogados, partes, peritos, dando-lhe vistas dos autos em livro próprio;</p> <p>j) controlar despachos e sentenças, conferindo e remetendo para publicação no Diário da Justiça, a fim de intimação das partes, através de seus advogados.</p> <p>k) realizar outras atividades pertinentes à sua área de atuação.</p>
COORDENADOR	CC/04	<p>a) coordenar no seu âmbito de atuação a implementação dos programas e projetos específicos de ação, para atingir os objetivos da maneira como foram propostos, no tempo previsto e com o uso legítimo dos meios disponíveis;</p> <p>b) controlar o cumprimento das funções dos diferentes serviços da instituição, acompanhando o desenvolvimento das atividades, para assegurar a realização dos programas em execução;</p> <p>c) responder pela definição de soluções nas atividades que envolvam sua área de atuação, analisando com seus colaboradores e demais unidades envolvidas, todos os aspectos da operação, para garantir segurança e confiabilidade na tomada de decisão;</p> <p>d) acompanhar o desenvolvimento das atividades realizadas, participando diretamente na estruturação das mesmas, para garantir a máxima fidelidade às políticas internas da Instituição;</p> <p>e) elaborar relatórios periódicos acerca dos resultados obtidos com os programas, projetos específicos de ação e atividades executadas para informar à administração superior.</p> <p>f) acompanhar os lançamentos nos sistemas eletrônicos em uso para permitir o controle da documentação e consulta;</p> <p>g) zelar pelo cumprimento dos regulamentos e ordens de serviço, aplicando as medidas e providências</p>



		<p>cabíveis para assegurar a consecução dos objetivos propostos;</p> <p>h) elaborar quadros, gráficos, planilhas e relatórios, das atividades executadas e controladas pela unidade, fazendo as exposições pertinentes, para avaliação periódica das atividades;</p> <p>i) realizar outras atividades pertinentes à sua área de atuação.</p>
ASSISTENTE DE IMPRENSA E DIVULGAÇÃO	CC/04	<p>a) possibilitar a adequada divulgação externa e interna de atos, serviços e eventos da Justiça, bem como dos pronunciamentos de magistrados ou dirigentes do Poder Judiciário;</p> <p>b) assessorar juízes e dirigentes do Poder Judiciário nos contatos com a imprensa;</p> <p>c) prestar informações e colaborar em entrevistas e reportagens sobre o Poder Judiciário e seus membros;</p> <p>d) elaborar, revisar e controlar matérias jornalísticas para publicação;</p> <p>e) planejar a inserção midiática de matérias de interesse do Tribunal de Justiça;</p> <p>f) auxiliar na elaboração de materiais para as campanhas de divulgação;</p> <p>g) realizar outras atividades pertinentes à sua área de atuação.</p>
ASSISTENTE DE CERIMONIAL	CC/04	<p>a) organizar e coordenar as solenidades e as visitas protocolares, observando as normas do cerimonial por parte do Tribunal de Justiça;</p> <p>b) organizar festividades internas do Tribunal de Justiça, visando o conagraçamento dos servidores, bem como sua participação em eventos externos;</p> <p>c) assessorar a Presidência em questões de cerimonial, bem como ao Presidente ou seu representante oficial, quando do seu comparecimento em solenidades, comemorações e recepções, e aos demais membros do Tribunal, quando solicitados;</p> <p>d) preparar e expedir convites para cerimoniais e festividades promovidas pelo Tribunal de Justiça;</p> <p>e) organizar e manter atualizado o cadastro de autoridades constituídas dos três poderes, no âmbito federal, estadual e municipal, bem como outras informações pertinentes;</p> <p>f) assessorar a Presidência na recepção a autoridades nacionais ou estrangeiras em visita de caráter oficial ao Tribunal de Justiça;</p> <p>g) colaborar com os demais assessores de sua unidade na elaboração e execução dos planos de trabalho e cronogramas de realização de atividades de forma a contribuir para o bom desempenho da unidade organizacional;</p> <p>h) realizar outras atividades pertinentes à sua área de atuação.</p>
ASSISTENTE DE SEGURANÇA	CC/04	<p>a) executar as ações de segurança pessoal elaboradas pela Assessoria Militar no tocante à preservação da integridade física dos magistrados com que estiver trabalhando;</p> <p>b) realizar outras atividades pertinentes à sua área de atuação.</p>
SECRETÁRIO DE SESSÃO	CC/05	<p>a) lavrar os termos e atas, fazendo constar o que na sessão houver ocorrido, lendo e expondo para aprovação dos presentes, ao final coletando a assinatura do desembargador que presidiu o ato e dos demais presentes;</p> <p>b) secretariar as sessões de julgamento conforme a respectiva lotação;</p> <p>c) realizar a leitura integral dos autos ou de parte deles quando o Desembargador Relator determinar que seja ela efetuada pelo secretário;</p> <p>d) consignar de modo sucinto, o que se passar nas sessões;</p> <p>e) elaborar certidões e lançá-las no sistema;</p> <p>f) elaborar moções de pesar e felicitações apresentadas em sessão;</p> <p>g) proceder às degravações;</p> <p>h) auxiliar na direção e coordenação da respectiva Coordenadoria Judiciária em que for lotado;</p> <p>i) proceder com a convocação de Desembargadores para compor o quorum de julgamento nas faltas, ausências e impedimentos;</p> <p>j) realizar outras atividades pertinentes à sua área de atuação.</p>

AJUDANTE DE ORDEM	CC/05	<ul style="list-style-type: none"> <li>a) executar os serviços de ajudância de ordem do Presidente do Tribunal de Justiça;</li> <li>b) elaborar os planos e roteiros de segurança dos magistrados e autoridades, nos deslocamentos;</li> <li>c) supervisionar o serviço de transporte do Presidente, Vice-Presidente e Corregedor-Geral da Justiça, do Tribunal de Justiça;</li> <li>d) atender às ocorrências solicitadas pela autoridade competente;</li> <li>e) executar tarefas relacionadas à execução de medidas de prevenção contra atos de violência e outras infrações, à ordem e à segurança;</li> <li>f) auxiliar o Presidente do Tribunal de Justiça nos assuntos relacionados com a segurança pública;</li> <li>g) realizar outras atividades pertinentes à sua área de atuação.</li> </ul>
SECRETÁRIO EXECUTIVO	CC/05	<ul style="list-style-type: none"> <li>a) organizar os serviços de pessoal sob sua responsabilidade, distribuindo tarefas, fixando horário e substituições para possibilitar o desenvolvimento harmônico dos trabalhos;</li> <li>b) fiscalizar o processamento dos serviços, inspecionando-os, periodicamente, para evitar transtornos;</li> <li>c) gerenciar as atividades de manutenção e conservação das instalações da unidade, visando manter as condições de funcionamento e segurança;</li> <li>d) zelar pelo cumprimento do regulamento interno do edifício, evitando o uso indevido das instalações e levando à administração os problemas surgidos, para possibilitar a manutenção da ordem e promover o bem-estar de seus ocupantes;</li> <li>e) redigir correspondências e documentos de rotina, observando os padrões estabelecidos de forma e estilo para assegurar o funcionamento do sistema de comunicação interna e externa;</li> <li>f) fazer chamadas telefônicas, requisições de material, registro e expedição de documentos e outras tarefas correlatas, seguindo os processos de rotina, para cumprir e agilizar os serviços no setor em colaboração com a chefia;</li> <li>g) receber visitantes que procuram a administração da unidade, prestando-lhes as informações necessárias para atender às solicitações dos mesmos;</li> <li>h) elaborar relatórios periódicos, fazendo exposições pertinentes, para informar seu superior imediato sobre o andamento dos trabalhos;</li> <li>i) realizar outras atividades pertinentes à sua área de atuação.</li> </ul>
CHEFE DE SEÇÃO	CC/06	<ul style="list-style-type: none"> <li>a) analisar o funcionamento das diversas rotinas, observando o desenvolvimento e efetuando estudos e ponderações a respeito, para propor medidas de simplificação e melhoria dos trabalhos;</li> <li>b) distribuir o serviço, fornecendo informações e implantando as rotinas de trabalho para assegurar e orientar a sua execução;</li> <li>c) informar papéis e processos, instruindo sobre o andamento dos mesmos para dar encaminhamento aos assuntos tratados;</li> <li>d) cumprir as normas e ordens de serviço, organizando, distribuindo e orientando os trabalhos a serem executados, para assegurar a produtividade das unidades a que está vinculado;</li> <li>e) relatar o andamento dos trabalhos, apresentando periodicamente relatórios e justificativas, para informar sobre a execução das atividades que lhe competem;</li> <li>f) zelar pelo cumprimento dos regulamentos e ordens de serviço, aplicando as medidas e providências cabíveis para assegurar a consecução dos objetivos propostos;</li> <li>g) avaliar a produção tanto no aspecto qualitativo quanto no quantitativo, considerando a eficiência de cada servidor e os recursos materiais disponíveis e determinar novas medidas, caso necessário;</li> <li>h) zelar pelo material de serviço, solicitando as providências necessárias para a sua conservação ou substituição;</li> <li>i) treinar o pessoal por quem responde, orientando-os e fazendo demonstrações das operações e tarefas a serem executadas, para obter o rendimento desejado na execução dos trabalhos;</li> <li>j) realizar outras atividades pertinentes à sua área de atuação.</li> </ul>
OFICIAL DE GABINETE DE MAGISTRADO	CC/06	<ul style="list-style-type: none"> <li>a) executar as atividades de apoio administrativo e processual junto ao gabinete do Magistrado, dando suporte no desenvolvimento de tarefas inerentes a este, tais como receber e devolver processos, exercendo</li> </ul>

		<p>controle de chegada e saída dos mesmos;</p> <p>b) responsabilizar-se pelo recebimento da correspondência dirigida ao Magistrado e, sob ordem do mesmo, dar o encaminhamento necessário;</p> <p>c) organizar os processos no gabinete, de forma a facilitar a atuação do juiz, podendo, inclusive, elaborar minuta de despachos e decisões, mediante prévia orientação do Magistrado;</p> <p>d) responsabilizar-se, sob a orientação do magistrado, pela prestação de informações estatísticas do gabinete;</p> <p>e) realizar outras atividades pertinentes à sua área de atuação.</p>
OFICIAL DA AUDIÊNCIA DE CUSTÓDIA	CC/06	<p>a) lavrar os termos e atas, fazendo constar o que na audiência de custódia houver ocorrido, lendo e expondo para aprovação dos presentes, ao final coletando a assinatura do magistrado que presidiu o ato e dos demais presentes;</p> <p>b) secretariar as audiências de custódia conforme o procedimento previsto;</p> <p>c) consignar de modo sucinto, o que se passar nas audiências de custódia;</p> <p>d) elaborar certidões e lançá-las no sistema;</p> <p>e) auxiliar na direção e coordenação da respectiva unidade;</p> <p>f) proceder às degravações;</p> <p>g) realizar outras atividades pertinentes à sua área de atuação.</p>
OFICIAL DA CORREGEDORIA DE PRESÍDIOS	CC/06	<p>a) assessorar o gestor nas ações que visam à regularidade procedimental dos feitos pertinentes ao setor;</p> <p>b) assessorar o gestor nas rotinas desenvolvidas no setor, observando o desenvolvimento, efetuando ponderações e atuando como facilitador na implantação de medidas de simplificação e melhoria;</p> <p>c) desenvolver estudos prospectivos de interesse do setor, comunicando-se com as fontes de informações e efetuando as anotações necessárias, para possibilitar a preparação de estudos e emissão de relatórios por parte da chefia;</p> <p>d) assistir o superior imediato com a preparação de subsídios para o processo decisório que requerem estudos prévios;</p> <p>e) realizar outras atividades pertinentes à sua área de atuação.</p>
DISTRIBUIDOR JUDICIÁRIO DE 1º E 2º GRAU - TERESINA	FC/01	<p>a) executar as atividades de protocolo e distribuição em sistema eletrônico de processos e petições, para posterior envio às unidades destinatárias, mediante protocolo, procedendo, ainda, à remessa respectiva no sistema informatizado, observadas as competências legais e atos normativos do Poder Judiciário;</p> <p>b) classificar, revisar e distribuir as ações judiciais quanto aos tipos e competência, zelando pela correta atribuição, no sistema informatizado, da classe processual e assunto, conforme Tabela Processual Unificada do Conselho Nacional de Justiça;</p> <p>c) efetuar o capeamento e proceder à numeração dos processos de competência originária;</p> <p>d) analisar e promover as anotações dos impedimentos, prevenções e dependências;</p> <p>e) certificar o recolhimento de guias de custas processuais;</p> <p>f) prestar informação nos autos, sempre que solicitado, dirimindo dúvidas quanto à distribuição;</p> <p>g) estabelecer padrões e propor a normatização dos serviços de distribuição;</p> <p>h) manter o sistema de registro, protocolo, acompanhamento e tramitação de processos sempre</p>

		<p>atualizados;</p> <p>i) efetuar averbações e cancelamentos de sua competência;</p> <p>j) realizar outras atividades pertinentes à sua área de atuação.</p>
PRESIDENTE DE COMISSÃO DE LICITAÇÃO	FC/02	<p>a) coordenar os trabalhos da equipe de apoio e a condução do procedimento licitatório e credenciamento dos interessados;</p> <p>b) receber declaração dos licitantes do pleno atendimento aos requisitos de habilitação, bem como os envelopes contendo as propostas e os documentos de habilitação;</p> <p>c) analisar e desclassificar as propostas que não atendem às especificações do objeto ou as condições e prazos de execução ou fornecimento fixado no edital;</p> <p>d) ordenar as propostas não desclassificadas e a seleção dos licitantes que participarão da fase de lances;</p> <p>e) verificar a decisão motivada a respeito da aceitabilidade do menor preço e analisar os documentos de habilitação do autor da oferta de menor preço;</p> <p>f) adjudicar o objeto ao licitante vencedor, se não tiver havido manifestação de recorrer por parte de algum licitante e elabora a ata da sessão pública;</p> <p>g) analisar os recursos eventualmente apresentados, reconsiderando o ato impugnado ou promovendo o encaminhamento do processo instruído com a sua manifestação à decisão da autoridade competente, propondo a homologação, anulação ou revogação do procedimento licitatório;</p> <p>h) realizar outras atividades pertinentes à sua área de atuação.</p>
PRESIDENTE DE COMISSÃO DE PROCESSO ADMINISTRATIVO	FC/02	<p>a) presidir os trabalhos das comissões de processos administrativos;</p> <p>b) determinar a realização de diligências;</p> <p>c) requisitar autos, papéis e informações;</p> <p>d) acessar os sistemas administrativos e judiciais para fins investigatórios;</p> <p>e) realizar outras atividades pertinentes à sua área de atuação.</p>
SECRETÁRIO DE VARA SECRETÁRIO DA CENTRAL DE INQUÉRITOS SECRETÁRIO DO NUPEMEC SECRETÁRIO DO CEJUSC	FC/02	<p>a) analisar o funcionamento das diversas rotinas, observando o desenvolvimento e efetuando estudos e ponderações a respeito, para propor medidas de simplificação e melhoria dos trabalhos;</p> <p>b) distribuir o serviço, fornecendo informações e implantando as rotinas de trabalho para assegurar e orientar a sua execução;</p> <p>informar papéis e processos, instruindo sobre o andamento dos mesmos para dar encaminhamento aos assuntos tratados;</p> <p>c) fazer cumprir as normas e ordens de serviço, organizando, distribuindo e orientando os trabalhos a serem executados, para assegurar a produtividade da unidade a que está vinculado;</p> <p>d) relatar o andamento dos trabalhos, apresentando periodicamente relatórios e justificativas, para informar sobre a execução das atividades que lhe competem;</p> <p>e) lavrar os termos de abertura e encerramento dos livros referentes as atividades cartorárias, empregando redação específica para liberá-lo à escrituração das atividades processadas;</p> <p>f) atender advogados e partes interessadas no processo, prestando informações verbais do estado e andamento dos feitos, salvo em assunto tratado em segredo de justiça, para cientificá-los da situação do mesmo;</p> <p>g) controlar todas as fases dos processos judiciais, mantendo atualizadas as informações pertinentes ao mesmo;</p> <p>h) monitorar os processos através de lançamentos e baixa dos mesmos aos advogados, partes, peritos, dando-lhe vistas dos autos em livro próprio;</p> <p>i) controlar despachos e sentenças, conferindo e remetendo para publicação no Diário da Justiça, a fim de intimação das partes, através de seus advogados.</p> <p>j) realizar outras atividades pertinentes à sua área de atuação.</p>
		<p>a) receber os atos decisórios da autoridade instauradora, tomando conhecimento do teor da reclamação e</p>

MEMBROS CPPAD	FC/03	<p>ciência da sua designação, por escrito;</p> <p>b) providenciar o local dos trabalhos e a instalação da comissão;</p> <p>c) realizar reuniões e audiências, confeccionando as atas e termos;</p> <p>d) prestar atendimento aos interessados e advogados;</p> <p>e) determinar expedição de ofícios e convocações;</p> <p>f) realizar diligências externas;</p> <p>g) reportar-se diretamente a todos os setores internos do Tribunal de Justiça e a terceiros de fora da instituição, em diligências e comunicações necessárias ao esclarecimento dos fatos;</p> <p>h) elaborar despacho de indicição e o relatório;</p> <p>i) manter sigilo sobre informações, ressalvadas as decorrentes de exercício de direito, prerrogativa ou de interesse legítimo;</p> <p>j) zelar pela incomunicabilidade das testemunhas;</p> <p>k) realizar medidas direcionadas para esclarecimento da verdade e a segurança jurídica dos atos;</p> <p>l) realizar a apuração das provas, obedecendo a todos os preceitos normativos, preservando o contraditório;</p> <p>m) tomar medidas que preservem a independência e a imparcialidade e garantam o sigilo necessário à elucidação do fato ou exigido pelo interesse da administração;</p> <p>n) assegurar ao servidor o acompanhamento do processo, pessoalmente ou por intermédio de procurador, bem como a utilização dos meios e recursos admitidos em direito, para comprovar suas alegações;</p> <p>o) zelar pela correta formalização dos procedimentos;</p> <p>p) reportar suas solicitações à autoridade superior;</p> <p>q) realizar outras atividades pertinentes à sua área de atuação.</p>
OFICIAL DE TECNOLOGIA DA INFORMAÇÃO	FC/03	<p>a) treinar os usuários nos aplicativos disponíveis, dando suporte na solução de problemas;</p> <p>b) operacionalizar a manutenção e controle de projetos mantidos pela unidade;</p> <p>c) assessorar na coleta, tabulação e análise dos dados informatizados inerentes aos trabalhos da unidade;</p> <p>d) sugerir, implantar, orientar e acompanhar rotinas de planejamento de projetos;</p> <p>e) facilitar a comunicação entre a Secretaria de Tecnologia da Informação e Comunicação e a unidade onde desenvolver suas atividades;</p> <p>f) realizar outras atividades pertinentes à sua área de atuação.</p>
OFICIAL DA CEJAI	FC/03	<p>a) instruir e secretariar os processos de adoção internacional;</p> <p>b) elaborar os relatórios estatísticos dos proponentes de adoção internacional;</p> <p>c) compilar os relatórios das equipes multidisciplinares;</p> <p>d) realizar outras atividades pertinentes à sua área de atuação.</p>
PREGOEIRO	FC/03	<p>a) examinar as propostas apresentadas em procedimento licitatório, quanto ao objeto e valor, e decidir motivadamente a respeito da sua aceitabilidade;</p> <p>b) proceder a abertura do invólucro contendo os documentos de habilitação do licitante que apresentar a melhor proposta, para verificação do atendimento das condições fixadas no edital;</p> <p>c) examinar as ofertas subsequentes e a qualificação dos licitantes, na ordem de classificação, e assim sucessivamente, até a apuração de uma que atenda ao edital;</p> <p>d) negociar diretamente com o proponente para que seja obtido preço melhor;</p> <p>e) realizar outras atividades pertinentes à sua área de atuação.</p>
SECRETÁRIO ASSISTENTE DE DIRETORIA DE FÓRUM	FC/03	<p>a) assessorar o Diretor do Fórum em suas funções, redigir minutas de atos, atender os interessados e se corresponder com outros órgãos e autoridades.</p> <p>b) gerir os recursos adiantados a título de suprimento de fundos;</p> <p>c) fiscalizar os contratos de fornecimento de material e serviços à Comarca;</p> <p>d) realizar outras atividades pertinentes à sua área de atuação.</p>

**ANEXO X**

**Detalhamento dos cargos em comissão e das funções de confiança e requisitos de escolaridade**

**(Quadro alterado pela Lei Complementar nº 232/2018, de 23 de abril de 2018)**

**(Quadro alterado pela Lei Complementar nº 237/2018, de 17 de julho de 2018)**

CC/FC	CARGO /FUNÇÃO	QUANT. GERAL	UNIDADE	QUANT. UNID.	LOTAÇÃO	REQUISITO
CC/01	Secretário	09	SEGER	1	Secretaria Geral	Bacharelado em Direito ou Administração
			SEJU	1	Secretaria Judiciária	Bacharelado em Direito
			SAJ	1	Secretaria de Assuntos Jurídicos	Bacharelado em Direito
			SEPRE	1	Secretaria da Presidência	Qualquer curso de nível superior
			SEGES	1	Secretaria de Gestão Estratégica	Qualquer curso de nível superior
			STIC	1	Secretaria de Tecnologia da Informação e Comunicação	Bacharelado nas áreas de Tecnologia da Informação
			SEAD	1	Secretaria de Administração e Gestão de Pessoas	Bacharelado em Direito ou Administração
			SOF	1	Secretaria de Orçamento e Finanças	Bacharelado em Direito, Ciências Contábeis, Economia ou Administração
			SECOR	1	Secretaria da Corregedoria	Bacharelado em Direito ou Administração
CC/02	Consultor Jurídico	<del>77</del> 81	SECPRE	4	Secretaria da Presidência	Bacharelado em Direito e 02 (dois) anos de prática jurídica
			VICEPRE	4	Gabinete da Vice-Presidência	
			GABCOR	4	Gabinete do Corregedor	
			VICECOR	4	Gabinete do Vice-Corregedor	
			SEJU	<del>64</del> 65	Secretaria Judiciária	
	Superintendente	9	SJI	1	Justiça Itinerante	Bacharelado em Direito
			SCI	1	Auditoria e Controle Interno	Bacharelado em Direito, Ciências Contábeis, Economia, Ciências Atuariais, Engenharia Civil, Administração ou Curso de Nível Superior na área de Tecnologia da Informação
			FERM	1	FERMOJUPI	Bacharelado em Direito, Ciências Contábeis, Economia ou Administração
			SENA	1	Engenharia e Arquitetura	Bacharelado em Engenharia ou Arquitetura
			SLC	1	Licitações e Contratos	Bacharelado em Direito, Ciências Contábeis, Economia ou Administração
			SGC	1	Gestão de Contratos e Convênios	Bacharelado em Direito, Ciências Contábeis, Economia ou Administração
			SUGESQ	1	Gestão de Saúde e Qualidade de Vida	Bacharelado em Medicina, Psicologia, Enfermagem e Odontologia

			SUPEJUD	1	Escola Judiciária	Qualquer curso de nível superior
			SUSEG	1	Segurança	Oficial Superior PM
CC/03	Assessor de Magistrado	<del>278</del> 282	2º GRAU	<del>60</del> 64	Gabinete de Desembargador	Bacharelado em Direito
				4	Secretaria da Presidência	
				4	Vice-Presidência do Tribunal de Justiça	
				4	Gabinete do Corregedor	
				4	Gabinete do Vice-Corregedor	
				1	NUPEMEC	
			1º GRAU	12	Turmas Recursais	
				1	Central de Inquéritos	
				183	Varas	
				5	Juizados Especiais	
	Assessor Judiciário	19	VICE-PRES	2	Vice Presidência	Bacharelado em Direito
			GAJA-PRE	2	Gabinetes dos Juízes Auxiliares da Presidência	
			SEJU	1	Secretaria Judiciária	
			SAJ	1	Secretaria de Assuntos Jurídicos	
			SEGES	1	Secretaria de Gestão Estratégica	
			SEGER	3	Secretaria Geral	
			FERM	1	FERMOJUPI	
			SLC	1	Superintendência de Licitações e Contratos	
			SEAD	2	Secretaria de Administração e Gestão de Pessoas	
GABCOR			1	Gabinete da Corregedoria		
VICE-COR			<del>4</del> 2	Gabinete da Vice-Corregedoria		
GAJA-COR			<del>2</del> 1	Gabinetes dos Juízes Auxiliares da Corregedoria		
SJI			1	Superintendência da Justiça Itinerante		
CC/03	Assessor Administrativo	33	SLC	1	Superintendência de Licitações e Contratos	Qualquer curso de nível superior
			FERM	1	FERMOJUPI	
			VICE	1	Vice Presidência	

			SEGES	1	Secretaria de Gestão Estratégica	
			SEPRE	2	Secretaria da Presidência	
			SAJ	2	Secretaria de Assuntos Jurídicos	
			SCI	1	Superintendência de Controle Interno	
			SEGER	4	Secretaria Geral	
			SEAD	3	Secretaria de Administração	
			STIC	2	Secretaria de Tecnologia da Informação e Comunicação	
			SOF	3	Secretaria de Orçamento e Finanças	
			SECCOR	1	Secretaria da Corregedoria	
			NAUJ-2º grau	3	Núcleo de Apoio às Unidades Judiciárias do 2º grau	
			NAUJ-1º grau	3	Núcleo de Apoio às Unidades Judiciárias do 1º grau	
			CSI	1	Conselho de Segurança Institucional	
			OUV	1	Ouvidoria Judiciária	
			SGC	2	Superintendência de Gestão de Contratos	
			SUJECCS	1	Supervisão dos Juizados Especiais	
Assessor de Comunicação Social	1	ASCOM	1	Assessoria de Comunicação da Presidência	Bacharelado em Jornalismo ou Comunicação Social	
CC/04	Coordenador	39	EJUD	1	Coordenadoria Pedagógica	Qualquer curso de nível superior
			NUGEP	1	Coordenadoria do Núcleo de Gerenciamento de Precedentes	Bacharelado em Direito
			CPREC	1	Coordenadoria de Precatórios	Qualquer curso de nível superior
			GMF	1	Coordenadoria do Grupo de Monitoramento e Fiscalização do Sistema Carcerário	Bacharelado em Direito
			NUSA	1	Coordenadoria do Núcleo Socioambiental	Qualquer curso de nível superior
			NUMEJ	1	Coordenadoria do Núcleo de Memória Judicial	
			OUV	1	Coordenadoria da Ouvidoria	
			NUPEMEC	1	Coordenadoria de Políticas Judiciárias de Cidadania	Bacharelado em Direito
				1	Coordenadoria de Modalidades Autocompositivas	
CSI	1	Conselho de Segurança Institucional	Qualquer curso de nível superior			



			SEJU	1	Coordenadoria Judiciária Cível e Câmaras Reunidas	Bacharelado em Direito
				1	Coordenadoria Judiciária Criminal e Câmaras Reunidas	
				1	Coordenadoria Judiciária do Pleno	
			SAJ	1	Coordenadoria Administrativa	Qualquer curso de nível superior
				1	Coordenadoria Jurídica	Bacharelado em Direito
			SCI	1	Coordenadoria de Auditoria	Qualquer curso de nível superior
				1	Coordenadoria de Acompanhamento da Gestão	
			FERMOJ UPI	1	Coordenadoria de Fiscalizações	
				1	Coordenadoria de Controle de Receitas	
				1	Coordenadoria de Controle de Processos Fiscais	
			SEGES	1	Escritório de Projetos	
			STIC	1	Coordenadoria de Governança de TIC	Bacharelado nas áreas de T.I.
				1	Coordenadoria de Software	
				1	Coordenadoria de Infraestrutura de TIC	
			SEAD	1	Coordenadoria de Patrimônio, Materiais e Documentação	Qualquer curso de nível superior
				1	Coordenadoria de Serviços Gerais	
				1	Coordenadoria de Gestão de Pessoal	
				1	Coordenadoria de Pagamento de Pessoal	
				1	Coordenadoria de Transportes	
			SOF	1	Coordenadoria de Execução Orçamentária	
				1	Coordenadoria de Execução Financeira	
1	Coordenadoria de Contabilidade e Controle					
1	Coordenadoria de Tesouraria					
GAJA-COR	1	Coordenadoria Judicial e Extrajudicial				
	1	Coordenadoria Disciplinar				
VICE-COR	1	<b>Coordenadoria Administrativa</b>				
SECCOR	4	<del>Coordenadoria Administrativa</del>				
	1	Coordenadoria de Planejamento e Modernização				

				1	Coordenadoria de Tramitação Processual	
				1	Coordenadoria de Orçamento e Finanças da CGJ	
	Assistente de Segurança	25	SUSEG	25	Superintendência de Segurança	Nível Médio
	Assistente de Imprensa e Divulgação	2	GABCOR	1	Gabinete do Corregedor	Qualquer curso de nível superior
			ASCOM	1	Assessoria de Comunicação da Presidência	
	Assistente de Cerimonial	1	ASCOM	1	Assessoria de Comunicação da Presidência	
	Auxiliar de apoio judiciário	20	NAUJ	20	NAUJ – Vice-Presidência	Qualquer curso de nível superior
Diretor de Secretaria	71	1º GRAU	1	Turmas Recursais	Bacharelado em Direito	
			68	Juizados Especiais		
		SJI	2	Justiça Itinerante		
CC/05	Secretário de Sessão	7	SEJU	4	Sessões das Câmaras Cíveis, Câmaras Reunidas e de Direito Público	Bacharelado em Direito
				2	Sessões das Câmaras Criminais, Câmaras Reunidas e de Direito Público	
				1	Sessões do Pleno	
	Ajudante de Ordem	2	SUSEG	2	Superintendência de Segurança	Oficial PM
	Secretário Executivo	3	CEJIJ	1	Coord. Est. da Infância e Juventude	Qualquer curso de nível superior
			CEM	1	Coord. Est. da Mulher	Qualquer curso de nível superior
GMF			1	Grupo de Monitoramento e Fiscalização do Sistema Carcerário	Bacharelado em Direito	
CC/06	Chefe de Seção	62	EJUD	1	Seção de Registro e Controle Acadêmico	Qualquer curso de nível superior
				1	Seção da Biblioteca	Bacharelado em Biblioteconomia
				1	Seção de Formação e Aperfeiçoamento	Qualquer curso de nível superior
				1	Seção de Ensino à Distância	Qualquer curso de nível superior
			CPRE	1	Seção de Controle de Contas	Bacharelado em Ciências Contábeis
				1	Seção de Controle Processual	Bacharelado em Direito
			SEGES	1	Seção de Governança	Qualquer curso de nível superior
				1	Seção de Análise Estatística	Bacharelado em Estatística
			FERM	1	Seção de Controle Processual	Qualquer curso de nível superior
				1	Seção de Controle de Contas	Qualquer curso de nível superior
				1	Seção de Administração do Selo	Qualquer curso de nível superior
			SLC	1	Seção de Apoio	Qualquer curso de nível superior

			SGC	1	Seção de Acompanhamento e Controle de Contratos e Convênios	Qualquer curso de nível superior
				1	Seção de Fiscalização de Contratos e Convênios	Qualquer curso de nível superior
			SUGESQ	1	Seção de Gestão de Saúde e Qualidade de Vida no Trabalho	Qualquer curso de nível superior
				1	Seção Administrativa	Qualquer curso de nível superior
			SENA	1	Seção de Fiscalização de Obras e Projetos	Qualquer curso de nível superior
				1	Seção de Manutenção Predial	Qualquer curso de nível superior
			STIC	1	Seção de Planejamento Estratégico e Gestão de Projetos	Bacharelado nas áreas de Tecnologia da Informação
				1	Seção de Aquisições e Contratações de Soluções de TIC	
				1	Seção de Sistemas Judiciais	
				1	Seção de Sistemas Administrativos	
				1	Seção de Sistema Extrajudiciais	
				1	Seção de Banco de Dados	
				1	Seção de Redes e Comunicação	
				1	Seção de Segurança de Informação	
				1	Seção de Suporte e Manutenção	
			SEAD	1	Seção de Patrimônio	Qualquer curso de nível superior
				1	Seção de Arquivos Administrativos	
				1	Seção de Logística de Materiais	
				1	Seção de Compras	
				1	Seção de Protocolo Geral	
				1	Seção de Serviços Gráficos	
				1	Seção de Registro e Cadastro Funcional	
				1	Seção de Acompanhamento e Avaliação de Desempenho	
1	Seção de Apoio aos Magistrados					
SOF	1	Seção de Análise e Cálculos				
	1	Seção de Execução Orçamentária – TJ e FERMOJUPI				

				1	Seção de Execução Orçamentária – CGJ e EJUD	
			<b>VICE-COR</b>	1	<b>Seção de Autuação e Organização Processual</b>	
			<b>SECCOR</b>	<del>4</del>	<del>Seção de Autuação e Organização Processual</del>	
				1	Seção de Cadastro de Serviços Judiciais e Cartorários	
				1	Seção do Arquivo e Depósito Judicial	
				1	Seção de Transportes	
				1	Seção de Expedientes	
				1	Seção de Contabilidade e Controle da CGJ	
				<b>GAJA-CGJ</b>	1	Seção de Metas e Indicadores
			1		Seção de Correição	Bacharelado em Direito
			<b>DIR FORUM CENTRAL DE TERESINA</b>	1	Central de Mandados - Teresina	Qualquer curso de nível superior
				1	Seção de Contadoria Judicial - Teresina	Bacharelado em Ciências Contábeis
				1	Seção de Apoio Psicossocial	Bacharelado em Psicologia ou Serviço Social
				1	Seção de Atendimento e Certidões - Teresina	Qualquer curso de nível superior
			<b>CAMPO MAIOR</b>	1	Seção de Protocolo e Distribuição - Campo Maior	Qualquer curso de nível superior
				1	Central de Mandados- Campo Maior	Qualquer curso de nível superior
			<b>FLORIANO</b>	1	Seção de Protocolo e Distribuição - Floriano	Qualquer curso de nível superior
				1	Central de Mandados - Floriano	Qualquer curso de nível superior
			<b>PARNAÍBA</b>	1	Seção de Protocolo e Distribuição - Parnaíba	Qualquer curso de nível superior
				1	Central de Mandados - Parnaíba	Qualquer curso de nível superior
			<b>PICOS</b>	1	Seção de Protocolo e Distribuição - Picos	Qualquer curso de nível superior
				1	Central de Mandados - Picos	Qualquer curso de nível superior
			<b>PIRIPIRI</b>	1	Seção de Protocolo e Distribuição - Piripiri	Qualquer curso de nível superior
				1	Central de Mandados – Piripiri	Qualquer curso de nível superior
	<b>Oficial de Gabinete de Magistrado</b>	<del>204</del> <b>205</b>	<sup>1ª</sup> INSTÂNCIA	32	Gabinete de Magistrado - Juizados Especiais	Qualquer curso de nível superior
			<sup>1ª</sup> INSTÂNCIA	152	Gabinete de Magistrado - Varas	
			NUPEMEC	1	NUPEMEC	
			PRES	1	Secretaria da Presidência	

			VICE PRES	1	Gabinete da Vice-Presidência	
			GABCOR	1	Gabinete do Corregedor	
			VICECOR	1	Gabinete do Vice-Corregedor	
			2º grau	<del>15</del> 16	Gabinetes dos Desembargadores	
	Oficial da Corregedoria de Presídios	8	VARAS	8	Varas de Execução Penal	Bacharelado em Direito
	Oficial da Audiência de Custódia	1	CENTRAL/ THE	1	Central de Inquéritos	Bacharelado em Direito
FC/01	Distribuidor Judiciário de 1º Grau - Teresina	1	DIR. FÓRUM	1	Distribuição do 1º Grau - Teresina	Bacharelado em Direito
	Distribuidor Judiciário de 2º Grau	1	SEJU	1	Distribuição do 2º Grau	Bacharelado em Direito
FC/02	Presidente de Comissão de Licitação	2	SLC	2	Superintendência de Licitações e Contratos	Qualquer curso de nível superior
	Presidente de Comissão de Processo Administrativo	3	1º grau	1	Comissão de Processo Administrativo – Servidores/Extrajudiciais	Bacharelado em Direito
			2º grau	1	Comissão de Processo Administrativo - Servidores	
					1	Comissão de Processo Administrativo - Contratual
	Secretario de Vara	164	1º GRAU	1	Central de Inquéritos	Bacharelado em Direito ou Administração
				152	Varas	
2º GRAU			1	Centro Judiciário de Solução de Conflitos e Cidadania – 2º grau		
			1	NUPEMEC		
1º GRAU		9	Centro Judiciário de Solução de Conflitos e Cidadania – 1º grau			
FC/03	Pregoeiro	2	SLC	2	Superintendência de Licitações e Contratos	Qualquer curso de nível superior
	Secretário Assistente de Fórum	84	1º GRAU	1	Fórum Central de Teresina	
				1	Diretoria de Fórum de Campo Maior	
				1	Diretoria de Fórum de Floriano	
				1	Diretoria de Fórum de Parnaíba	
				1	Diretoria de Fórum de Picos	
				1	Diretoria de Fórum de Piri-piri	
				46	Varas	
				32	Juizados Especiais	

	Membros da CCPAD	6	1º grau	2	Comissão de Processo Administrativo – Servidores/Extrajudiciais	
			2º grau	2	Comissão de Processo Administrativo - Servidores	
				2	Comissão de Processo Administrativo – Contratual	
	Oficial de Tecnologia da Informação	5	STIC	5	Secretaria de Tecnologia e Informação	
	Oficial da CEJAI	1	SECCOR	1	Secretaria da Corregedoria	
FC/PM-01	Oficial (Coronel, Major, Ten. Coronel, Capitão, Tenentes) - Companhia da Guarda do Tribunal de Justiça	9	SUSEG	9	Superintendência de Segurança	Militar I
FC/PM-02	Subtenente e Sargentos (1º, 2º e 3º)- Companhia da Guarda do Tribunal de Justiça	35	SUSEG	35	Superintendência de Segurança	Militar II
FC/PM-03	Cabo e Soldado - Companhia da Guarda do Tribunal de Justiça	109	SUSEG	109	Superintendência de Segurança	Militar III

**ANEXO XI**  
**Funções comissionadas dos Policiais Militares - FC/PM**

<b>CARGO/FUNÇÃO</b>	<b>SÍMBOLO</b>	<b>VALOR (R\$)</b>	<b>QUANTIDADE</b>
Oficiais (Coronel, Major, Ten. Coronel, Capitão, Tenentes) - Companhia da Guarda do Tribunal de Justiça	FC/PM-01	1.300,00	9
Subtenentes e Sargentos (1º, 2º e 3º)- Companhia da Guarda do Tribunal de Justiça	FC/PM-02	1.100,00	35
Cabos e Soldados - Companhia da Guarda do Tribunal de Justiça	FC/PM-03	900,00	109



**TRIBUNAL DE JUSTIÇA DO ESTADO DO PIAUÍ**

# **REGIMENTO INTERNO**

**Março/2017**



## **TRIBUNAL DE JUSTIÇA DO ESTADO DO PIAUÍ**

Desembargador ERIVAN LOPES, Presidente

Desembargador JOSÉ JAMES GOMES PEREIRA, Vice- Presidente

Desembargador RICARDO GENTIL EULÁLIO DANTAS, Corregedor Geral da Justiça

Desembargador LUIZ GONZAGA BRANDÃO DE CARVALHO

Desembargador RAIMUNDO NONATO DA COSTA ALENCAR

Desembargador EDVALDO PEREIRA DE MOURA

Desembargadora EULÁLIA MARIA RIBEIRO GONÇALVES N. PINHEIRO

Desembargador JOSÉ RIBAMAR OLIVEIRA

Desembargador FERNANDO CARVALHO MENDES

Desembargador HAROLDO OLIVEIRA REHEM

Desembargador RAIMUNDO EUFRÁSIO ALVES FILHO

Desembargador JOAQUIM DIAS DE SANTANA FILHO

Desembargador FRANCISCO ANTÔNIO PAES LANDIM FILHO

Desembargador SEBASTIÃO RIBEIRO MARTINS

Desembargador PEDRO DE ALCÂNTARA DA SILVA MACÊDO

Desembargador JOSÉ FRANCISCO DO NASCIMENTO

Desembargador HILO DE ALMEIDA SOUSA

Desembargador RICARDO GENTIL EULÁLIO DANTAS

Desembargador OTON MÁRIO JOSÉ LUSTOSA TORRES

Desembargador FERNANDO LOPES E SILVA NETO

## SUMÁRIO

### DISPOSIÇÃO INICIAL

Disposição Inicial.....	5
-------------------------	---

### PARTE I

#### DA ORGANIZAÇÃO E DA COMPETÊNCIA

Capítulo I – Da Composição do Tribunal .....	5
Capítulo II – Dos Órgão de Direção.....	6
Seção I - Do Presidente, do Vice-Presidente e do Corregedor Geral da Justiça.....	6
Seção II - Da escolha do Presidente e do Vice-Presidente e do Corregedor Geral da Justiça.....	7
Capítulo III – Dos Desembargadores.....	8
Seção I - Das Nomeações e da Posse.....	8
Seção II - Das Proibições, das Incompatibilidades, dos Impedimentos e das Suspeições.....	9
Seção III - Das Garantias, das Prerrogativas e dos Vencimentos dos Desembargadores.....	10
Seção IV - Das Férias, Licenças e Concessões.....	11
Seção V - Das Substituições.....	13
Seção VI - Dos Deveres, das Proibições e da Responsabilidade civil.....	14
Seção VII - Da Aposentadoria.....	16
Capítulo IV – Das Atribuições.....	17
Seção I - Das atribuições do Tribunal Pleno.....	17
Seção II - Das Atribuições das Câmaras Reunidas.....	22
Seção III - Das Atribuições das Câmaras Especializadas Cíveis.....	24
Seção IV - Das Atribuições das Câmaras Criminais.....	24
Seção V - Das Atribuições do Presidente do Tribunal.....	25
Seção VI - Das Atribuições do Vice-Presidente.....	27
Seção VII - Das Atribuições dos Presidentes das Câmaras Reunidas.....	28
Seção VIII - Das Atribuições dos Presidentes das Câmaras Especializadas.....	28
Seção IX - Das Atribuições dos Relatores.....	28
Seção X - Das Atribuições dos Revisores.....	31
Seção XI - Da Representação para a Perda do Posto e Patente e da Graduação.....	31
Capítulo V - Do Conselho da Magistratura.....	31
Capítulo VI - Da Corregedoria Geral da Justiça.....	32
Seção I - Disposições Gerais.....	32
Seção II - Das Atribuições do Corregedor Geral.....	32
Capítulo VII - Da Polícia do Tribunal.....	32
Capítulo VIII - Da Representação por Desacato.....	32

#### PARTE II - DAS ATIVIDADES JURISDICIONAIS E ADMINISTRATIVAS

Capítulo IX - Dos Atos e Formalidades.....	33
Seção I - Disposições Gerais.....	33
Seção II - Do Registro e da Classificação dos Feitos.....	35
Seção III - Do Preparo.....	36
Seção IV - Da Distribuição.....	37
Seção V - Das Atas e dos Termos.....	39
Seção VI - Das Decisões.....	40
Capítulo X - Das Sessões e das Audiências.....	41

Seção I - Das Sessões Ordinárias, Extraordinárias e Solenes.....	41
Seção II - Das Audiências.....	53
Capítulo XI - Dos Feitos de Competência Originária do Tribunal.....	53
Seção I -Do Pedido Originário de Habeas <i>Corpus</i> .....	53
Seção II - Do Pedido Originário de Mandado de Segurança.....	54
Seção III - Da Ação Penal Originária.....	55
Seção IV - Da Revisão Criminal.....	58
Seção V - Da Ação Rescisória.....	60
Seção VI - Do Conflito de Jurisdição ou Competência e de Atribuições.....	61
Seção VII - Da Perda de Cargo, da Disponibilidade e da Remoção Compulsória de Magistrados.....	63
Seção VIII - Da Imposição das Penalidades de Advertência e de Censura.....	64
Seção IX - Da Declaração da Perda de Posto e de Patente de Oficiais e da Graduação das Praças.....	65
Capítulo XII - Dos Processos Incidentes.....	66
Seção I - Das Arguições de Incompetência, de Suspeição e de Impedimento.....	66
Seção II - Da Habilitação Incidente.....	70
Seção III – Do Incidente de Arguição de Inconstitucionalidade.....	72
Seção IV – Do Incidente de Assunção de Competência.....	72
Seção V - Da Suspensão de Liminar em Mandado de Segurança e da Suspensão Provisória de Execução de Sentença.....	73
Seção VI - Do Desaforamento de Julgamento.....	73
Seção VII - Da Impugnação ao Valor da Causa.....	73
Seção VIII - Do Incidente de Falsidade.....	74
Seção IX - Da Restauração de Autos.....	75
Seção X - Da Reclamação.....	75
Seção XI – Do Incidente de Resolução de Demandas Repetitivas.....	76
Capítulo XII - A Procedimentos de Uniformização De Jurisprudência.....	77
Seção I - Disposições Gerais.....	77
Seção III - Do Incidente de Arguição de Inconstitucionalidade de Lei ou Ato Normativo.....	83
Capítulo XIII - Dos Recursos para o Tribunal de Justiça.....	85
Seção I - Dos Recursos Criminais.....	85
Seção II - Dos Recursos Cíveis.....	86
Capítulo XIV - Dos Recursos de Atos, Deliberação e Decisões dos Órgãos Jurisdicionais e Administrativos do Tribunal.....	89
Seção I - Dos Embargos Declaratórios.....	89
Seção II - Dos Embargos Infringentes e de Nulidade.....	90
Seção III – Do Agravo Interno.....	91
Seção IV - Dos Recursos Administrativos.....	91
Seção V - Do Recurso Extraordinário e do Recurso Especial.....	92
Capítulo XV – Da Execução ou do Cumprimento das Decisões do Tribunal nas Causas de sua Competência Originária.....	93
Seção I - Disposições Gerais.....	93
Seção II – Do Cumprimento de Decisão Judicial e da Execução de Título Extrajudicial contra a Fazenda Pública.....	94
Seção III – Do Cumprimento Provisório de Decisão Judicial.....	96
Seção IV - Da Intervenção do Estado em Município.....	97
Capítulo XVI - Dos Atos Normativos.....	97
Capítulo XVII - Dos Atos de Provimento e de Vacância.....	98
Seção I - Das Nomeações para os Cargos da Secretaria e dos Serviços Auxiliares.....	98
Seção II - Das Secretarias dos Juizados Especiais.....	99
Capítulo XVIII - Das Comissões.....	100

## DISPOSIÇÕES FINAIS

Disposições finais.....	101
-------------------------	-----

# **RESOLUÇÃO Nº 02, DE 12 DE NOVEMBRO DE 1987**

## **REGIMENTO INTERNO DO TRIBUNAL DE JUSTIÇA DO ESTADO DO PIAUÍ**

- Na medida do possível, o texto encontra-se ajustado com as regras de redação legislativa estabelecidas pela Lei Complementar n. 95, de 26 de fevereiro de 1998, e pela Lei nº 13.105, de 16 de março de 2015 (Código de Processo Civil)

**O EGRÉGIO TRIBUNAL DE JUSTIÇA DO ESTADO DO PIAUÍ**, instalado a 1º de outubro de 1891, usando da faculdade outorgada pelo art. 115, inciso III, da Constituição da República Federativa do Brasil; pelo art. 21, III, da Lei Orgânica da Magistratura Nacional; pelo art. 54, II, da Constituição do Estado do Piauí e pelo art. 287, da Lei nº 3.716, de 12 de dezembro de 1979, APROVA, por unanimidade de votos de seus membros, a presente RESOLUÇÃO Nº 02/87, que dispõe sobre seu

### **REGIMENTO INTERNO**

#### **DISPOSIÇÃO INICIAL**

Art. 1º Este Regimento dispõe sobre a competência e o funcionamento dos órgãos do Tribunal de Justiça do Estado do Piauí.

#### **PARTE I**

(A expressão “Parte I” consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça.)

#### **DA ORGANIZAÇÃO E DA COMPETÊNCIA**

##### **CAPÍTULO I - DA COMPOSIÇÃO DO TRIBUNAL**

Art. 2º O Tribunal de Justiça do Estado do Piauí compõe-se de treze juízes com a denominação própria de Desembargadores, tem sede na Capital e jurisdição em todo o território estadual. (O Tribunal de Justiça do Piauí tem atualmente 19 (dezenove) Desembargadores, conforme a redação do art. 12 da Lei Ordinária nº 3.716, de 12 de dezembro de 1979, modificada pela Lei Complementar nº 169, de 19 de julho de 2011)

§1º Dependerá de proposta do Tribunal de Justiça a alteração do número de seus membros. (Parágrafo único transformado em parágrafo 1º pelo art. 1º da Resolução n.º 02/2012, de 12/01/2012)

§2º O Tribunal Pleno do Tribunal de Justiça poderá autorizar a realização de sessão de julgamento do Plenário e de suas Câmaras na sede de comarca do interior do Estado. (Redação dada pelo art. 1º da Resolução n.º 02/2012, de 12/01/2012)

Art. 3º O Tribunal de Justiça, na prestação da tutela jurisdicional, funcionará em

Plenário, em Câmaras Especializadas, sendo quatro Cíveis e duas Criminais, e em Câmaras Reunidas, com as atribuições e competências que lhes são cometidas neste Regimento Interno, com observância das normas de processo e das garantias processuais das partes. (Redação dada pelo art. 1º da Resolução nº 09/2013, de 27/06/2013)

§ 1º As Câmaras Reunidas são constituídas dos juízes que estiverem integrando as câmaras cíveis e as câmaras criminais. (Redação dada pelo art. 1º da Resolução nº 02/2003, de 13/02/2003)

§ 2º As Câmaras Especializadas Cíveis e Criminais são constituídas de no mínimo três juízes, cada uma. (Redação dada pelo art. 1º da Resolução nº 02/2003, de 13/02/2003)

§ 3º As Câmaras Especializadas são presididas, uma pelo Vice-Presidente do Tribunal que à mesma pertencer e as outras pelos desembargadores eleitos dentre seus membros para mandato de 01 (um) ano. (Redação dada pelo art. 1º da Resolução nº 02/2003, de 13/02/2003)

Art. 4º São integrantes do Tribunal de Justiça, como órgãos auxiliares, o Conselho da Magistratura, a Corregedoria Geral da Justiça, as Secretarias e os serviços auxiliares, e os Gabinetes do Presidente e dos Desembargadores.

## **CAPÍTULO II - DOS ÓRGÃOS DE DIREÇÃO**

### **Seção I - Do Presidente, do Vice-Presidente e do Corregedor Geral da Justiça**

Art. 5º O Tribunal de Justiça é dirigido por um dos seus membros como Presidente, dois outros desempenham as funções de Vice-Presidente e Corregedor Geral da Justiça.

Art. 6º O Presidente, o Vice-Presidente e o Corregedor Geral serão eleitos pela maioria dos membros efetivos do Tribunal de Justiça, mediante votação secreta, dentre seus juízes mais antigos e desimpedidos, com mandato por dois anos, proibida a reeleição.

Art. 7º É obrigatória a aceitação do cargo, salvo recusa manifestada e aceita antes da eleição.

Art. 8º Cada um dos três Desembargadores mais antigos desimpedidos poderá manifestar a sua recusa até a sessão ordinária imediatamente anterior à sessão em que se tiver de proceder à escolha dos dirigentes.

Art. 9º Na hipótese de um dos três Desembargadores mais antigos desimpedidos manifestar recusa quanto à aceitação de algum dos cargos de direção, ou de todos eles, integrará a lista o nome do Desembargador que se seguir em ordem de antiguidade e também estiver desimpedido.

§ 1º O Desembargador, cujo nome, em vista de recusa de qualquer dos mais antigos, venha a compor a lista de votação, poderá manifestar a recusa até quarenta e oito horas antes da eleição.

§ 2º Em caso de recusa prevista no parágrafo anterior, o Desembargador cujo nome venha a integrar a lista de votação, poderá expressar a recusa até o momento da eleição.

Art. 10. Não havendo recusa quanto à totalidade dos cargos de direção, pelo Desembargador mais antigo, aquele que vier a integrar a lista, como substituto, será elegível apenas para o cargo ou os cargos em relação aos quais tenha havido manifestação de recusa do mais antigo, figurando este como elegível para os demais

cargos.

## **Seção II - Da escolha do Presidente e do Vice-Presidente e do Corregedor Geral da Justiça**

Art. 11. O Presidente e o Vice-Presidente do Tribunal e o Corregedor Geral da Justiça serão eleitos no 15º (décimo quinto) dia útil anterior ao dia do término do mandato dos ocupantes destes cargos, que é por dois anos, devendo os eleitos tomarem posse no primeiro dia útil após o término do mandato do respectivo titular. [\(Redação dada pelo art. 1º da Resolução nº 02, de 24/08/1995\)](#)

Art. 12. Na sessão ordinária imediatamente anterior àquela em que se tiver de proceder à eleição, o Plenário do Tribunal aprovará a lista com os nomes dos três Desembargadores mais antigos e desimpedidos, dentre os quais serão escolhidos o Presidente, o Vice-Presidente e o Corregedor Geral da Justiça.

Art. 13. Antes do início da votação para os cargos de direção, os Desembargadores presentes deliberarão sobre se aceitam a recusa nos casos previstos nos artigos 8º e 9º, §§ 1º e 2º, deste Regimento.

Art. 14. Não figurarão como elegíveis, na lista de votação:

- a) quem tiver exercido, por quatro anos, cargos de direção.
- b) quem tiver exercido o cargo de Presidente;
- c) quem tiver exercido mandato de direção no período imediatamente anterior àquele a que se referir a eleição.

§ 1º O impedimento a que se refere a letra “b” não atinge os que houverem exercido, eventualmente, a Presidência do Tribunal, na condição de substituto, seja como Vice-Presidente, seja em razão da ordem de antiguidade, salvo se o período de substituição for superior a um ano.

§ 2º Esgotados todos os nomes, na ordem de antiguidade, deixarão de subsistir os impedimentos a que se referem as letras “a” e “b”.

§ 3º A inelegibilidade constante da letra “c” do "caput" é restrita a novo mandato para o mesmo cargo, no período imediato, não constituindo impedimento à eleição para cargo de direção diverso.

Art. 15. A cédula de votação conterá os nomes dos três Desembargadores mais antigos e desimpedidos, figurando, ao lado, três colunas, nas quais serão assinalados, respectivamente, os votos para Presidente, para Vice-Presidente e para Corregedor Geral da Justiça.

Parágrafo único. Havendo manifestação de recusa quanto a algum ou a alguns dos cargos, a cédula de votação conterá também o nome do Desembargador imediato em antiguidade e desimpedido, mas, ao lado desse nome, figurará apenas a coluna correspondente ao cargo a que tenha havido recusa de aceitação.

Art. 16. Não serão computados votos conferidos, na mesma cédula, a dois nomes para o mesmo cargo, bem como os constantes de cédulas com marcas ou sinais que possibilitem identificação ou quebra de sigilo.

Art. 17. Não se considera eleito quem não obtiver mais da metade dos votos dos Desembargadores do Tribunal; se nenhum reunir essa votação, realizar-se-á novo escrutínio entre os dois mais votados, considerando-se eleito, em caso de empate, o

mais antigo no Colegiado.

Art. 18. Ocorrendo vaga em qualquer dos cargos de direção, em caráter definitivo, haverá nova eleição, no prazo fixado no artigo 11, para seu preenchimento com mandato de dois anos. (Redação dada pelo art. 1º da Resolução nº 02, de 24/08/1995)

Parágrafo único. (Revogado) (Redação dada pelo art. 2º da Resolução nº 02, de 24/08/1995)

Art. 19. (Revogado) (Redação dada pelo art. 2º da Resolução nº 02, de 24/08/1995)

Art. 20. O Presidente e o Vice-Presidente do Tribunal de Justiça, bem como o Corregedor Geral da Justiça, não poderão participar do Tribunal Regional Eleitoral, inclusive como suplentes.

§ 1º São elegíveis para os cargos de direção do Tribunal de Justiça, os participantes do Tribunal Regional Eleitoral que, em vista de exercerem mandato correspondente a segundo biênio, renunciarem ao restante deste mandato, até a sessão ordinária imediatamente anterior àquela em que se tiver de proceder à escolha dos cargos de direção.

§ 2º O Desembargador que já houver servido ao Tribunal Regional Eleitoral por um biênio é inelegível para outro período até que se esgotem todos os nomes dos Desembargadores que ainda assento não tiveram naquela Corte, observada a ordem de antiguidade desses no Tribunal de Justiça.

### **CAPÍTULO III - DOS DESEMBARGADORES**

#### **Seção I - Das Nomeações e da Posse**

Art. 21. Os Desembargadores serão nomeados pelo Governador do Estado, observadas as normas da Constituição, da Lei Orgânica da Magistratura Nacional e da Lei de Organização Judiciária do Estado.

Art. 22. Ressalvados os lugares que tenham de ser preenchidos por advogados ou membros do Ministério Público, as vagas dos Desembargadores serão providas mediante acesso, por antiguidade e por merecimento, alternadamente, de Juízes de Direito.

Art. 23. Tratando-se de acesso ou antiguidade, de Juiz de Direito, o ato de provimento, expedido pelo Chefe do Poder Executivo, recairá no Juiz indicado pelo Tribunal de Justiça.

Art. 24. A antiguidade será apurada na última entrância e o Tribunal de Justiça somente poderá recusar o Juiz mais antigo pelo voto da maioria dos Desembargadores, repetindo-se a votação até fixar-se a indicação.

Art. 25. No caso de acesso por merecimento, o Tribunal organizará lista tríplice com os nomes escolhidos dentre os Juízes de qualquer entrância.

Art. 26. O Juiz de Direito promovido ao cargo de Desembargador poderá recusar o acesso.

Art. 27. Na composição do Tribunal de Justiça, um quinto dos lugares será preenchido por advogados, em efetivo exercício da profissão, e membros do Ministério Público, todos de notório saber jurídico e reputação ilibada, com dez ou mais anos de prática forense, depois de formados, dos quais os cinco últimos da classe a que

pertencer a vaga, observado o art. 100 da Lei Orgânica da Magistratura Nacional.

§ 1º Somente membros do Ministério Público de carreira podem concorrer ao preenchimento da vaga.

§ 2º O preenchimento será feito alternadamente, uma vaga para advogado e outra para membro do Ministério Público, não podendo ser votado para o lugar daquele o integrante deste, ainda que exerça a advocacia.

§ 3º Os lugares reservados a membros do Ministério Público ou advogados serão preenchidos, respectivamente, por membros do Ministério Público ou advogados, indicados em lista tríplice pelo Tribunal de Justiça, e que houverem requerido inscrição, cumprindo as exigências legais, ao presidente do Tribunal, contados do edital publicado no Diário da Justiça pelo prazo de quinze dias.

§ 4º Não se consideram membros do Ministério Público, para preenchimento de vagas do Tribunal de Justiça, os juristas estranhos à carreira, nomeados em comissão para o cargo de Procurador Geral da Justiça ou de outro de chefia.

§ 5º A prática forense resultará comprovada através do exercício de advocacia, do desempenho de cargo de Ministério Público ou de atividades forenses privativas de Bacharel em Direito.

Art. 28. As listas de nomeação e acesso serão organizadas em sessão reservada e por escrutínio secretos, tomando parte na organização das mesmas os Desembargadores em férias ou licenciados, observando-se as normas relativas à prévia ciência dos votantes, constantes do art. 115.

Art. 29. Antes de assumir o exercício de seu cargo, o Desembargador, exibindo o título de nomeação devidamente legalizado, tomará posse perante o Presidente do Tribunal de Justiça, prestando o compromisso de cumprir com retidão os seus deveres, sendo, na ocasião, lavrado termo em livro próprio.

§ 1º A posse deverá verificar-se dentro de trinta dias, contados da publicação do ato no “Diário Oficial” ou no “Diário da Justiça”, podendo esse prazo ser prorrogado, por mais trinta dias, pelo Presidente do Tribunal de Justiça.

§ 2º O exercício deverá iniciar-se dentro de trinta dias, a contar da data da posse.

Art. 30. O início, a interrupção e o reinício do exercício serão comunicados, dentro de vinte e quatro horas, ao Presidente do Tribunal.

## **Seção II - Das Proibições, das Incompatibilidades, dos Impedimentos e das Suspeições**

Art. 31. É vedado ao Desembargador, sob pena de perda do cargo:

I – exercer, ainda que em disponibilidade, qualquer outra função, salvo em cargo de magistério superior, público ou particular;

II – receber, a qualquer título e sob qualquer pretexto, percentagens ou custas nos processos sujeitos a seu despacho e julgamento;

III – exercer atividade político-partidária.

Art. 32. As incompatibilidades, os impedimentos e as suspeições dos Desembargadores ocorrerão nos casos previstos em lei.

§ 1º No Tribunal de Justiça não poderão ter assento, na mesma Câmara, cônjuge e



parentes consanguíneos ou afins em linha reta, bem como em linha colateral até o terceiro grau.

§ 2º Nas sessões do Tribunal Pleno, o primeiro dos membros mutuamente impedidos, que votar, excluirá a participação do outro no julgamento.

Art. 33. O Desembargador que se julgar suspeito ou impedido deverá declará-lo nos autos e, se for revisor, passará o feito ao respectivo substituto, ou, se relator, apresentará os autos em Mesa, para nova distribuição.

§ 1º Se não for relator nem revisor, o Desembargador que houver de se dar por suspeito ou impedido, deverá fazê-lo verbalmente, na sessão de julgamento, registrando-se na ata a declaração.

§ 2º Se o Presidente do Tribunal se der por suspeito ou impedido, competirá ao seu substituto a presidência do julgamento.

§ 3º Será observado no que couber, quanto à arguição de suspeição ou impedimento, pela parte, o que dispõe a lei processual relativamente ao processamento das suspeições e impedimentos opostos a juízes singulares.

§ 4º A suspeição, não sendo reconhecida, será julgada pelo Tribunal Pleno, funcionando com relator o Presidente.

§ 5º Se o recusado for o Presidente, o relator será o Vice-Presidente.

### **Seção III - Das Garantias, das Prerrogativas e dos Vencimentos dos Desembargadores**

Art. 34. Os Desembargadores, depois de empossados, serão vitalícios, não podendo perder o cargo senão por sentença judiciária.

Art. 35. A perda do cargo de Desembargador somente ocorrerá:

I – em ação penal por crime comum ou de responsabilidade;

II – em procedimento administrativo, nas hipóteses previstas no art. 31.

Art. 36. São prerrogativas do Desembargador:

I – ser ouvido como testemunha em dia, hora, e local previamente ajustados com a autoridade ou juiz de instância igual ou inferior;

II – não ser preso senão por ordem escrita do Tribunal competente para o julgamento, salvo em flagrante de crime inafiançável, caso em que a autoridade fará imediata comunicação e apresentação do magistrado ao Tribunal de Justiça;

III – ser recolhido à prisão especial, ou à sala especial de Estado-Maior, por ordem e à disposição do Tribunal competente, quando sujeito a prisão antes do julgamento final;

IV – não está sujeito à notificação ou a intimação para comparecimento, salvo se expedida por autoridade judicial;

V – portar arma de defesa pessoal.

Art. 37. Quando, no curso de investigação, houver indício da prática de crime por parte de Desembargador, a autoridade policial, civil ou militar, remeterá os respectivos autos ao Tribunal competente para o julgamento, a fim de que prossiga na investigação.

Art. 38. O título de Desembargador é privativo dos membros do Tribunal de Justiça.

Art. 39. Os vencimentos dos Desembargadores, fixados em Lei, em valor certo, são irredutíveis, sujeitos, entretanto, aos impostos gerais, inclusive o de renda, e aos impostos extraordinários.

Parágrafo único. A irredutibilidade dos vencimentos dos magistrados não impede os descontos fixados em lei, em base igual à estabelecida para os servidores públicos, para fins previdenciários.

Art. 40. Os vencimentos dos Desembargadores não serão inferiores, em nenhum caso, aos de Secretário de Estado, não podendo ultrapassar, porém, os fixados para os Ministros do Supremo Tribunal Federal.

Parágrafo único. Para efeito de equivalência e limite de vencimentos previstos neste artigo, são excluídas do cômputo apenas as vantagens de caráter pessoal ou de natureza transitória.

Art. 41. Os vencimentos dos Desembargadores serão pagos na mesma data fixada para o pagamento dos vencimentos dos Secretários de Estado ou dos subsídios dos membros do Poder Legislativo, considerando-se que desatende às garantias do Poder Judiciário atraso que ultrapasse o décimo dia útil do mês seguinte ao vencido.

Art. 42. Além dos vencimentos, poderão ser outorgadas, aos Desembargadores, nos termos da lei, as seguintes vantagens:

I – representação mensal, fixada em lei, sobre o vencimento básico;

II – gratificação adicional por tempo de serviço, não excedente a trinta e cinco por cento dos vencimentos, computando-se a partir dos cinco anos de serviço público, cinco por cento por quinquênio;

III – ajuda de custo, para despesa de transporte e moradia;

IV – salário-família;

V – diárias;

VI – representação de função;

VII – gratificação pela prestação de serviço à Justiça Eleitoral;

VIII – gratificação por exercício do magistério em cursos de aperfeiçoamento de magistrados.

§ 1º A verba de representação, salvo em exercício de cargo em função temporária, integra os vencimentos para todos os efeitos legais.

§ 2º É proibida qualquer outra vantagem não prevista em lei.

#### **Seção IV - Das Férias, Licenças e Concessões**

*(Seção com designação alterada pelo art. 3º da Resolução nº 13, de 29/09/2005)*

Art. 43. Os Desembargadores terão direito a licença para tratamento de sua saúde e, bem assim, por motivo de doença em pessoa da família.

Art. 44. A licença para tratamento de saúde por prazo superior a trinta dias bem como as prorrogações que importem em licença por período ininterrupto, também superior a trinta dias, dependem de inspeção por junta médica.

Art. 45. Os requerimentos, assinados pelo próprio Desembargador ou, na impossibilidade, por pessoa de sua família, devem especificar a razão e o prazo do licenciamento solicitado e a data a partir da qual se pretende o afastamento, fazendo-se acompanhar da competente atestação médica, quando não for o caso de inspeção por junta.

Art. 46. Será dispensável requerimento de licença para ausências que não excedam a três dias durante o mês, aplicando-se a legislação pertinente ao funcionalismo estadual.

Art. 47. Os períodos de licença concedidos aos Desembargadores não terão limites inferiores reconhecidos por lei ao funcionalismo estadual.

Art. 48. O Desembargador licenciado na conformidade do art. 43, deste Regimento, não pode exercer qualquer das funções jurisdicionais ou administrativas, ressalvado o direito de voto nas deliberações de ordem administrativa, nem pode exercitar qualquer função pública ou administrativa.

Parágrafo único - Salvo contra-indicação médica, o Desembargador licenciado poderá proferir decisões ou participar de julgamento em processo que, antes da licença, tenha recebido o seu visto como relator ou revisor.

Art. 49. Sem prejuízo do vencimento, remuneração ou de qualquer direito ou vantagem legal, o Desembargador poderá afastar-se de suas funções até oito dias consecutivos por motivo de:

I – casamento

II – falecimento de cônjuge, ascendente, descendente ou irmão.

Parágrafo único. Será concedido afastamento ao Desembargador, sem prejuízo de seus vencimentos e vantagens:

I – para frequência a curso ou seminários de aperfeiçoamento e estudo, a critério do Tribunal, pelo prazo máximo de um ano, podendo, a critério do Tribunal, ser-lhe atribuída ajuda de custo correspondente a até cem por cento dos seus vencimentos.

II – para a prestação de serviços exclusivamente à Justiça Eleitoral, por período não excedente a dois anos.

### **Subseção única - Das férias**

(Subseção acrescentada pelo art. 3º da Resolução nº 13, de 29/09/2005)

Art. 49-A. Os Desembargadores desfrutarão férias anuais individuais de 60 (sessenta) dias, conforme escala organizada pelo Tribunal de Justiça, no mês de dezembro, de acordo com as preferências manifestadas, obedecidas a rotativa antiguidade no cargo e as necessidades do serviço. (Incluído pelo art. 3º da Resolução nº 13, de 29/09/2005)

Parágrafo único. O afastamento do Desembargador por motivo de férias não poderá comprometer a prestação da atividade jurisdicional do Tribunal de forma ininterrupta. (Incluído pelo art. 3º da Resolução nº 13, de 29/09/2005)

Art. 49-B. As férias não poderão ser fracionadas em períodos inferiores a 30 (trinta) dias e somente poderão acumular-se por imperiosa necessidade de serviço e pelo prazo máximo de 2 (dois) meses, mediante autorização do Presidente do Tribunal. (Incluído pelo art. 3º da Resolução nº 13, de 29/09/2005)

Art. 49-C. É vedado o afastamento simultâneo de Desembargadores em número que possa comprometer o *quorum* de julgamento no Tribunal Pleno, nas Câmaras Reunidas e nas Câmaras Especializadas. (Incluído pelo art. 3º da Resolução nº 13, de 29/09/2005)

Art. 49-D. Não poderão afastar-se, simultaneamente, o Presidente e o Vice-Presidente do Tribunal, em face de férias individuais e para frequência a cursos ou seminários de aperfeiçoamento. (Incluído pelo art. 3º da Resolução nº 13, de 29/09/2005)

## Seção V - Das Substituições

Art. 50. As substituições de Desembargadores, nas licenças, faltas e impedimentos, serão processadas entre os próprios membros do Tribunal, somente havendo convocação de Juiz de Direito em casos excepcionais, na forma prevista neste regimento.

Art. 51. O Presidente do Tribunal será substituído pelo Vice-Presidente, e este e o Desembargador Corregedor Geral da Justiça, pelos demais membros desimpedidos, na ordem decrescente de antiguidade.

Art. 52. Os Presidentes das Câmaras Reunidas e das Câmaras Especializadas, nas licenças, faltas e impedimentos, serão substituídos pelos Desembargadores mais antigos dos respectivos órgãos judicantes, desimpedidos.

Art. 53. O Relator é substituído: [\(Redação dada pelo art. 1º da Resolução nº 05/2007, de 10/04/2007\)](#)

I – pelo Revisor, se houver, ou pelo Desembargador imediato em antiguidade, dentre os do Tribunal ou da Câmara, conforme a competência, em caso de ausência ou impedimento eventual, quando se tratar de deliberação sobre medida urgente; [\(Redação dada pelo art. 1º da Resolução nº 05/2007, de 10/04/2007\)](#)

II – pelo Desembargador designado para lavrar o acórdão, quando vencido no julgamento; [\(Redação dada pelo art. 1º da Resolução nº 05/2007, de 10/04/2007\)](#)

III – em caso de aposentadoria, renúncia ou morte: [\(Redação dada pelo art. 1º da Resolução nº 05/2007, de 10/04/2007\)](#)

a) pelo Desembargador nomeado para a sua vaga; [\(Redação dada pelo art. 1º da Resolução nº 05/2007, de 10/04/2007\)](#)

b) pelo Desembargador que tiver proferido o primeiro voto vencedor, acompanhando o do Relator, para lavrar ou assinar os acórdãos dos julgamentos anteriores à abertura da vaga; [\(Redação dada pelo art. 1º da Resolução nº 05/2007, de 10/04/2007\)](#)

c) pela mesma forma da letra *b* desde inciso, e enquanto não empossado o novo Desembargador. [\(Redação dada pelo art. 1º da Resolução nº 05/2007, de 10/04/2007\)](#)

Art. 54. O Desembargador revisor será substituído pelo Desembargador do mesmo órgão judicante que se lhe seguir na ordem decrescente de antiguidade.

Art. 55. Fica o Presidente do Tribunal de Justiça autorizado a convocar 02 (dois) juízes de direito da comarca da Capital para atuarem como 1º e 2º suplentes em cada órgão fracionário, em caso de vaga ou afastamento de desembargador, por prazo superior a 30 (trinta) dias. [\(Redação dada pelo art. 1º da Resolução nº 43, de 24/11/2016\)](#)

§ 1º. O magistrado convocado receberá a diferença de vencimentos correspondente ao cargo de desembargador, de acordo com a quantidade de vezes que efetivamente atuou no órgão para o qual foi convocado. [\(Redação dada pelo art. 1º da Resolução nº 43, de 24/11/2016\)](#)

§ 2º. Na convocação de juízes será respeitada a ordem de antiguidade, dando-se preferência entre os juízes de direito de varas não criminais da Capital, para as câmaras cíveis, e de juízes de direito de varas criminais da Capital, para as câmaras criminais. [\(Redação dada pelo art. 1º da Resolução nº 43, de 24/11/2016\)](#)

§ 3º Não poderão ser convocados Juízes de Direito punidos com as penas de advertência, censura, remoção compulsória, disponibilidade com vencimentos proporcionais, bem assim o que estiver respondendo a processo para a decretação da

perda do cargo, ou que não se achar com os serviços de sua vara em dia.

§ 4º. Também não poderão ser convocados os juízes que estiverem exercendo a função de auxiliar da Presidência ou da Corregedoria da Justiça Estadual ou Eleitoral, e os que acumularem as funções eleitorais de primeiro e segundo grau, de direção de fórum, de coordenação de juizados especiais ou de infância e juventude, os titulares dos juizados especiais e os membros titulares das turmas recursais. [\(Incluído pelo art. 1º da Resolução nº 43, de 24/11/2016\)](#)

Art. 56. Somente em caso de afastamento a qualquer título, por período superior a trinta dias, os feitos em poder do magistrado afastado e aqueles em que tenha lançado relatório, como os que puseram em mesa para julgamento, serão redistribuídos aos demais membros do Tribunal Pleno, Câmaras Reunidas ou Câmaras Especializadas, mediante oportuna compensação, e, os feitos em que seja revisor passarão ao substituto legal.

Art. 57. Quando o afastamento se der por período igual ou superior a três dias, serão redistribuídos, mediante oportuna compensação, os *habeas-corpus*, os mandados de segurança e os feitos que, consoante fundada alegação dos interessados, reclamarem solução urgente. Em caso de vaga, ressalvados esses processos, os demais serão distribuídos ao nomeado para julgamento.

Art. 58. Haverá, nas Câmaras Reunidas e nas Câmaras Cíveis e Criminais, livro próprio, em que serão anotados, em ordem cronológica, os nomes dos Desembargadores convocados como substituto, seja para funcionar como vogal, seja para funcionar como revisor, mencionando-se a data, o número e a natureza do processo.

Art. 59. A antiguidade dos Desembargadores, seja para efeito de substituição, seja para qualquer outro, conta-se da data da posse no cargo.

Parágrafo único. No caso de igualdade de tempo, considera-se mais antigo o primeiro nomeado e, no caso de nomeação na mesma data, o mais idoso.

Art. 60. A redistribuição de feitos, a substituição nos casos de ausência ou de impedimento eventual e a convocação para completar quorum de julgamento não autorizam a concessão de qualquer vantagem, salvo diárias e transporte, se for o caso.

## **Seção VI - Dos Deveres, das Proibições e da Responsabilidade civil**

Art. 61. São deveres do Desembargador:

I – cumprir e fazer cumprir, com independência, serenidade e exatidão, as disposições legais e os atos de ofício;

II – não exceder injustificadamente os prazos para proferir decisões ou despachar;

III – determinar as providências necessárias para que os atos processuais se realizem nos prazos legais;

IV – tratar com urbanidade as partes, os membros do Ministério Público, os advogados, as testemunhas, os funcionários e auxiliares da Justiça, e atender aos que o procurarem, a qualquer momento, quando se tratar de providência que reclame e possibilite solução de urgência;

V – comparecer pontualmente à hora de iniciar-se a sessão e não se ausentar injustificadamente antes de seu término;

VI – exercer assídua fiscalização quanto aos autos sujeitos ao seu exame, especialmente

no que se refere à cobrança de custas e emolumentos, embora não haja reclamação das partes;

VII – manter conduta irrepreensível na vida pública e particular.

Art. 62. É vedado ao Desembargador:

I – exercer o comércio ou participar de sociedade comercial, inclusive de economia mista, exceto como acionista ou quotista;

II – exercer cargo de direção ou técnico de sociedade civil, associação ou fundação, de qualquer natureza ou finalidade, salvo de associação de classe e sem remuneração;

III – manifestar, por qualquer meio de comunicação, opinião sobre processo pendente de julgamento, seu ou de outrem, ou juízo depreciativo sobre despachos, votos ou sentenças, de órgãos judiciais, ressalvada a crítica nos autos e em obras técnicas ou no exercício do magistério.

Art. 63. O Tribunal de Justiça fará publicar mensalmente, no órgão oficial, dados estatísticos sobre seu trabalho no mês anterior, dentre os quais o número de votos que cada um de seus membros, nominalmente indicado, proferiu como relator e revisor; o número de feitos que lhe foram distribuídos no mesmo período; o número de processos que recebeu em

consequência de pedido de vista como revisor; a relação dos feitos que lhe foram conclusos para o voto, despacho e lavratura de acórdão, ainda não devolvidos, embora decorridos os prazos legais, com as datas das respectivas conclusões.

Parágrafo único. Compete ao Presidente do Tribunal velar pela regularidade e pela exatidão das publicações.

Art. 64. A atividade censória do Tribunal de Justiça e do Conselho da Magistratura será exercida com o resguardo devido à dignidade e à independência do magistrado.

Art. 65. Salvo os casos de impropriedade ou excesso de linguagem, o Desembargador não poderá ser punido ou prejudicado pelas opiniões que manifestar ou pelo teor das decisões que proferir.

Art. 66. São penas disciplinares a que está sujeito o Desembargador:

I – disponibilidade com vencimentos proporcionais ao tempo de serviço;

II – aposentadoria compulsória com vencimentos proporcionais ao tempo de serviço;

III – demissão.

Art. 67. O Tribunal de Justiça poderá determinar, por motivo de interesse público, em sessão reservada e escrutínio secreto, pelo voto de dois terços de seus membros efetivos, a disponibilidade de Desembargador, com vencimentos proporcionais ao tempo de serviço.

Parágrafo único. O quorum de dois terços de membros efetivos do Tribunal será apurado em relação ao número de Desembargadores em condições legais de votar, como tal se considerando os não atingidos por impedimento ou suspeição e os não licenciados por motivo de saúde.

Art. 68. O procedimento para a decretação de disponibilidade obedecerá ao prescrito no art. 27 da Lei Complementar nº 35, de 14 de março de 1979 (Lei Orgânica da Magistratura Nacional).

Art. 69. O tempo de disponibilidade imposto ao Desembargador como penalidade não será computado senão para efeito de aposentadoria.

Art. 70. Elevado o número de membros do Tribunal de Justiça ou neste ocorrendo vaga,

serão previamente aproveitados os Desembargadores em disponibilidade.

§ 1º O Desembargador posto em disponibilidade, por determinação do Conselho Nacional da Magistratura ou do Tribunal de Justiça, poderá pleitear o seu aproveitamento, decorridos dois anos de afastamento.

§ 2º O pedido, devidamente instruído e justificado, acompanhado de parecer do Tribunal de Justiça, será apreciado pelo Conselho Nacional da Magistratura, quando a disponibilidade tiver decorrido de determinação do mesmo Conselho e, no caso de deferimento, o aproveitamento será feito a critério do Tribunal.

Art. 71. Responderá por perdas e danos o Desembargador quando:

I – no exercício de suas funções, proceder com dolo ou fraude;

II – recusar, omitir ou retardar, sem justo motivo, providência que deva ordenar de ofício, ou a requerimento das partes.

Parágrafo único. Serão reputadas verificadas as hipóteses previstas no inciso II somente depois que a parte, através de requerimento protocolado na Secretaria, requerer ao Desembargador que determine a providência, e este não lhe atender o pedido dentro de dez dias.

## **Seção VII - Da Aposentadoria**

Art. 72. A aposentadoria do Desembargador será:

I – compulsória aos setenta anos de idade ou por invalidez comprovada;

II – facultativa, após trinta anos de serviço público;

III – punitiva, nos casos do art. 56, da Lei Complementar nº 35, de 14 de março de 1979 (Lei Orgânica da Magistratura Nacional).

Art. 73. Salvo as hipóteses do inciso III, do artigo antecedente, a aposentadoria do Desembargador será sempre com vencimentos integrais.

Art. 74. Será computado, para efeito de aposentadoria e disponibilidade, o tempo de exercício de advocacia, até o máximo de dez anos, em favor dos membros do Tribunal de Justiça.

Art. 75. Os proventos da aposentadoria serão reajustados na mesma proporção dos aumentos de vencimentos concedidos, a qualquer título, aos magistrados em atividade.

Art. 76. A aposentadoria do Desembargador, por invalidez comprovada, terá lugar:

I – em vista de requerimento do próprio magistrado;

II – de ofício, em vista de processo iniciado por ordem do Presidente do Tribunal, em cumprimento de deliberação do Tribunal ou por provocação do Conselho da Magistratura.

Art. 77. Requerida a aposentadoria por invalidez, será o Desembargador submetido a inspeção de Junta Médica. Na hipótese de o laudo concluir pela invalidez definitiva, será o processo encaminhado ao Governador do Estado, para a expedição do ato de aposentadoria.

Art. 78. Tratando-se de verificação *ex-officio* de invalidez, a Portaria do Presidente do Tribunal será distribuída ao Desembargador que tenha de funcionar como relator.

§ 1º Cabe ao relator, inicialmente, mandar citar o magistrado para apresentar, querendo, contestação, no prazo de quinze dias, e requerer a produção de provas que entender necessárias em prol de suas alegações.

§ 2º O Procurador Geral de Justiça terá vista do processo pelo prazo de cinco dias, podendo requerer o que for a bem do interesse público.

§ 3º Apresentada, ou não, defesa, o relator determinará seja o magistrado submetido a inspeção de saúde por Junta Médica, fixando prazo para oferecimento do laudo.

§ 4º Na hipótese de ter havido requerimento para produção de prova, o relator designará dia, hora e local para que a mesma se realize, feitas as intimações devidas, inclusive do Ministério Público.

§ 5º Encontrando-se o paciente em outro Estado, a sua citação e, bem assim, a inspeção de saúde e a produção de provas poderão ser decretadas à autoridade Judiciária da respectiva comarca.

§ 6º Cogitando-se de incapacidade mental, o Presidente do Tribunal nomeará curador ao paciente, sem prejuízo da defesa que este queira oferecer, pessoalmente ou por procurador que lhe aprouver constituir.

§ 7º O paciente deverá ser afastado, desde logo, do exercício do cargo, até final decisão, devendo ficar concluído o processo no prazo de sessenta dias.

§ 8º A recusa do paciente em submeter-se a perícia médica permitirá o julgamento baseado em quaisquer outras provas.

Art. 79. Concluída a instrução do processo o relator apresentará em mesa, para a designação do dia para o julgamento.

Parágrafo único. Se o Tribunal concluir pela incapacidade do magistrado, comunicará imediatamente a decisão ao Poder Executivo, para os devidos fins.

## **CAPÍTULO IV - DAS ATRIBUIÇÕES**

(Embora já tivesse essa mesma denominação na redação original, a designação “DAS ATRIBUIÇÕES” para o Capítulo IV foi repetida pelo art. 4º da Resolução nº 03/1999, de 10/06/1999)

### **Seção I - Das atribuições do Tribunal Pleno**

Art. 80. Além da competência jurisdicional, originária ou recursal, cabe ao Tribunal Pleno deliberar sobre assuntos de ordem interna e de disciplina judiciária.

14

Art. 81. Ao Tribunal Pleno compete: (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

I – processar e julgar originariamente: (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

a) a ação direta de inconstitucionalidade de lei ou ato normativo estadual ou municipal e a ação declaratória de constitucionalidade em face da Constituição do Estado (art. 123, I, C.E); (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

b) nos crimes comuns, o Vice-Governador, os deputados estaduais e o Procurador Geral da Justiça (art. 123, III, “c”, C.E); (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

c) nos crimes comuns e de responsabilidade os Secretários de Estado, o Advogado Geral do Estado e o Procurador Geral da Defensoria Pública, salvo nos crimes de responsabilidade conexos com os do Governador (art. 123, III, “d”, 1 – C.E); (Redação



dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

d) os juízes de direito, os juízes de direito substitutos e os membros do Ministério Público, nos crimes comuns e de responsabilidade (art. 123, III, “d”, “2”); (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

e) o Procurador-Geral do Estado, o Procurador-Geral de Justiça, nos crimes comuns e de responsabilidade (art. 123, III, “e”, e 123, III, “d”, 1); (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

f) os juízes da Justiça Militar, o Comandante Geral da Polícia Militar e os integrantes da carreira da Advocacia Pública e da Defensoria Pública do Estado, nos crimes comuns e de responsabilidade; (Redação dada pelo art. 1º da Resolução nº 03/1999, de 10/06/1999)

g) conflitos de competência que envolvam órgãos fracionários do Tribunal, desembargadores, juízes em exercício no Tribunal ou entre as autoridades judiciárias e administrativas quando neles forem parte o governador, secretário de estado, magistrados ou o procurador geral de Justiça. (Redação dada pelo art. 1º da Resolução nº 06/2016, de 04/04/2016)

h) os conflitos de competência entre os juízes de direito entre si, com os Juizados Especiais e com o Conselho da Justiça Militar; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

i) os mandados de segurança contra atos do Governador, dos Secretários de Estado, da Assembleia Legislativa, de sua Mesa e de seu Presidente, do Tribunal de Justiça e de seu Presidente, das Câmaras Especializadas e de seus Presidentes, do Conselho da Magistratura, do Corregedor Geral da Justiça, dos procuradores-gerais da Justiça e do Estado, do Tribunal de Contas e de seu Presidente, dos juízes de direito e de juízes substitutos e dos Juizados Especiais; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

j) os *habeas corpus*, quando alegado constrangimento partir de autoridade diretamente subordinada ao Tribunal de Justiça; quando se tratar de crime sujeito a esta mesma jurisdição, em única instância; e quando houver perigo de se consumir a violência, antes que outro juízo possa conhecer da espécie; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

l) a execução de sentença proferida em causa de sua competência, facultada a delegação de atos do processo, a Juiz do primeiro grau de jurisdição ou de primeira instância; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

m) as habilitações incidentes nas causas de sua competência; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

n) as ações rescisórias de seus acórdãos; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

o) (Revogado) (Redação dada pelo art. 1º da Resolução nº 03/1999, de 10/06/1999)

p) as revisões e reabilitações, quando as condenações a ele competirem; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

q) as remoções compulsórias de juízes de direito e as reclamações sobre a colocação de juízes nas listas de antiguidade, de nomeação e promoção, e sobre nulidade dos concursos de juízes de direito substitutos; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

r) as incompatibilidades, as suspeições e os impedimentos, opostos e não reconhecidos, aos Desembargadores, ao Procurador-Geral da Justiça e ao Corregedor Geral da Justiça;  
s) os pedidos de revisão criminal; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

t) os embargos a seus acórdãos e, bem assim, aqueles a que se refere o art. 609, parágrafo único, do Código de Processo Penal, e os que forem opostos às decisões, não unânimes, das Câmaras Criminais, nos processos oriundos do Conselho de Justificação da Polícia Militar; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

u) a reabilitação dos condenados, quando houver proferido a sentença condenatória; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

v) as habilitações com feitos pendentes do seu julgamento; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

x) os agravos dos despachos ou atos do Presidente ou dos Relatores, quando da competência do Tribunal; (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

y) a deserção dos recursos nos feitos pendentes do seu julgamento, quando o Presidente ou o Relator não a houver declarado. (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

z) (Revogado) (Redação dada pelo art. 1º da Resolução nº 4/2008, de 27/03/2008).

§ 1º São partes legítimas para promover a ação direta de inconstitucionalidade de lei ou ato normativo estadual ou municipal (art. 124 – CE): (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

I – o Governador do Estado;

II – a Mesa da Assembleia Legislativa;

III – o Procurador-Geral de Justiça;

IV – o Prefeito Municipal;

V – a Mesa da Câmara Municipal;

VI – o Conselho Seccional da Ordem dos Advogados do Brasil;

VII – os partidos políticos com representação na Assembleia Legislativa ou em Câmaras Municipais;

VIII – as federações sindicais e as entidades de classe de âmbito estadual.

§ 2º Podem propor a ação declaratória de constitucionalidade o Governador do Estado, a Mesa da Assembleia Legislativa e o Procurador Geral de Justiça. (art. 103, § 4º, C.F.) (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

§ 3º As decisões definitivas de mérito, proferidas pelo Tribunal de Justiça, nas ações declaratórias de constitucionalidade de lei ou ato normativo estadual, produzirão eficácia contra todos e efeito vinculante, relativamente ao Poder Judiciário e ao Poder Executivo do Estado (art.102, § 2º, C.F.). (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

§ 4º Quando o Tribunal de Justiça apreciar a inconstitucionalidade, em tese, de norma legal ou ato normativo, citará previamente o Advogado Geral, que defenderá o ato ou texto impugnado, ou, em se tratando de norma legal ou ato municipal, o Prefeito Municipal, para a mesma finalidade (art. 103, § 3º, C.F e art. 124, § 4º, C.E). (Redação dada pelo art. 3º da Resolução nº 01/1999, de 10/02/1999)

II – julgar:

- a) os crimes contra a honra em que forem querelantes as pessoas designadas nas letras "a" e "b" do inciso I, do art. 15, da Lei de Organização Judiciária do Estado, bem como avocar o processo de outros indiciados no caso do art. 85 do Código de Processo Penal;
- b) a suspeição não reconhecida, que se arguiu contra Desembargadores ou o Procurador Geral da Justiça;
- c) os recursos de despacho do Presidente do Tribunal de Justiça e do Relator nos feitos de sua competência;
- d) o recurso previsto no parágrafo único do art. 557 do Código de Processo Penal;
- e) o incidente de arguição de inconstitucionalidade de lei ou ato normativo; [\(Redação dada pelo art. 1º da Resolução nº 21/2016, de 15/09/2016\)](#)
- f) os recursos contra os despachos do Presidente do Tribunal, determinando que se suspenda a execução de medida liminar em mandado de segurança, ou de sentença que a houver concedido;
- g) os embargos de declaração, os embargos infringentes dos seus julgados e os opostos na execução dos seus acórdãos;
- h) os pedidos de arquivamento de inquérito, feitos pelo Procurador Geral da Justiça;
- i) os recursos interpostos pelos interessados contra ato decisório das Comissões Examinadoras de concurso de provas para o cargo de Juiz de Direito Substituto;
- j) os recursos contra as decisões do Conselho da Magistratura;
- l) o agravo regimental de ato do Presidente contra despacho do Relator, nos processos de sua competência;
- m) os pedidos de revogação de medidas de segurança que tiver aplicado;
- n) a perda do cargo de Juiz de Direito, nos casos previstos na Constituição Federal e na Lei Orgânica da Magistratura Nacional;
- o) as reclamações das partes contra embargos opostos pelos Juízes ao uso legítimo do recurso;
- p) a decretação da disponibilidade de Desembargadores e Juízes de Direito, nos casos e pela forma prescritos na Constituição e na Lei Orgânica da Magistratura Nacional;
- q) as dúvidas suscitadas, na execução do Regimento Interno e, bem assim, aquelas que se levantarem sobre a competência das Câmaras;
- r) julgar processo oriundo do Conselho de Justificação ou representação do Ministério Público referente à perda do posto e da patente dos oficiais e da graduação das praças. [\(Incluído pelo art. 1º da Resolução nº 020/2006, de 09/11/2006\)](#)
- s) o incidente de resolução de demandas repetitivas e o incidente de assunção de competência; [\(Incluído pelo art. 1º da Resolução nº 21/2016, de 15/09/2016\)](#)

### III – adotar:

- a) medidas cautelares e de segurança nos feitos de sua competência;
- b) penas disciplinares impondo-as aos Juízes; ou representação para o mesmo fim, ao Conselho Seccional da Ordem dos Advogados e ao Conselho Superior do Ministério Público nos casos de advogados, promotor ou procurador, respectivamente;
- c) a remoção ou a disponibilidade de magistrado, nos termos do art. 45 e seus incisos da Lei Complementar nº 35, de 14 de março de 1979;

### IV – conhecer:

- a) os incidentes de falsidade de documentos ou de insanidade mental de acusados, nos processos de sua competência;

b) o pedido de livramento condicional ou de suspensão condicional de pena, nas condenações que houver proferido;

V – elaborar, por intermédio de comissão eleita, o seu Regimento Interno, interpretá-lo e modificá-lo;

VI – declarar a inconstitucionalidade de lei ou ato do Poder Público.

VII – requerer a intervenção federal no Estado ao Supremo Tribunal Federal, nos termos do art.11, § 1º, alíneas a, b e c da Constituição Federal, para garantir o livre exercício do Poder Judiciário ou para prover a execução de decisão judiciária;

VIII – conceder aposentadoria aos funcionários do Poder Judiciário, de acordo com as leis em vigor.

IX – propor a Assembleia Legislativa alterações da divisão e organização judiciárias sempre que sejam necessárias.

X – organizar os serviços das Secretarias e seus órgãos auxiliares, provendo-lhes os cargos, na forma da lei.

XI – indicar ao Governador do Estado, para nomeação, com fundamento na Lei Orgânica da Magistratura Nacional, de 14 de março de 1979, art. 78, § 3º, os candidatos aprovados nos concursos de Juiz de Direito Substituto, observando-se a ordem classificatória.

XII – efetuar, em sessão reservada e escrutínio secreto as listas para que se removam e promovam Juízes, depois de cumpridas as determinações do art. 81, da Lei Orgânica da Magistratura Nacional.

XIII – compor, em sessão reservada e escrutínio secreto, dependentes de inscrição, as listas tríplexes para acesso, por merecimento, de juízes ao Tribunal de Justiça, bem assim as relações para que sejam providas as vagas reservadas a advogado e membro do Ministério Público.

XIV – eleger, pela maioria dos seus membros efetivos, por votação secreta, dentre seus juízes mais antigos, em número correspondente aos dos cargos de direção, os Titulares destes, com mandato por dois anos, proibida a reeleição; [\(Redação dada pelo art. 1º da Resolução nº 02, de 24/08/1995\)](#)

XV – eleger por maioria de seus membros em sessão reservada e escrutínio secreto, mediante solicitações do Tribunal Regional Eleitoral, os Desembargadores e juízes de direito, que devam integrá-lo, bem como os respectivos suplentes, e indicar, no mesmo caso, as listas tríplexes de juristas e seus substitutos.

XVI – Determinar, pelo voto de dois terços, no mínimo, de seus membros efetivos, a remoção ou a disponibilidade de Juiz de Direito, quando assim exigir o interesse público, e proceder da mesma forma relativamente a seus próprios membros, nos termos do art. 45 e seus incisos da Lei Orgânica da Magistratura Nacional.

XVII – decidir sobre pedido de permuta de juízes de direito;

XVIII – providenciar a aposentadoria compulsória de magistrados ou servidores da Justiça por implemento de idade ou invalidez compulsória.

XIX – licenciar, de ofício, magistrado ou servidor judicial em caso de invalidez ou incapacidade comprovadas.

XX – declarar, nos casos em que ocorrer o abandono ou a perda do cargo de magistrado ou servidor da justiça.

XXI – afastar do exercício do cargo o Juiz de Direito que, submetido a processo

criminal ou administrativo, esteja removido compulsoriamente nos termos do inciso XVI deste artigo.

XXII – decidir as reclamações sobre antiguidade de Juiz de Direito contra a lista respectiva, organizada e publicada de ordem do Presidente do Tribunal.

XXIII – propor, no interesse da Justiça, o aproveitamento de Juiz em disponibilidade;

XXIV – elaborar súmulas de jurisprudência do Tribunal e publicá-las no Diário da Justiça;

XXV – regulamentar os concursos para ingresso na Magistratura, nos termos da Lei (art. 78 da Lei Orgânica da Magistratura Nacional).

XXVI – representar a autoridade competente, quando, em autos ou documentos de que conhecer, houver indícios de crime de ação pública;

XXVII – conceder licença e férias a seu Presidente e demais membros do Tribunal, aos juízes de direito e aos juízes de direito substituto.

XXVIII – homologar a transação das partes, nos feitos pendentes do seu julgamento.

XXIX – aplicar penas disciplinares em acórdãos, a juízes de direito e auxiliares da Justiça, por infração dos deveres do cargo verificada em processo sob o seu julgamento;

XXX – representar ao Governador do Estado quanto à adoção de medidas úteis à boa marcha da administração da Justiça.

XXXI – dar posse ao Governador e ao seu substituto legal, quando não reunida a Assembleia Legislativa.

XXXII – deliberar sobre a proposta orçamentária do Poder Judiciário, a ser encaminhada aos órgãos competentes, e, bem assim, sobre as alterações que se fizerem necessárias durante o exercício.

XXXIII – conceder, a magistrados, o afastamento previsto no art. 73, inciso I, da Lei Complementar nº 35, de 14.03.79, e, bem assim, a autorização a que se refere o art. 35, V, da mesma Lei.

XXXIV – Exercer quaisquer outras atribuições conferidas em lei ou neste Regimento.

## **Seção II - Das Atribuições das Câmaras Reunidas**

### **Subseção I - Disposições gerais**

(Incluída pelo art. 2º Resolução nº 03/1999, de 10/06/1999)

Art. 82. Às Câmaras Reunidas Cíveis e às Câmaras Reunidas Criminais, sem prejuízo dos cometimentos específicos definidos expressamente neste Regimento Interno, ou nele implícitos, compete: (Redação dada pelo art. 2º Resolução nº 03/1999, de 10/06/1999)

I – executar o que for decidido nos feitos de suas respectivas competências;

II – delegar poderes, quando conveniente e oportuno, a juízes de direito e a juízes de direito substituto, para a prática de atos que não envolvam decisão;

III – impor penas disciplinares aos seus funcionários ou representar para idêntico fim ao Procurador Geral da Justiça e à Ordem dos Advogados, consoante se trate de membro do Ministério Público ou advogado;

IV – uniformizar jurisprudência, editando súmulas, quando possível;

V – resolver as dúvidas em matéria de suas competências e das respectivas câmaras especializadas, na forma deste regimento;

VI – declarar extinto o processo, nos casos previstos em lei.

### **Subseção II - Das Câmaras Reunidas Cíveis**

(Incluído pelo art. 2º Resolução nº 03/1999, de 10/06/1999)

Art. 83. Compete especificamente às Câmaras Reunidas Cíveis: (Redação dada pelo art. 2º Resolução nº 03/1999, de 10/06/1999)

I – processar e julgar:

a) os embargos infringentes dos julgados das Câmaras Especializadas e de suas decisões;

b) as ações rescisórias de seus acórdãos, das Câmaras Especializadas e das decisões dos Juízes singulares.

c) a restauração dos autos extraviados ou destruídos, em feitos de sua competência;

d) as habilitações nas causas sujeitas a seu julgamento.

II – julgar:

a) os embargos de declaração opostos a seus acórdãos;

b) o recurso denegatório de embargos infringentes de sua competência;

c) os recursos, quando cabíveis, das decisões do seu Presidente;

d) as suspeições e impedimentos, nos casos em que lhe competirem;

e) os recursos das decisões do Relator, em feitos de sua competência, nos casos previstos neste regimento.

### **Subseção III - Das Câmaras Reunidas Criminais**

(Incluído pelo art. 2º Resolução nº 03/1999, de 10/06/1999)

Art. 84. Compete especificamente às Câmaras Reunidas Criminais: (Redação dada pelo art. 2º Resolução nº 03/1999, de 10/06/1999)

I – processar e julgar:

a) os recursos das decisões do seu Presidente, na forma deste regimento;

b) os pedidos de desaforamento;

c) os conflitos de competência entre as Câmaras e o Conselho de Justiça Militar do Estado;

II – julgar:

a) os embargos de declaração opostos aos seus acórdãos;

b) os recursos de decisão do Relator, quando este indeferir, liminarmente, a interposição de embargos infringentes;

c) as suspeições e impedimentos, nos feitos de sua competência, dos membros das Câmaras e do Procurador Geral de Justiça;

d) os pedidos de *habeas corpus*, nos feitos submetidos ao seu julgamento, concedendo-os de ofício nos casos previstos em lei.

III – executar o que for decidido nos feitos de sua competência.

### **Seção III - Das Atribuições das Câmaras Especializadas Cíveis**

Art. 85. Compete às Câmaras Especializadas Cíveis: (Redação originária renumerado por força do art. 3º da Resolução nº 03/1999, de 10/06/1999)

I – julgar os recursos das sentenças e decisões dos juízes do cível e do juízo arbitral, ressalvadas a competência do Tribunal Pleno e das Câmaras Cíveis Reunidas, e os embargos declaratórios opostos a seus acórdãos.

II – promover a restauração de autos, nos feitos de sua competência.

III – exercer, no que lhe for aplicável, as atribuições conferidas ao Tribunal Pleno e às Câmaras Reunidas, e, bem assim, desempenhar atribuições outras que lhe sejam cometidas por lei prevista neste Regimento.

### **Seção IV - Das Atribuições das Câmaras Criminais**

(Redação dada pelo art. 4º da Resolução nº 01/1999, de 10/02/1999)

Art. 86. Compete às Câmaras Criminais: (Artigo correspondente ao art. 84 com a Redação dada pelo art. 4º da Resolução nº 01/1999, de 10/02/1999, renumerado por força do art. 3º da Resolução nº 03/1999, de 10/06/1999)

I – processar e julgar originariamente, nos crimes comuns e de responsabilidade os prefeitos, vice-prefeitos e vereadores;

II – os secretários municipais ou ocupantes de cargos equivalentes, nos crimes de responsabilidade, quando conexos com os do Prefeito;

III – julgar, como instância de segundo grau, os recursos das sentenças e decisões dos juízes criminais da auditoria militar, do Tribunal do Júri e de tribunais especiais;

IV – julgar as reclamações contra aplicação das penalidades previstas nos arts. 801 e 802, do Código de Processo Penal; e os *habeas corpus* que fugirem à competência do Tribunal Pleno;

V – ordenar o exame a que se refere o art. 777, do Código de Processo Penal;

VI – reexaminar a decisão definitiva proferida em processos de menores de dezoito anos;

VII – executar, no que couber, as suas decisões;

VIII – promover a restauração de autos relativos a feitos submetidos ao seu julgamento;

IX – exercer, no que lhe for aplicável, as atribuições conferidas ao Tribunal Pleno e Câmaras Reunidas e, bem assim, desempenhar atribuições outras previstas em lei e neste Regimento;

X – (Revogado) (Redação dada pela Resolução nº 4/2008, de 27/03/2008).

### **Seção V - Das Atribuições do Presidente do Tribunal**

(Embora já tivesse essa mesma denominação na redação original, a designação “Das atribuições do Presidente do Tribunal” para esta Seção V foi repetida pelo art. 4º da Resolução nº 03/1999, de 10/06/1999)

Art. 87. Sem prejuízo de outras atribuições expressas ou implícitas neste regimento, ao Presidente do Tribunal compete: (Redação dada pelo art. 4º da Resolução nº 03/1999, de 10/06/1999)

- I – dirigir os trabalhos do Colegiado e presidir-lhe as sessões plenárias, fazendo cumprir este Regimento;
- II – promover o cumprimento imediato das decisões do Tribunal;
- III – corresponder-se com as autoridades públicas sobre assuntos relacionados com a administração da Justiça;
- IV – representar o Tribunal nas solenidades e atos oficiais podendo, quando conveniente, delegar a incumbência a um ou mais Desembargadores;
- V – presidir o Conselho da Magistratura;
- VI – dar posse aos Desembargadores, Juízes e aos Servidores do Poder Judiciário;
- VII – convocar, na hipótese de falta ou impedimento de Desembargadores, os respectivos substitutos dentre os juízes da Capital, mediante sorteio público.
- VIII – conhecer do pedido de recurso extraordinário e de recurso especial, e se o julgar relativamente amparado, mandar processá-lo, resolvendo os incidentes suscitados.
- IX – funcionar como Relator privativo, com direito a voto, nos seguintes feitos:
- a) *habeas corpus* de julgamento da competência originária do Tribunal Pleno;
  - b) suspeição de Desembargador, inclusive no caso do art. 135, do Código de Processo Civil;
  - c) reclamação sobre antiguidade dos magistrados, apurada pelo Conselho da Magistratura;
  - d) os conflitos de competência entre as Câmaras Especializadas ou entre as Câmaras Reunidas e o Tribunal Pleno;
  - e) remoção, disponibilidade, aposentadoria compulsória de magistrados, serventuários e funcionários do Poder Judiciário;
  - f) reversão ou aproveitamento de magistrados e demais servidores referidos na letra anterior;
  - g) nos pedidos de licença e férias dos magistrados;
- X – conceder prorrogação de prazo para que magistrados e demais servidores da Justiça tomem posse e entrem em exercício dos cargos;
- XI – ordenar a suspensão de liminar e a execução da sentença concessiva de mandado de segurança no juízo a quo (art. 4º da Lei nº 4.348, de 26.06.64);  
(No texto publicado, por equívoco, foi feito referência ao “art. 4º, da Lei 4.384, de 26.06.54”)
- XII – assinar os acórdãos do Tribunal Pleno e do Conselho da Magistratura com os Juízes-Relatores e com os que expressamente tenham requerido declaração de voto;
- XIII – expedir ordens que não dependam de acórdãos ou não sejam da privativa competência dos Relatores.
- XIV – ordenar o pagamento dos precatórios em virtude de sentença proferida contra a Fazenda Pública, estadual ou municipal, nos termos do art. 100 da Constituição do Brasil e dos arts. 730, inciso I e 731, do Código de Processo Civil;
- XV – determinar a restauração dos feitos perdidos nas Secretarias do Tribunal;
- XVI – julgar os recursos das decisões que incluam jurados na lista geral ou dela excluam;
- XVII – conceder licença para casamento nos casos do art. 183, XVI, do Código Civil;
- XVIII – encaminhar ao Governador do Estado, depois de aprovados pelo Tribunal, os pedidos de permuta de Juiz.



- XIX – comunicar à Ordem dos Advogados as faltas cometidas por advogados, provisionados e estagiários;
- XX – conhecer e julgar as suspeições opostas aos serventuários e demais Funcionários do Poder Judiciário;
- XXI – nomear, demitir, exonerar, admitir, dispensar, transferir e aposentar os funcionários do Poder Judiciário, inclusive preenchimento de função gratificada;
- XXII – mandar anunciar a abertura de vaga de Juiz de Direito, bem como a abertura de concurso para Juizes de Direito Substituto, funcionários, servidores e serventuários da Justiça;
- XXIII – encaminhar ao Juiz competente as cartas rogatórias, bem assim a carta de sentença estrangeira, homologada pelo Supremo Tribunal Federal, para o devido cumprimento;
- XXIV – superintender os serviços das Secretarias do Tribunal e fiscalizar o andamento e a regularidade de seus trabalhos;
- XXV – abrir, rubricar e encerrar os livros destinados ao serviço do Tribunal;
- XXVI – vedar o acesso ao recinto das sessões e às Secretarias a pessoas reconhecidas como intermediárias de negócios ilícitos ou reprováveis ou que, pela sua conduta, possam comprometer o decoro da Justiça;
- XXVII – apostilar os títulos dos Desembargadores, Juizes e funcionários do Tribunal, ainda que em disponibilidade ou aposentados, com referência a acréscimo de vencimentos ou vantagens e alterações de situação funcional, e conceder, a esses servidores, salário-família e gratificações adicionais e, bem assim apostilar os títulos dos pensionistas de magistrados falecidos, cujos cálculos, das respectivas pensões, sejam feitos pelo setor competente do Tribunal;
- XXVIII – substituir o Governador do Estado, nos casos previstos na Constituição;
- XXIX – conhecer das reclamações contra a exigência de custas indevidas ou excessivas, ordenando as restituições e impondo penalidades cabíveis, providências que poderão ser tomadas independentemente de reclamação, sempre que tais ocorrências constarem dos autos ou papéis que lhe forem presentes;
- XXX – prestar as informações solicitadas por outros Tribunais;
- XXXI – processar e julgar pedidos de concessão de Justiça gratuita, quando o feito não estiver ainda distribuído, ou depois de cessadas as atribuições do Relator;
- XXXII – exercer qualquer outra atribuição mencionada em lei ou prevista no Regimento Interno.

## **Seção VI - Das Atribuições do Vice-Presidente**

Art. 88. Ao Vice-Presidente do Tribunal compete: ([Artigo correspondente ao art. 86 da redação originária, renumerado por força do art. 3º da Resolução nº 03/1999, de 10/06/1999](#))

- I – presidir as sessões de uma das Câmaras Reunidas e da Câmara Especializada a que integrar; ([Redação dada pelo art. 5º da Resolução nº 01/1999, de 10/02/1999](#))
- II – assinar os acordos com o Relator e os Juizes que requeiram declaração de voto;
- III – ([Revogado](#)) ([Redação dada pelo art. 3º da Resolução nº 01/1996, de 07/03/1996](#))
- IV – substituir o Presidente nas faltas, férias, licenças e impedimentos;

V – integrar o Conselho da Magistratura;

VI – exercer qualquer outra atribuição conferida em lei ou no Regimento Interno.

Art. 88-A. Se ocorrer vacância dos Órgãos de Direção, será o Plenário convocado para o necessário provimento. [\(Incluído pelo art. 1º da Resolução nº 014/2005, de 20/10/2005\)](#)

Parágrafo único. Para manter a coincidência, da data da posse dos dirigentes do Tribunal de Justiça, o eleito permanecerá no cargo até o término do mandato dos atuais dirigentes, sem se tornar inelegível para o período subsequente, salvo se o lapso temporal de substituição for superior a um ano. [\(Incluído pelo art. 1º da Resolução nº 014/2005, de 20/10/2005\)](#)

### **Seção VII - Das Atribuições dos Presidentes das Câmaras Reunidas**

Art. 89. Aos Presidentes das Câmaras Reunidas compete: [\(Artigo correspondente ao art. 87 da redação originária, renumerado por força do art. 3º da Resolução nº 03/1999, de 10/06/1999, publicada no DJE nº 4.059, de 11/06/1999, pp. 1/11\)](#)

I – dirigir e manter a regularidade dos trabalhos e a polícia das sessões pela forma determinada neste Regimento;

II – sustar a decisão de qualquer processo, remetendo este ao Presidente do Tribunal, para que seja julgado pelo Plenário, quando da competência do Tribunal Pleno;

III – marcar dia para julgamento dos feitos e organizar a pauta da sessão imediata;

IV – exigir dos funcionários das Secretarias o cumprimento dos atos necessários ao regular funcionamento das sessões e execução de suas determinações, sem ofensa das prerrogativas do Presidente;

V – providenciar para a organização e publicação trimestral do ementário dos acórdãos e da estatística dos julgamentos das Câmaras Reunidas;

VI – ordenar a exclusão, do recinto de julgamento, de advogado ou pessoas outras que faltarem ao devido decoro;

Parágrafo único. As Câmaras Reunidas Cíveis ou as Criminais serão presididas, uma pelo Vice-Presidente do Tribunal, conforme disposto no art. 88, I, e a outra pelo Presidente da Primeira Câmara Especializada Cível ou Criminal não presidida por aquele. [\(Parágrafo único correspondente ao art. 88 da redação originária, transformado em parágrafo único do art. 89 por força do art. 5º da Resolução nº 03/1999, de 10/06/1999, que também lhe deu nova redação\)](#)

### **Seção VIII - Das Atribuições dos Presidentes das Câmaras Especializadas**

Art. 90. Aos presidentes das Câmaras Especializadas compete: [\(Artigo correspondente ao art. 89 da redação originária, renumerado por força do art. 3º da Resolução nº 03/1999, de 10/06/1999\)](#)

I – dirigir e manter a regularidade dos trabalhos e a ordem das sessões, pela forma determinada neste Regimento.

II – sustar decisão em que juiz concluir pela inconstitucionalidade de lei ou de ato do poder público, encaminhando o processo ao Presidente do Tribunal de Justiça para julgamento pelo Colegiado;

III – redigir os resumos de julgamento e assinar os acórdãos com os relatores e com os

juízes que tenham feito declaração de voto;

IV – marcar dia para julgamento das causas e organizar a pauta das sessões;

V – exigir dos funcionários das secretarias o cumprimento dos atos necessários ao regular funcionamento das sessões e a execução de suas determinações, sem ofensa das prerrogativas do Presidente;

VI – providenciar para a organização e publicação trimestral do ementário dos acórdãos e da estatística dos julgamentos da Câmara;

VII – ordenar a exclusão, do recinto de julgamento, de advogado ou pessoas outras que faltarem ao devido decoro.

Parágrafo único. As Câmaras Especializadas Cíveis, e as Criminais, serão presididas, uma pelo Vice-Presidente do Tribunal, na forma do art. 88, I, e as demais por Desembargadores eleitos dentre seus membros. (Parágrafo único correspondente ao art. 90 da redação originária, transformado em parágrafo único do art. 89 por força do art. 5º da Resolução nº 03/1999, de 10/06/1999, que também lhe deu nova redação)

### **Seção IX - Das Atribuições dos Relatores**

Art. 91. Compete ao Relator, nos feitos que lhe forem distribuídos, além de outros deveres legais e deste Regimento:

I – processar os feitos e relatá-los;

II – resolver os incidentes relativos à ordem e regularidade do processo, quando independam de acórdão, e executar as diligências necessárias ao julgamento;

III – fazer cumprir as decisões administrativas de sua competência;

IV – lavrar o acórdão, quando não for voto vencido e assiná-lo juntamente com o Desembargador que houver presidido a decisão;

V – proceder ao interrogatório do acusado, reinquirir testemunhas ou determinar outras diligências, na hipótese do art. 616 do Código de Processo Penal;

VI - não conhecer de recurso inadmissível, prejudicado ou que não tenha impugnado especificamente os fundamentos da decisão recorrida; (Redação dada pelo art. 1º da Resolução nº 21/2016, de 15/09/2016)

VI-A - negar provimento a recurso que for contrário a súmula ou acórdão proferido pelo Supremo Tribunal Federal ou pelo Superior Tribunal de Justiça em julgamento de recursos repetitivos; (Redação dada pelo art. 1º da Resolução nº 21/2016, de 15/09/2016)

VI-B - negar provimento a recurso que for contrário a súmula deste Tribunal ou entendimento firmado em incidente de resolução de demandas repetitivas ou de assunção de competência; (Incluído pelo art. 1º da Resolução nº 21/2016, de 15/09/2016)

VI-C - depois de facultada a apresentação de contrarrazões, dar provimento ao recurso se a decisão recorrida for contrária a súmula ou acórdão proferido pelo Supremo Tribunal Federal ou pelo Superior Tribunal de Justiça em julgamento de recursos repetitivos; (Incluído pelo art. 1º da Resolução nº 21/2016, de 15/09/2016)

VI-D - depois de facultada a apresentação de contrarrazões, dar provimento ao recurso se a decisão recorrida for contrária a súmula deste Tribunal ou entendimento firmado em incidente de resolução de demandas repetitivas ou de assunção de competência;

(Incluído pelo art. 1º da Resolução nº 21/2016, de 15/09/2016)

VII – assinar as cartas ou títulos executivos de sentença;

VIII – expedir alvará de soltura, dando imediato conhecimento ao Juiz de primeira instância no caso de decisão absolutória ou proferida em grau de recurso;

IX – denegar ou decretar prisão preventiva nos processos criminais;

X – conceder ou recusar fiança nos processos-crime;

XI – apresentar em mesa para julgamento os feitos que independam de pauta; (Redação dada pelo art. 6º da Resolução nº 03/1999, de 10/06/1999)

XII – lançar da acusação o queixoso que deixar de comparecer (art. 516, inciso II do Código de Processo Penal) nos crimes de competência originária do Tribunal;

XIII – processar as habilitações requeridas e outros incidentes;

XIV – homologar por despacho o pedido de desistência dos recursos que lhe sejam distribuídos;

XV – homologar desistência nas ações rescisórias;

XVI – promover as diligências e atos que não dependam de julgamento, nos feitos que lhe sejam distribuídos;

XVII – decidir os pedidos originários de benefícios de justiça gratuita, nos feitos que lhe couberem por distribuição;

XVIII – encaminhar os pedidos de mandado de segurança à autoridade legítima para julgamento, quando for incompetente o Tribunal de Justiça, nos termos da legislação processual civil;

XIX – negar, liminarmente, os pedidos de revisão criminal, quando se verificar a incompetência do Tribunal ou de Câmara Criminal, ou não estiver instruído o processo e for desaconselhável aos interesses da Justiça que se apensem aos autos originais;

XX – lavrar, em forma de acórdão, as decisões tomadas nos processos administrativos;

XXI – requisitar os autos originais, quando necessário;

XXII – delegar atribuições a outras autoridades judiciárias, nos casos previstos em lei e neste Regimento;

XXIII – mandar riscar injúrias escritas, em autos, pelos advogados;

XXIV -requerer a instauração do incidente de resolução de demandas repetitivas ou do incidente de assunção de competência; (Redação dada pelo art. 1º da Resolução nº 21/2016)

XXV – fazer as ementas dos acórdãos que lavrar;

XXVI – denegar ou conceder a ordem de mandado de segurança, desde que a matéria versada no *writ* em questão constitua objeto de jurisprudência consolidada no tribunal.

(Incluído pelo art. 1º da Resolução nº 28/2014, 27/11/14)

§ 1º O disposto no inciso VI não se aplica ao recurso extraordinário e ao recurso especial. (Incluído pelo art. 6º da Resolução nº 03/1999, de 10/06/1999)

§ 2º Ao pedir dia para julgamento, ou apresentar o feito em mesa, indicará o relator, nos autos, se o submete ao Plenário ou à Câmara, salvo se pela simples designação da classe estiver fixado o órgão competente. (Incluído pelo art. 6º da Resolução nº 03/1999, de 10/06/1999)

§ 3º. Nos casos do inciso II deste artigo, constatada a ocorrência de vício sanável, inclusive aquele que possa ser conhecido de ofício, o relator determinará a realização ou

a renovação do ato processual, no próprio Tribunal ou em primeiro grau de jurisdição, intimadas as partes. (Redação dada pelo art. 2º da Resolução nº 06/2016, de 04/04/2016)  
§ 4º. Cumprida a diligência de que trata o § 3º, o relator, sempre que possível, prosseguirá no julgamento do recurso. (Redação dada pelo art. 2º da Resolução nº 06/2016, de 04/04/2016)

§ 5º. Reconhecida a necessidade de produção de prova, o relator converterá o julgamento em diligência, que se realizará no Tribunal ou em primeiro grau de jurisdição, decidindo-se o recurso após a conclusão da instrução. (Redação dada pelo art. 2º da Resolução nº 06/2016, de 04/04/2016)

### **Seção X - Das Atribuições dos Revisores**

Art. 92. (Revogado) (Redação dada pelo art. 85 da Resolução nº 06/2016, de 04/04/2016)

### **Seção XI - Da Representação para a Perda do Posto e Patente e da Graduação**

Art. 92-A. (Revogado) (Redação dada pelo art. 1º da Resolução nº 4/2008, de 27/03/2008).

Art. 92-B. (Revogado) (Redação dada pelo art. 1º da Resolução nº 4/2008, de 27/03/2008).

Art. 92-C. (Revogado) (Redação dada pelo art. 1º da Resolução nº 4/2008, de 27/03/2008).

Art. 92-D. (Revogado) (Redação dada pelo art. 1º da Resolução nº 4/2008, de 27/03/2008).

## **CAPÍTULO V - DO CONSELHO DA MAGISTRATURA**

Art. 93. O Conselho da Magistratura, órgão disciplinar, composto de três membros, o Presidente, o Vice-Presidente do Tribunal e o Corregedor da Justiça, tem como órgão superior o Tribunal Pleno.

Art. 94. O membro do Conselho da Magistratura, nas suas faltas, impedimentos ou suspeição, será substituído pelo Desembargador, desimpedido, que lhe seguir na ordem decrescente de antiguidade.

Art. 95. As atribuições do Conselho da Magistratura são definidas no respectivo Regimento Interno, o qual regula o seu funcionamento.

## **CAPÍTULO VI - DA CORREGEDORIA GERAL DA JUSTIÇA**

### **Seção I - Disposições Gerais**

Art. 96. A Corregedoria Geral da Justiça, que funciona na sede do Tribunal, órgão de fiscalização, orientação, controle e instrução dos serviços forenses e administrativos da justiça de primeiro grau, tem competência em todo o Estado e é exercido por Desembargador eleito

por dois anos, juntamente com os demais titulares de cargos de direção do Poder Judiciário, na forma da lei.

## **Seção II - Das Atribuições do Corregedor Geral**

(A seção II consta do Regimento Interno editado pelo Tribunal, mas não no texto publicado no Diário de Justiça)

Art. 97. Compete ao Corregedor Geral da Justiça as atribuições especificadas na Seção VIII, da Lei de Organização Judiciária do Estado e, bem assim, as que lhe forem cometidas por outros diplomas legais constantes do seu Regimento Interno.

## **CAPÍTULO VII - DA POLÍCIA DO TRIBUNAL**

Art. 98. Cabe ao Tribunal de Justiça o poder de polícia no recinto e nas dependências do prédio em que funciona e em que tem a respectiva sede.

Art. 99. No exercício da atribuição a que se refere o artigo anterior, poderá ser requisitado o auxílio de outras autoridades, quando necessário.

Art. 100. Ocorrendo infração à lei penal na sede ou dependência do Tribunal, o Presidente instaurará inquérito, se envolver autoridade ou pessoa sujeita à sua jurisdição, ou delegará esta atribuição a outro Desembargador.

§ 1º Nos demais casos, o Presidente poderá proceder na forma deste artigo ou requisitar a instauração de inquérito à autoridade competente.

§ 2º O Desembargador incumbido do inquérito designará escrivão um dos servidores do Tribunal.

Art. 101. A polícia das sessões e das audiências compete ao respectivo Presidente.

Art. 102. Os inquéritos administrativos serão realizados consoante as normas próprias.

## **CAPÍTULO VIII - DA REPRESENTAÇÃO POR DESACATO**

Art. 103. Sempre que tiver conhecimento de desobediência a ordem emanada do Tribunal ou de seus Desembargadores no exercício da função, ou de desacato ao Tribunal, ou a seus Desembargadores, o Presidente comunicará o fato ao órgão competente do Ministério Público, provendo-o dos elementos de que dispuser a propositura da ação penal.

Art. 104. Decorrido o prazo de trinta dias, sem que tenha sido instaurada ação penal, o Presidente dará ciência ao Tribunal, em sessão, para as providências que julgar necessárias.

## **PARTE II - DAS ATIVIDADES JURISDICIONAIS E ADMINISTRATIVAS**

### **CAPÍTULO IX - DOS ATOS E FORMALIDADES**

#### **Seção I - Disposições Gerais**

Art. 105. A atividade jurisdicional no Tribunal de Justiça será ininterrupta, sendo vedado

férias coletivas nos juízos e tribunais de 2º grau, funcionando, nos dias em que não houver expediente forense normal, juízes em plantão permanente.

Art. 106. (Revogado) (Redação dada pelo art. 5º da Resolução nº 13, de 29/09/2005)

Art. 107. (Revogado) (Redação dada pelo art. 5º da Resolução nº 13, de 29/09/2005)

Art. 108. (Revogado) (Redação dada pelo art. 5º da Resolução nº 13, de 29/09/2005)

Art. 109. Suspendem-se os trabalhos do Tribunal nos sábados, domingos, feriados e nos dias em que o Tribunal o determinar, observado o disposto no art. 105, deste Regimento Interno. (Redação dada pelo art. 2º da Resolução nº 013, de 29/09/2005)

Art. 110. Os atos processuais serão autenticados, conforme o caso, mediante a assinatura ou a rubrica do Presidente, dos Desembargadores ou dos servidores para tal fim qualificados.

§ 1º É exigida a assinatura usual nos acórdãos, na correspondência oficial, no fecho das cartas de sentença e nas certidões.

§ 2º Os livros necessários ao expediente serão rubricados pelo presidente ou por funcionário designado.

§ 3º As rubricas e assinaturas usuais dos servidores serão registradas em livro próprio, para identificação do signatário.

Art. 111. As peças que devam integrar ato ordinatório ou executório poderão ser-lhe anexadas em cópia autenticada.

Art. 112. As intimações efetuam-se de ofício, em processos pendentes, e consideram-se feitas pela só publicação dos atos no órgão oficial, sendo, todavia, a intimação do Ministério Público feita pessoalmente.

Parágrafo único. Da publicação do expediente de cada processo constarão os nomes das partes e de seus advogados, suficientes para sua identificação.

Art. 113. A retificação de publicação no Diário da Justiça, com efeito de intimação, decorrente de incorreções ou omissões, será providenciada pela Secretaria, de ofício ou mediante despacho do Presidente do Tribunal, das Câmaras Reunidas, das Câmaras Especializadas Cíveis ou Criminais, ou do Relator.

Art. 114. A publicação da pauta deverá ser feita no prazo de, pelo menos, 05 (cinco) dias úteis antes da sessão de julgamento, ressalvados os processos criminais, cujo prazo será de 48 (quarenta e oito) horas. (Redação pelo art. 1º dada pela Resolução nº 35/2016), de 29/09/2016)

§ 1º. Na contagem desse prazo, computar-se-ão somente os dias úteis. (Redação dada pelo art. 3º da Resolução nº 06/2016, de 04/04/2016)

§ 2º. O cômputo desse prazo se iniciará regressivamente a partir do primeiro dia útil anterior à data da sessão e, terminando em dia não útil, retrocederá a contagem do prazo para julgamento para o primeiro dia útil anterior. (Redação dada pelo art. 3º da Resolução nº 06/2016, de 04/04/2016)

§ 3º. À contagem desse prazo, não se aplicam as regras do art. 180 e 183 da Lei n. 13.105, de 16 de março de 2015 (Código de Processo Civil). (Redação dada pelo art. 3º da Resolução nº 06/2016, de 04/04/2016)

§ 4º. É obrigatória a inclusão na pauta de julgamento de todos os recursos e demais processos de competência originária, ressalvados os casos de *habeas corpus* e dos embargos de declaração julgados na primeira sessão subsequente à sua oposição, assim como os processos cujos julgamentos tiverem sido expressamente adiados para a sessão

seguinte. (Redação dada pelo art. 3º da Resolução nº 06/2016, de 04/04/2016)

Art. 115. Qualquer matéria de natureza administrativa, sobre a qual tenha de deliberar o Tribunal, deverá ser cientificada aos Desembargadores com antecedência mínima de quarenta e oito horas, ressalvadas as hipóteses de manifesta e evidente urgência que não permitam tal anterioridade.

Parágrafo único. Presumem-se feitas as cientificações pela entrega da respectiva “pauta” nas residências ou endereços indicados pelos Desembargadores, os quais, todavia, poderão renunciar ao prazo de anterioridade.

Art. 116. A pauta de julgamento será afixada na entrada da sala em que se realizar a sessão de julgamento.

Art. 117. As partes poderão ter acesso aos autos em secretaria após a publicação da pauta de julgamento. (Redação dada pelo art. 4º da Resolução nº 06/2016, de 04/04/2016)

Parágrafo único. Se os autos estiverem conclusos ao relator ou a qualquer outro integrante do Tribunal, deverá ser garantido acesso imediato aos autos do processo ao procurador da parte em gabinete. (Redação dada pelo art. 4º da Resolução nº 06/2016, de 04/04/2016)

Art. 118. Aos Desembargadores que hajam de participar do julgamento será distribuída antecipadamente, cópia do relatório, nos embargos infringentes, na ação rescisória e em casos outros, quando a lei o determinar.

Parágrafo único. Nas arguições de inconstitucionalidade de lei ou de ato normativo do poder público e nos casos de pronunciamento prévio do Tribunal acerca de interpretação do direito (CPC, art. 476), a Secretaria distribuirá a todos os Desembargadores cópia do acórdão que, nas Câmaras Reunidas Cíveis ou nas Câmaras Especializadas Cíveis, houver acolhido a alegação de inconstitucionalidade ou reconhecido a divergência de interpretação.

## **Seção II - Do Registro e da Classificação dos Feitos**

Art. 119. Os autos, as petições e os documentos remetidos ou entregues ao Tribunal de Justiça serão registrados em fichas, no protocolo, no dia de sua entrega, cabendo à Secretaria para Assuntos Cartorários verificar-lhes a numeração das folhas e ordená-los para distribuição.

Art. 120. Além do registro nas fichas do protocolo, será procedido, no dia imediato, o registro no tomo geral e nos livros destinados a cada categoria de feitos.

Art. 121. O registro será feito em numeração contínua no tomo geral e seriada em cada uma das classes seguintes:

I – ação penal (originária);

II – ação rescisória;

III – agravo;

IV – apelação cível e exame obrigatório da segunda Instância;

V – apelação criminal;

VI – arguição de incompetência, impedimento ou suspeição;

VII – carta de ordem ou carta rogatória;

VIII – comunicação e petição;



IX – conflito de jurisdição ou de competência e de atribuições;  
X – desaforamento;  
XI – embargos infringentes;  
XII – *habeas corpus*;  
XIII – inquérito;  
XIV – mandado de segurança;  
XV – processo ou recurso administrativo;  
XVI – reclamação ou representação;  
XVII – recurso criminal;  
XVIII – suspensão de segurança;  
XIX – uniformização de jurisprudência;  
XX – revisão criminal;  
XXI – carta testemunhal;  
XXII – declaração de inconstitucionalidade.

Art. 122. Não se altera a classe do processo:

- a) pela interposição de embargos ou de agravo regimental;
- b) pela reclamação por erro de ata;
- c) pelos pedidos incidentes ou acessórios;
- d) pelos pedidos de execução, salvo nos casos de intervenção.

Art. 123. Será feita na autuação nota distintiva do recurso ou do incidente, quando este não alterar a classe e o número do processo.

Art. 124. A restauração dos autos perdidos terá a numeração destes e será distribuída a um Relator, observados os arts. 338 e 339.

### **Seção III - Do Preparo**

Art. 125. Sem o respectivo preparo, exceto em caso de isenção legal, nenhum processo será distribuído, nem serão praticados nele atos processuais, salvo os que forem ordenados de ofício pelo Relator, pelo Plenário, pelas Câmaras Reunidas, pelas Câmaras Especializadas, ou pelos respectivos Presidentes.

Art. 126. O preparo compreende todos os atos do processo, inclusive a baixa dos autos, se for o caso, mas não dispensa o pagamento das despesas de remessa e de retorno, salvo os casos de remessa e de retorno no processo de autos eletrônicos, nos quais estas últimas parcelas não serão devidas. ([Redação dada pelo art. 5º da Resolução nº 06/2016, de 04/04/2016](#))

Art. 127. Quando o autor e o réu recorrerem, cada recurso estará sujeito a preparo integral.

§ 1º Tratando-se de litisconsortes necessários, bastará que um dos recursos seja preparado, para que todos sejam julgados, ainda que não coincidam suas pretensões.

§ 2º O disposto no parágrafo anterior é extensivo ao assistente.

§ 3º O terceiro prejudicado que recorrer fará o preparo do seu recurso, independentemente do preparo dos recursos que, porventura, tenham sido interpostos pelo autor ou pelo réu.

Art. 128. O preparo será feito no prazo previsto na lei processual, mediante guia à repartição ou estabelecimento bancário competente, juntando-se aos autos o

comprovante.

§ 1º. No ato de interposição do recurso, o recorrente comprovará, quando exigido pela legislação pertinente, o respectivo preparo, inclusive porte de remessa e de retorno, sob pena de deserção. (Redação dada pelo art. 6º da Resolução nº 06/2016, de 04/04/2016)

§ 2º. A insuficiência no valor do preparo, inclusive porte de remessa e de retorno, implicará deserção se o recorrente, intimado na pessoa de seu advogado, não vier a supri-lo no prazo de 5 (cinco) dias. (Redação dada pelo art. 6º da Resolução nº 06/2016, de 04/04/2016)

§ 3º. O recorrente que não comprovar, no ato de interposição do recurso, o recolhimento do preparo, inclusive porte de remessa e de retorno, será intimado, na pessoa de seu advogado, para realizar o recolhimento em dobro, sob pena de deserção. (Redação dada pelo art. 6º da Resolução nº 06/2016, de 04/04/2016)

§ 4º. É vedada a complementação se houver insuficiência parcial do preparo, inclusive porte de remessa e de retorno, no recolhimento realizado na forma do § 3º. (Redação dada pelo art. 6º da Resolução nº 06/2016, de 04/04/2016)

§ 5º. Provando o recorrente justo impedimento, o relator relevará a pena de deserção, por decisão irrecorrível, fixando o prazo de 5 (cinco) dias para efetuar o preparo. (Redação dada pelo art. 6º da Resolução nº 06/2016, de 04/04/2016)

§ 6º. O equívoco no preenchimento da guia de custas não implicará a aplicação da pena de deserção, cabendo ao relator, na hipótese de dúvida quanto ao recolhimento, intimar o recorrente para sanar o vício no prazo de 5 (cinco) dias. (Redação dada pelo art. 6º da Resolução nº 06/2016, de 04/04/2016)

Art. 129. Cabe às partes prover o pagamento antecipado das despesas dos atos que realizem ou requeiram no processo, ficando o vencido, afinal, responsável pelas custas e despesas pagas pelo vencedor:

Art. 130. Haverá isenção do preparo:

I – nos conflitos de jurisdição ou competência, nos conflitos de atribuições, nos *habeas-corpus* e nos demais processos criminais, salvo a ação penal privada;

II – nos procedimentos instaurados e nos pedidos e recursos formulados ou interpostos pelo Ministério Público, pela Fazenda Pública em geral ou por beneficiário de assistência judiciária.

Art. 131. A assistência judiciária, perante o Tribunal, será requerida ao Presidente, antes da distribuição, e, nos demais casos, ao Relator.

Art. 132. Sem prejuízo da nomeação, quando couber, de defensor ou curador dativo, o pedido de assistência judiciária será deferido ou não, de acordo com a legislação em vigor.

Parágrafo único. Prevalecerá no Tribunal a assistência judiciária já concedida em outra instância.

Art. 133. O pagamento dos preços cobrados pelo fornecimento de cópias, autenticadas ou não, ou de certidões por fotocópia ou por meio equivalente será antecipado ou garantido com depósito na Secretaria, consoante tabela aprovada pelo Presidente.

Art. 134. A deserção do recurso por falta de preparo será declarada:

I – pelo Presidente, antes da distribuição;

II – pelo Relator;

III – pelo Plenário, pelas Câmaras Reunidas ou pelas Câmaras Especializadas, ao

conhecerem do feito.

#### Seção IV - Da Distribuição

Art. 135. Verificado o preparo, sua isenção ou dispensa, serão os autos distribuídos diariamente, ao final do expediente, em audiência pública, com emissão do respectivo relatório. (Redação dada pelo art. 1º da Resolução nº 01, de 07/03/1996)

Art. 135-A. Far-se-á a distribuição de acordo com este Regimento Interno, observando-se a alternatividade, o sorteio eletrônico e a publicidade. (Redação dada pelo art. 7º da Resolução nº 06/2016, de 04/04/2016)

Parágrafo único. O primeiro recurso protocolado no Tribunal tornará prevento o relator para eventual recurso subsequente interposto no mesmo processo ou em processo conexo, ainda que aquele recurso já tenha sido julgado quando da interposição do segundo. (Redação dada pelo art. 7º da Resolução nº 06/2016, de 04/04/2016)

Art. 136. (Revogado) (Redação dada pelo art. 3º da Resolução nº 01, de 07/03/1996)

Art. 137. A distribuição disciplinada nesta seção, far-se-á pelo sistema de computação eletrônica, mantidos o equilíbrio de pesos em cada classe de processo, ressalvadas as exceções previstas neste regimento. (Redações dadas pelo art. 1º da Resolução nº 01, de 07/03/1996)

Parágrafo único. Ocorrendo a impossibilidade de realização da distribuição pelo sistema eletrônico ficará a critério do Presidente realizá-la mediante sorteio. (Redação dada pelo art. 1º da Resolução nº 01, de 07/03/1996)

Art. 138. Não estão sujeitos à distribuição os pedidos originários de *habeas-corpus* da competência do Tribunal Pleno e as arguições de suspeição ou impedimento de Desembargador, que serão relatados pelo Presidente.

Parágrafo único. Nos demais casos de *habeas corpus* haverá imediata distribuição aos membros de Câmara Criminal.

Art. 139. Far-se-á a distribuição entre todos os Desembargadores competentes em razão da matéria, com exceção do Presidente do Tribunal de Justiça, do Corregedor Geral da Justiça e dos Desembargadores afastados, a qualquer título, por período superior a trinta dias, ou em gozo de férias, operando-se, nos dois últimos casos, compensação posterior. (Redação dada pelo art. 1º da Resolução nº 42, de 24/11/2011)

§1º Não será distribuído a desembargador afastado por período igual ou inferior a trinta dias, compensando-se posteriormente a distribuição, o mandado de segurança, o habeas corpus, o habeas data, o mandado de injunção, o agravo de instrumento, a medida cautelar preparatória e a incidental, a reclamação e o processo criminal com réu preso. (Incluído pelo art. 1º da Resolução nº 42, de 24/11/2011)

§1º-A Nos 3 (três) dias úteis que antecedem os afastamentos por mais de 30 (trinta) dias e as férias de Desembargador, não lhe serão distribuídos autos de processos com pedido de tutela de urgência (tutela antecipada e cautela). (Redação dada pelo art. 1º da Resolução nº 22/2015, de 24/09/15)

§2º Não serão distribuídos processos a desembargador no período de noventa dias que antecede a aposentadoria compulsória ou voluntária, neste último caso, desde que previamente comunicada, por escrito, ao Tribunal. (Incluído pelo art. 1º da Resolução nº 42, de 24/11/2011)

Art. 139-A. Distribuídos, os autos serão imediatamente conclusos ao relator, que, em 30 (trinta) dias, depois de elaborar o voto, restituir-los-á, com relatório, à secretaria. [\(Redação dada pelo art. 8º da Resolução nº 06/2016, de 04/04/2016\)](#)

Art. 140. *(Revogado)* [\(Redação dada pelo art. 3º da Resolução nº 01, de 07/03/1996\)](#)

Art. 141. Ressalvados os processos de competência do Tribunal Pleno ou das Câmaras Reunidas, os feitos criminais serão distribuídos pelos desembargadores das Câmaras Criminais, e, os Cíveis, pelos desembargadores das Câmaras Cíveis. [\(Redação dada pelo art. 6º da Resolução nº 01/1999, de 10/02/1999\)](#)

Art. 142. Distribuído um feito cível a determinado Desembargador, ficará automaticamente firmada a competência da Câmara Especializada Cível a que integre, inclusive para os processos acessórios, ressalvada as competências das Câmaras Reunidas ou do Tribunal Pleno.

Art. 143. Ficará sem efeito a distribuição, tanto ao Desembargador quanto à correspondente Câmara, segundo dispõe o artigo anterior, quando, conclusos os autos ao Relator, este declinar impedimento ou suspeição.

Art. 144. Em caso de impedimento ou suspeição do Relator, a quem o feito houver sido distribuído, será feita nova distribuição, operando-se, oportunamente, a compensação.

Art. 145. A distribuição de ação originária e de recurso cível ou criminal torna o órgão e o relator preventos, observada a legislação processual respectiva, para todos os feitos posteriores, referentes ao mesmo processo ou procedimento, tanto na ação de conhecimento quanto na de execução, ressalvadas as hipóteses de suspeição ou de impedimento supervenientes, procedendo-se à devida compensação. [\(Redação dada pelo art. 2º da Resolução nº 42/2011, de 24/11/2011\)](#)

§1º Vencido o Relator, a prevenção referir-se-á ao Magistrado designado para lavrar o acórdão. [\(Incluído pelo art. 2º da Resolução nº 42/2011, de 24/11/2011\)](#)

§2º A prevenção, se não for concedida de ofício, poderá ser arguida por qualquer das partes ou pelo órgão do Ministério Público até o início do julgamento. [\(Incluído pelo art. 2º da Resolução nº 42/2011, de 24/11/2011\)](#)

§3º A certidão de prevenção constará do termo de autuação e distribuição, cabendo ao relator determinar nova distribuição, caso entenda não se tratar de prevenção. [\(Incluído pelo art. 2º da Resolução nº 42/2011, de 24/11/2011\)](#)

Art. 146. Sempre que a prevenção for reconhecida após a distribuição do feito, implicando mudança de competência, operar-se-á, oportunamente, a compensação. [\(Redação dada pelo art. 3º da Resolução nº 42, de 24/11/2011\)](#)

Art. 147. Assim que recebida, a reclamação será autuada e distribuída ao relator do processo principal, sempre que possível. [\(Redação dada pelo art. 9º da Resolução nº 06/2016, de 04/04/2016\)](#)

Art. 148. Os embargos declaratórios e as questões incidentes terão como Relator o Desembargador que houver lavrado o Acórdão ou o do processo principal.

Art. 149. A ação penal será distribuída ao mesmo Relator do inquérito.

Art. 150. *(Revogado)* [\(Redação dada pelo art. 85 da Resolução nº 06/2016, de 04/04/2016\)](#)

Art. 151. Na ação rescisória e de revisão criminal, serão excluídos da distribuição, sempre que possível, os julgadores que hajam participado do julgamento de que se originou a decisão rescindenda ou objeto da revisão criminal. [\(Redação dada pelo art.](#)

[10 da Resolução nº 06/2016, de 04/04/2016\)](#)

Art. 152. Se o Desembargador deixar o Tribunal, se for eleito Presidente ou Corregedor da Justiça, ou se vier a transferir-se de Câmara, os processos de que era Relator serão distribuídos ao Desembargador nomeado ou ao que passar a preencher sua vaga no órgão judicante.

### **Seção V - Das Atas e dos Termos**

Art. 153. As atas consignarão de modo sucinto, o que se passar nas sessões, e serão submetidas a aprovação na sessão seguinte, adiando-se a aprovação para outra oportunidade, na hipótese de circunstância de ordem relevante.

Art. 154. Contra erro contido em ata, poderá o interessado reclamar, dentro de 48 horas, em petição dirigida ao Presidente do Tribunal, das Câmaras Reunidas ou Especializadas, conforme o caso.

Parágrafo único. Não se admitirá a reclamação a pretexto de modificar o julgado.

Art. 155. A petição será entregue ao protocolo, e daí, encaminhada ao encarregado da ata, que a levará a despacho, no mesmo dia, com sua informação.

Art. 156. Se o pedido for julgado procedente, serão feitas retificação da ata e nova publicação.

Art. 157. Na oportunidade de ser a ata submetida à aprovação, poderão os integrantes do órgão judicante apresentar impugnações e propor retificação, as quais serão submetidas à deliberação do colegiado.

Art. 158. Os termos mencionarão, em resumo, o essencial do que se passar nas audiências, inclusive requerimentos e alegações das partes e despachos do Relator, e, depois de lidos e achados conforme pelos presentes, serão subscritos pelo Desembargador que presidir a audiência e pelos interessados.

### **Seção VI - Das Decisões**

Art. 159. O julgamento dos órgãos colegiados constarão de acórdãos, que serão redigidos, datados e assinados. ([Redação dada pelo art. 11 da Resolução nº 06/2016, de 04/04/2016](#))

Art. 160. As decisões proferidas verbalmente e gravadas ou taquigrafadas terão os seus textos submetidos aos Desembargadores para a devida revisão.

Art. 161. Os textos submetidos à revisão e não devolvidos pelos Desembargadores no prazo de vinte dias, contados da respectiva entrega, passarão a constar dos registros da Secretaria, com observação de não terem sido revistos.

Art. 162. Os acórdãos serão lavrados pelo Relator do feito, ou, se este for vencido, pelo autor do primeiro voto vencedor, designado para a lavratura pelo Presidente, e apresentados à conferência dentro do prazo legal.

Parágrafo único. Os acórdãos, assim como os votos que os integram, além dos demais atos processuais, podem ser redigidos em documento eletrônico inviolável e assinados eletronicamente, na forma da lei, devendo ser impressos para juntada aos autos do processo quando este não for eletrônico. ([Redação dada pelo art. 12 da Resolução nº 06/2016, de 04/04/2016](#))

Art. 163. É facultado a qualquer Desembargador, que haja participado do julgamento, exarar os fundamentos do seu voto, vencedor ou vencido.

Art. 164. Tanto o acórdão quanto os votos nele exarados deverão espelhar fielmente o que tenha sido decidido, por ocasião do julgamento, e, bem assim, os fundamentos invocados nessa oportunidade.

§ 1º. Não será considerado fundamentado o acórdão, ou qualquer outra decisão judicial, de caráter monocrático, seja ela interlocutória ou extintiva do processo, que: [\(Redação dada pelo art. 13 da Resolução nº 06/2016, de 04/04/2016\)](#)

I – se limitar à indicação, à reprodução ou à paráfrase de atos normativos, sem explicar sua relação com a causa ou a questão decidida; [\(Redação dada pelo art. 13 da Resolução nº 06/2016, de 04/04/2016\)](#)

II – empregar conceitos jurídicos indeterminados, sem explicar o motivo concreto de sua incidência no caso; [\(Redação dada pelo art. 13 da Resolução nº 06/2016, de 04/04/2016\)](#)

III – invocar motivos que se prestariam a justificar qualquer outra decisão; [\(Redação dada pelo art. 13 da Resolução nº 06/2016, de 04/04/2016\)](#)

IV – não enfrentar todos os argumentos deduzidos no processo capazes de, em tese, infirmar a conclusão adotada pelo julgador; [\(Redação dada pelo art. 13 da Resolução nº 06/2016, de 04/04/2016\)](#)

V – se limitar a invocar precedente ou enunciado de súmula, sem identificar seus fundamentos determinantes nem demonstrar que o caso sob julgamento se ajusta àqueles fundamentos; [\(Redação dada pelo art. 13 da Resolução nº 06/2016, de 04/04/2016\)](#)

VI – deixar de seguir enunciado de súmula, jurisprudência ou precedente invocado pela parte, sem demonstrar a existência de distinção no caso em julgado ou a superação do entendimento; [\(Redação dada pelo art. 13 da Resolução nº 06/2016, de 04/04/2016\)](#)

§ 2º. No caso de colisão entre normas, o julgador deve justificar o objeto e os critérios gerais da ponderação efetuada, enunciando as razões que autorizam a interferência na norma afastada e as premissas fáticas que fundamentam a conclusão. [\(Redação dada pelo art. 13 da Resolução nº 06/2016, de 04/04/2016\)](#)

§ 3º. A decisão judicial deve ser interpretada a partir da conjugação de todos os seus elementos e em conformidade com o princípio da boa-fé. [\(Redação dada pelo art. 13 da Resolução nº 06/2016, de 04/04/2016\)](#)

Art. 165. Os acórdãos trarão, em seu todo uma ementa ou súmula do que nele se contém e deverão consignar que a decisão haja sido tomada por unanimidade ou por maioria de votos.

Art. 166. Serão assinados os acórdãos pelo Desembargador Relator, pelo Presidente e pelo Procurador Geral da Justiça, bem como, se for o caso, pelo Desembargador que fizer declaração de voto.

Parágrafo único. Não sendo possível colher-se a assinatura de alguns julgadores, serão consignados, ao pé do acórdão, que o Desembargador participou do julgamento e votou, na conformidade da conclusão do acórdão ou em sentido diverso.

Art. 167. Não havendo impugnação ao texto do acórdão e achando-se este assinado pelos participantes do julgamento, o Presidente o subscreverá, dando-o por conferido.

Art. 168. Conferido o acórdão, a secretaria providenciará a publicação de sua ementa no

órgão oficial no prazo de 10 (dez) dias. (Redação dada pelo art. 14 da Resolução nº 06/2016, de 04/04/2016)

§ 1º. Não publicado o acórdão no prazo de 30 (trinta) dias, contados da data da sessão de julgamento, as notas taquigráficas ou as transcrições do áudio da respectiva sessão o substituirão, para todos os fins legais, independentemente de revisão. (Redação dada pelo art. 14 da Resolução nº 06/2016, de 04/04/2016)

§ 2º. No caso do § 1º, o presidente do Tribunal lavrará, de imediato, as conclusões e a ementa e mandará publicar o acórdão. (Redação dada pelo art. 14 da Resolução nº 06/2016, de 04/04/2016)

§ 3º. Os despachos e as decisões monocráticas, de natureza interlocutória ou extintiva do processo, serão publicadas na íntegra no diário de justiça eletrônico. (Redação dada pelo art. 14 da Resolução nº 06/2016, de 04/04/2016)

## **CAPÍTULO X - DAS SESSÕES E DAS AUDIÊNCIAS**

### **Seção I - Das Sessões Ordinárias, Extraordinárias e Solenes**

#### **Subseção I - Disposições gerais**

(A Subseção I consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça)

Art. 169. O Tribunal Pleno, as Câmaras Reunidas e as Câmaras Especializadas se reunirão, ordinariamente, nos dias designados, e, extraordinariamente, mediante convocação especial.

Art. 170. Além das sessões ordinárias e extraordinárias, o Tribunal Pleno poderá realizar sessões solenes:

I – para dar posse ao Presidente, ao Vice-Presidente, ao Corregedor Geral da Justiça.

II – para dar posse ao Desembargador.

III – em razão de acontecimento de relevância, quando convocado por deliberação plenária em sessão administrativa.

Parágrafo único. O cerimonial das sessões solenes será regulado por ato do Presidente.

Art. 171. As sessões ordinárias começarão às 09.00 horas e poderão estender-se além das 12.00 horas, se prorrogadas, com intervalo de 30 (trinta) minutos. (Redação dada pelo art. 2º da Resolução nº 43/2016, de 24/12/2016)

§ 1º. Será de 15 (quinze) minutos o prazo máximo de tolerância para o início da sessão. (Incluído pelo art. 2º da Resolução nº 43/2016, de 24/12/2016)

§2º. Transcorrido o prazo para início, fica autorizado o Presidente de cada órgão (Pleno, Câmaras Reunidas e Câmaras Especializadas) a adiar a sessão, determinando-se ao secretário que conste em ata o motivo do adiamento e os nomes dos membros presentes e ausentes (justificada e injustificadamente). (Incluído pelo art. 2º da Resolução nº 43/2016, de 24/12/2016)

Art. 172. As sessões extraordinárias terão início à hora designada e serão encerradas quando cumprido o fim a que se destinarem, devendo ser convocadas com antecedência mínima de 48 horas, salvo deliberação diversa adotada por maioria absoluta do Tribunal de Justiça.

Art. 173. Os órgãos jurisdicionais e administrativos do Tribunal de Justiça, em função de suas atribuições e competências se reunirão: (Redação dada pelo art. 1º da Resolução nº 04/2014, de 27/02/2014)

I – o Tribunal Pleno, às quintas-feiras, para funções judicantes, exceto na última quinta-feira de cada mês, destina a assuntos administrativos; (Redação dada pelo art. 1º da Resolução nº 10/2013, de 08/08/2013)

II – a 1ª, 2ª e 4ª Câmaras Especializadas Cíveis, às terças-feiras; (Redação dada pelo art. 1º da Resolução nº 18/2013, de 08/11/2013)

III – 1ª e 2ª Câmaras Especializadas Criminais e a 3ª Câmara Especializada Cível, às quartas-feiras; (Redação dada pelo art. 1º da Resolução nº 10/2013, de 08/11/2013)

IV – as Câmaras Reunidas: (Redação dada pelo art. 1º da Resolução nº 10/2013, de 08/08/2013)

a) Cíveis, às terceiras sextas-feiras de cada mês; (Redação dada pelo art. 1º da Resolução nº 10/2013, de 08/08/2013)

b) Criminais, às segundas sextas-feiras de cada mês. (Redação dada pelo art. 1º da Resolução nº 04/2014, de 27/02/2014)

Art. 174. As sessões e votações serão públicas, ressalvada a hipótese prevista no art. 93, inciso IX, da Constituição Federal e as disposições inscritas nos arts. 6º, 67 e 417, § 2º, I, deste Regimento. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

Art. 175. O Plenário, que se reúne com a presença da maioria absoluta dos seus membros, é dirigido pelo Presidente do Tribunal. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

Parágrafo único. Será observado o quorum de dois terços para as deliberações, quando preceitos constitucionais, legais ou regimentais o determinam.

Art. 176. Nas sessões do Plenário, o Presidente tem assento especial à mesa na parte central; o Desembargador mais antigo ocupará a primeira cadeira da bancada, à direita, e seu imediato, a primeira da bancada, à esquerda, e, assim, sucessivamente. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

§ 1º O Procurador Geral de Justiça ocupará a direita e, o Secretário, a esquerda da mesa do Presidente. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

§ 2º Aos desembargadores, seguirão, na ordem em que forem sorteados, os juízes de direito convocados. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

§ 3º Os advogados da causa submetida a julgamento ocuparão a primeira fila de cadeiras destinadas ao público. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

Art. 177. As Câmaras Reunidas funcionarão com a presença de, pelo menos, seis desembargadores, membros das Câmaras Especializadas respectivas, convocados, no caso de falta de quorum para julgamento, juízes de direito, na forma da lei. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

Art. 178. As Câmaras Especializadas Cíveis e Criminais funcionarão com a presença de, pelo menos, três Desembargadores, convocando-se substituto legal quando, em virtude de impedimento ou afastamento de algum dos seus membros, se verificar insuficiência numérica para o quorum exigido. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

Art. 179. Observado o disposto no art. 174, serão reservadas as sessões: (Redação dada



pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

I – quando, por deliberação da maioria absoluta do Plenário, o Presidente ou qualquer outro de seus membros, pedir que o Tribunal Pleno ou Câmara se reúna em Conselho Administrativo;

II – quando a matéria a apreciar for de natureza administrativa ou se referir à economia interna do Tribunal.

Art. 180. As sessões destinadas a feitos administrativos serão reservadas. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

Parágrafo único. Nenhuma pessoa além dos desembargadores, será admitida às sessões de Conselho Administrativo e nos casos do inciso II, do artigo anterior. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

Art. 181. As decisões tomadas em sessão administrativa serão motivadas, sendo as disciplinares tomadas pelo voto de maioria absoluta do órgão julgador. (Redação dada pelo art. 8º da Resolução nº 03/1999, de 10/06/1999)

Art. 182. Não havendo *quorum* na hora regimental ou nos seguintes trinta minutos, o Presidente, ou quem o substituir, declarará que deixa de haver sessão, fazendo mencionar, no livro de atas, a ocorrência, seus motivos e circunstâncias.

### **Subseção II - Da ordem dos trabalhos**

(A Subseção II consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça)

Art. 183. Havendo quorum no Plenário, nas Câmaras Reunidas ou nas Câmaras Especializadas, o Presidente declarará aberta a sessão e obedecerá, nos trabalhos, à ordem seguinte:

I – leitura, discussão e aprovação da ata da sessão anterior;

II – conferência de acórdão;

III – relatórios, debates e decisões dos processos;

IV – decisões e deliberações administrativas;

V – indicações e propostas.

Art. 183-A. Os órgãos julgadores do Tribunal de Justiça atenderão, preferencialmente, à ordem cronológica de conclusão para proferir sentença ou acórdão. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 1º A lista da ordem cronológica de conclusão de processos aptos para julgamento deverá estar permanentemente à disposição para consulta pública em cartório e na rede mundial de computadores. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 2º Estão excluídos da regra do *caput*: (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

I - as sentenças proferidas em audiência, homologatórias de acordo ou de improcedência liminar do pedido; (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

II - o julgamento de processos em bloco para aplicação de tese jurídica firmada em julgamento de casos repetitivos; (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

III - o julgamento de recursos repetitivos ou de incidente de resolução de demandas repetitivas; (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

IV - as decisões proferidas com base nos arts. 485 e 932 da Lei n. 13.105, de 16 de março de 2015 (Código de Processo Civil de 2015); (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

V - o julgamento de embargos de declaração; (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

VI - o julgamento de agravo interno; (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

VII - as preferências legais e as metas estabelecidas pelo Conselho Nacional de Justiça; (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

VIII - os processos criminais, nos órgãos jurisdicionais que tenham competência penal; (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

IX - a causa que exija urgência no julgamento, assim reconhecida por decisão fundamentada. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 3º Após elaboração de lista própria, respeitar-se-á a ordem cronológica das conclusões entre as preferências legais. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 4º Após a inclusão do processo na lista de que trata o § 1º, o requerimento formulado pela parte não altera a ordem cronológica para a decisão, exceto quando implicar a reabertura da instrução ou a conversão do julgamento em diligência. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 5º Decidido o requerimento previsto no § 4º, o processo retornará à mesma posição em que anteriormente se encontrava na lista. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 6º Ocupará o primeiro lugar na lista prevista no § 1º ou, conforme o caso, no § 3º, o processo que: (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

I - tiver sua sentença ou acórdão anulado, salvo quando houver necessidade de realização de diligência ou de complementação da instrução; (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

II - se enquadrar na hipótese do art. 1.040, inciso II, da Lei n. 13.105, de 16 de março de 2015 (Código de Processo Civil de 2015). (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 7º. A lista da ordem cronológica de conclusão de processos aptos para julgamento será elaborada pelo Chefe da Secretaria Cartorária Cível (SESCAR Cível) do Tribunal de Justiça. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

183-B. O escrivão ou o chefe de secretaria atenderá, preferencialmente, à ordem cronológica de recebimento para publicação e efetivação dos pronunciamentos judiciais. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 1º A lista de processos recebidos para publicação e efetivação dos pronunciamentos judiciais deverá ser disponibilizada, de forma permanente, para consulta pública. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 2º Estão excluídos da regra do *caput*: (Redação dada pelo art. 15 da Resolução nº

06/2016, de 04/04/2016)

I - os atos urgentes, assim reconhecidos pelo juiz no pronunciamento judicial a ser efetivado; (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

II - as preferências legais. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 3º Após elaboração de lista própria, respeitar-se-ão a ordem cronológica de recebimento entre os atos urgentes e as preferências legais. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 4º A parte que se considerar preterida na ordem cronológica poderá reclamar, nos próprios autos, ao juiz do processo, que requisitará informações ao servidor, a serem prestadas no prazo de 2 (dois) dias. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 5º Constatada a preterição, o juiz determinará o imediato cumprimento do ato e a instauração de processo administrativo disciplinar contra o servidor. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

§ 6º. A lista de processos recebidos para publicação e efetivação dos pronunciamentos judiciais será elaborada pelo Chefe da Secretaria Cartorária Cível (SESCAR Cível) do Tribunal de Justiça. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

183-C. A primeira lista da ordem cronológica de conclusão de processos aptos para julgamento (art. 183-A) observará a antiguidade da distribuição entre os já conclusos no dia 17 de março de 2016. Os processos que forem conclusos para julgamento a partir do dia 18 de março de 2016 serão incluídos na lista de acordo com a data de conclusão para julgamento, nos termos do art. 183-A deste Regimento. (Redação dada pelo art. 15 da Resolução nº 06/2016, de 04/04/2016)

### **Subseção III - Da prioridade e da preferência**

(A Subseção III consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça)

Art. 184. Terão prioridade para julgamento:

I – os *habeas corpus*;

II – as causas criminais e, dentre estas, as de réus presos;

III – os conflitos de jurisdição ou de competência e os de atribuições;

IV – os mandados de segurança;

V – as reclamações

§ 1º Observando o disposto no *caput*, os processos de cada classe serão chamados pela ordem de antiguidade decrescente dos respectivos relatores.

§ 2º Em cada classe o relator seguirá o critério da ordem crescente de numeração dos feitos.

§ 3º Preferirá aos demais, na sua classe, o processo em mesa, cujo julgamento tenha sido iniciado.

Art. 185. Nas Câmaras Criminais, os recursos em sentido estrito serão julgados antes das apelações e, nas Câmaras Cíveis, os agravos terão preferência em relação às

apelações.

Art. 186. O agravo de instrumento será julgado antes da apelação interposta no mesmo processo, ainda que ambos os recursos tenham de ser julgados na mesma sessão. (Redação dada pelo art. 16 da Resolução nº 06/2016, de 04/04/2016)

Parágrafo único - (Revogado) (Redação dada pelo art. 85 da Resolução nº 06/2016, de 04/04/2016)

Art. 187. As pautas de julgamento serão organizadas tendo em vista o disposto nos art. 188, 189, 190 e 191.

Art. 188. Em caso de urgência justificada, poderá o Relator propor preferência para o julgamento de determinado feito.

Art. 189. Ressalvadas as preferências legais e regimentais, os recursos, a remessa necessária e os processos da competência originária do Tribunal serão julgados na seguinte ordem: (Redação dada pelo art. 17 da Resolução nº 06/2016, de 04/04/2016)

I – aqueles nos quais houver sustentação oral, observada a ordem dos requerimentos, tendo absoluta prioridade aqueles processos nos quais advogada gestante ou lactante tiver que realizar sustentação oral; (Redação dada pelo art. 17 da Resolução nº 06/2016, de 04/04/2016)

II – os requerimentos de preferência apresentados até o início da sessão de julgamento; (Redação dada pelo art. 17 da Resolução nº 06/2016, de 04 de abril de 2016, publicado no D.J.E. nº 7.951, em 07/04/2016, pp. 02/12)

III – aqueles cujo julgamento tenha sido iniciado em sessão anterior; e (Redação dada pelo art. 17 da Resolução nº 06/2016, de 04/04/2016)

IV – Os demais casos. (Redação dada pelo art. 17 da Resolução nº 06/2016, de 04/04/2016)

Art. 190. Poderá ser deferida preferência, a requerimento do Procurador Geral de Justiça, de Julgamento relativo a processos em que houver medida cautelar.

#### **Subseção IV - Da sustentação oral**

(A Subseção IV consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça)

Art. 191. Na sessão de julgamento, depois da exposição da causa pelo relator, o presidente dará a palavra, sucessivamente, ao recorrente, ao recorrido, e, nos casos de sua intervenção, ao membro do Ministério Público, pelo prazo improrrogável de 15 (quinze) minutos para cada um, a fim de sustentarem as suas razões de direito e de fato: (Redação dada pelo art. 18 da Resolução nº 06/2016, de 04/04/2016)

I – no recurso de apelação; (Redação dada pelo art. 18 da Resolução nº 06/2016, de 04/04/2016)

II – no agravo de instrumento interposto contra decisões interlocutórias que versem sobre tutelas provisórias de urgência ou da evidência; (Redação dada pelo art. 18 da Resolução nº 06/2016, de 04/04/2016)

III – na ação rescisória, no mandado de segurança e na reclamação; (Redação dada pelo art. 18 da Resolução nº 06/2016, de 04/04/2016)

IV – em outras hipóteses previstas em lei ou neste Regimento; (Redação dada pelo art. 18 da Resolução nº 06/2016, de 04/04/2016)

§ 1º Nos demais julgamentos o Presidente do Tribunal, das Câmaras Reunidas e das Câmaras Especializadas, dará a palavra, sucessivamente, ao autor, ao recorrente, ao peticionário ou ao impetrante, e ao réu; ao recorrido ou ao impetrado, para a sustentação de suas alegações, pelo prazo máximo de quinze minutos, excetuada a ação penal originária na

qual será de uma hora, prorrogável pelo Tribunal.

§ 2º O representante do Ministério Público terá prazo igual ao das partes, salvo disposição legal em contrário.

§ 3º Se houver litisconsortes não representados pelo mesmo advogado, o prazo será contado em dobro e dividido igualmente entre os do mesmo grupo.

§ 4º Intervindo terceiro, para excluir autor e réu, terá prazo próprio, igual ao das partes.

§ 5º Havendo assistente, na ação penal pública, falará depois do representante do Ministério Público a menos que o recurso seja dele.

§ 6º Nos processos criminais, havendo corréus que sejam coautores, se não tiverem o mesmo defensor, o prazo será contado em dobro e dividido igualmente entre os defensores.

§ 7º O Procurador Geral de Justiça falará depois do autor da ação penal privada.

§ 8º Se, em ação penal, houver recurso de corréus em posição antagônica, cada grupo terá horário completo para falar.

§ 9º. Nos processos de competência originária, previstos no inciso III do *caput* deste artigo, caberá sustentação oral no agravo interno interposto contra decisão monocrática do relator que extinguir a ação de competência originária do Tribunal. [\(Redação dada pelo art. 19 da Resolução nº 06/2016, de 04/04/2016\)](#)

### **Subseção V -Do julgamento**

[\(A Subseção V consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça\)](#)

Art. 192. Concluído o debate oral, o Presidente dará a palavra ao Relator, para proferir seu voto, e, em seguida, ao revisor, se houver, seguindo-se os votos dos demais Desembargadores, na ordem decrescente de antiguidade.

§ 1º Os juízes de direito convocados votarão após os Desembargadores.

§ 2º Os desembargadores poderão antecipar o voto se o Presidente autorizar.

§ 3º. Proferidos os votos, o presidente anunciará o resultado do julgamento, designando para redigir o acórdão o relator ou, se vencido este, o autor do primeiro voto vencedor. [\(Redação dada pelo art. 20 da Resolução nº 06/2016, de 04/04/2016\)](#)

§ 4º. O voto poderá ser alterado até o momento da proclamação do resultado pelo presidente, salvo aquele já proferido por desembargador afastado ou substituído. [\(Redação dada pelo art. 20 da Resolução nº 06/2016, de 04/04/2016\)](#)

§ 5º. O voto vencido será necessariamente declarado e considerado parte integrante do acórdão, para todos os fins legais, inclusive de prequestionamento. [\(Redação dada pelo art. 20 da Resolução nº 06/2016, de 04/04/2016\)](#)

Art. 193. O relator ou outro julgador que não se considerar habilitado a proferir imediatamente o seu voto poderá solicitar vista pelo prazo máximo de 10 (dez) dias, após o qual o recurso será reincluído em pauta para julgamento na sessão seguinte à

data da devolução, salvo se na sessão originariamente convocada o julgamento for expressamente adiado para a primeira sessão seguinte. (Redação dada pelo art. 21 da Resolução nº 06/2016, de 04/04/2016)

§ 1º Nos julgamentos, o pedido de vista não impede votem os Desembargadores que se tenham por habilitados a fazê-lo.

§ 2º Ao reencentar-se o julgamento, serão computados os votos já proferidos pelos Desembargadores, ainda que não compareçam ou hajam deixado o exercício do cargo, mesmo que seja o Relator.

3º. Se os autos não forem devolvidos tempestivamente ou se não for solicitada prorrogação de prazo de no máximo mais 10 (dez) dias, o presidente do órgão os requisitará para julgamento do recurso (Incluído pelo art. 21 da Resolução nº 06/2016, de 04/04/2016).

§ 4º. Quando requisitar os autos, na forma do § 3º, se aquele que fez o pedido de vista ainda não tiver habilitado a votar, o presidente convocará o substituto, na forma estabelecida neste Regimento. (Incluído pelo art. 21 da Resolução nº 06/2016, de 04/04/2016)

Art. 194. Em caso de afastamento, a qualquer título, em período superior a trinta dias, o julgamento que tiver sido iniciado prosseguirá, computando-se o voto que haja proferido.

Art. 195. Não participarão do julgamento os Desembargadores que não tenham assistido ao relatório ou aos debates, salvo quando, não tendo havido sustentação oral, se derem por esclarecidos.

Art. 196. Cada Desembargador poderá falar duas vezes sobre o assunto em discussão e mais uma vez, se for o caso, para explicar a modificação do voto.

Parágrafo único - Nenhum desembargador falará sem autorização do Presidente, nem interromperá a quem estiver usando a palavra, salvo para apartes, quando solicitados e concedidos.

Art. 197. Nos feitos de competência do Tribunal Pleno, salvo em arguição de inconstitucionalidade, *habeas corpus*, matéria administrativa e questões de ordem, o Presidente somente votará se houver empate na votação.

Parágrafo único. No julgamento de *habeas corpus* pelo Plenário, havendo empate na votação, o Presidente proclamará a decisão mais favorável ao paciente.

Art. 198. Os Presidentes das Câmaras Reunidas, das Câmaras Especializadas Cíveis e das Câmaras Especializadas Criminais terão sempre direito a voto.

Art. 199. Qualquer questão preliminar, suscitada no julgamento, será decidida antes do mérito, deste não se conhecendo se incompatível com a decisão daquela.

§ 1º. O órgão colegiado competente para o julgamento do recurso, ou, se for o caso, da ação originária, poderá determinar, em sede de preliminar, as providências indicadas pelo art. 91, §§ 3º, 4º e 5º, deste Regimento, quando não tiverem sido tomadas pelo próprio relator do processo, desde que se trate de vício sanável, inclusive daquele que possa ser conhecido de ofício. (Redação dada pelo art. 22 da Resolução nº 06/2016, de 04/04/2016)

§ 2º. (Revogado) (Redação dada pelo art. 85 da Resolução nº 06/2016, de 04/04/2016)

§ 3º Rejeitada a preliminar, ou se com ela for compatível a apreciação do mérito, seguem-se a discussão e o julgamento da matéria principal, pronunciando-se sobre esta

os Desembargadores vencidos na preliminar.

Art. 200. Uma vez iniciado, o julgamento será ultimado na mesma sessão, ainda que excedida a hora regimental.

Art. 201. O acórdão será apresentado, para conferência, na primeira sessão seguinte à do julgamento, pelo Desembargador incumbido de lavrá-lo.

Art. 202. O Presidente da sessão manterá a disciplina no recinto, advertindo ou fazendo retirar da sala quem perturbar os trabalhos, mandando prender ou autuar os que cometerem crime ou contravenção penal.

Art. 203. Os Desembargadores usarão obrigatoriamente, nas audiências, nas sessões solenes, nos atos e sessões de julgamento, vestes talares, de modelo aprovado pelo Tribunal.

Art. 203-A. Os agravos internos e embargos de declaração poderão, a critério do relator, ser submetidos a julgamento em ambiente eletrônico, por meio de sessões virtuais, observadas as respectivas competências das Câmaras ou do Pleno. [\(Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016\).](#)

Art. 203-B. As sessões virtuais serão realizadas semanalmente, com início às sextas-feiras, respeitado o prazo de 5 (cinco) dias úteis exigido pelo art. 935 do Código de Processo Civil entre a data da publicação da pauta no DJe, com divulgação das listas no sítio eletrônico do Tribunal, e o início do julgamento. § 1º. O relator inserirá ementa, relatório e voto no ambiente virtual e, com o início do julgamento, os demais desembargadores terão até 7 (sete) dias corridos para manifestação. [\(Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016\).](#)

§ 2º. Considerar-se-á que acompanhou o relator o desembargador que não se pronunciar no prazo previsto no § 1º. [\(Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016\).](#)

§ 3º. A ementa, o relatório e voto somente serão tornados públicos depois de concluído seu julgamento. [\(Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016\).](#)

§ 4º. O início da sessão de julgamento definirá a composição do Plenário e das Câmaras. [\(Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016\).](#)

§ 5º. Os votos serão computados na ordem cronológica de sua manifestação. [\(Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016\).](#)

Art. 203-C. O relator poderá retirar do sistema qualquer lista ou processo antes de iniciado o respectivo julgamento. [\(Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016\).](#)

Art. 203-D. Não serão julgados em ambiente virtual a lista ou o processo com pedido de: I. destaque ou vista por um ou mais desembargadores; II. destaque por qualquer das partes, desde que requerido em até 24 (vinte e quatro) horas antes do início da sessão e deferido o pedido pelo relator.

[\(Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016\).](#)

Parágrafo único. Também não serão julgados por meio virtual os agravos em que houver pedido de sustentação oral, quando cabível (§ 3º. art. 937, do CPC). [\(Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016\).](#)

Art. 203-E. A lista ou processo objeto de pedido de vista ou de destaque serão encaminhados ao órgão colegiado competente para julgamento presencial, oportunidade em que os desembargadores poderão renovar ou modificar os seus votos. [\(Incluído pelo](#)

art. 1º da Resolução nº 24/2016, de 15/09/2016).

Art. 203-F. Os desembargadores poderão votar nas listas como um todo ou em cada processo separadamente. (Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016).

§ 1º. As opções de voto serão as seguintes: (Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016).

a - acompanho o Relator; (Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016).

b - acompanho o Relator com ressalva de entendimento; (Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016).

c - divirjo do Relator; ou (Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016).

d - acompanho a divergência. (Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016).

§ 2º. Eleitas as opções "b" ou "c", o desembargador declarará o seu voto no próprio sistema. (Incluído pelo art. 1º da Resolução nº 24/2016, de 15/09/2016).

## **Seção II - Das Audiências**

Art. 204. Serão públicas as audiências:

I – para distribuição dos feitos;

II – para instrução do processo, salvo motivo relevante.

Art. 205. O Desembargador que presidir a audiência deliberará sobre o que lhe for requerido.

§ 1º Respeitada a prerrogativa dos advogados, nenhum dos presentes se dirigirá ao Presidente da audiência, a não ser de pé e com sua licença.

§ 2º O Secretário da audiência lavrará o termo, do qual fará constar o que nela tiver ocorrido e depois de lido e achado conforme pelos presentes, será assinado pelo desembargador que presidiu o ato e pelas demais pessoas presentes.

## **CAPÍTULO XI - DOS FEITOS DE COMPETÊNCIA ORIGINÁRIA DO TRIBUNAL**

### **Seção I -Do Pedido Originário de Habeas Corpus**

Art. 206. O Tribunal de Justiça concederá *habeas-corpus* sempre que alguém sofrer ou se achar ameaçado de sofrer violência ou coação em sua liberdade de locomoção:

I – quando o constrangimento partir de Secretário de Estado, do Corregedor Geral da Justiça ou de Juiz de Direito;

II – quando se tratar de crime sujeito à jurisdição privativa do Tribunal;

III – quando houver iminente perigo de consumir-se a violência antes que o Juiz de Direito dela possa tomar conhecimento.

Art. 207. O *habeas-corpus* pode ser impetrado:

I – por qualquer pessoa, em seu favor ou de outrem;

II – pelo Ministério Público.



Art. 208. A petição de *habeas corpus* deverá conter:

I – o nome do impetrado, bem como o do paciente e do coator;

II – os motivos do pedido e, quando possível, a prova documental dos fatos alegados;

III – a assinatura do impetrante ou de alguém a seu rogo, se não souber ou não puder escrever.

Art. 209. O Relator requisitará informações, do apontado coator e poderá:

I – ordenar diligências necessárias à instrução do pedido, no prazo que estabelecer, se a deficiência deste não for imputável ao impetrante;

II – determinar a apresentação do paciente à sessão do julgamento se entender conveniente;

III – expedir salvo-conduto, no *habeas corpus* preventivo, em favor do paciente, até decisão do feito, se houver grave risco de consumir-se a violência.

Art. 210. Instruído o processo e ouvido o Procurador Geral de Justiça, em dois dias o Relator o colocará em mesa para julgamento na primeira sessão do Tribunal Pleno.

Parágrafo único. Não se conhecerá do pedido de *habeas-corpus* desautorizado pelo paciente.

Art. 211. O Tribunal poderá, de ofício:

I – usar da faculdade prevista no art. 206, II;

II – expedir ordem de *habeas corpus* quando, no curso de qualquer processo, verificar que alguém sofre ou se acha ameaçado de sofrer violência ou coação em sua liberdade de locomoção, por ilegitimidade ou abuso de poder.

Art. 212. A decisão concessiva de *habeas corpus* será imediatamente comunicada às autoridades a quem couber cumpri-la, sem prejuízo da remessa de cópia autenticada do acórdão.

Parágrafo único. A comunicação mediante ofício, telegrama ou radiograma, bem como o salvo-conduto, em caso de ameaça de violência ou coação, serão firmados pelo Presidente do Tribunal.

Art. 213. Ordenada a soltura do paciente, em virtude de *habeas corpus*, a autoridade que, por má-fé ou evidente abuso de poder, tiver determinado a coação será condenada nas custas, remetendo-se ao Ministério Público traslado das peças necessárias à apuração de sua responsabilidade penal.

Art. 214. O carcereiro ou o diretor da prisão, o escrivão, o oficial de Justiça ou a autoridade judiciária, policial ou militar que embaraçarem ou procrastinarem o encaminhamento do pedido de *habeas corpus*, as informações sobre a causa da violência, coação ou ameaça ou a condução e apresentação do paciente, serão multados na forma da legislação processual vigente, sem prejuízo de outras sanções penais e administrativas.

Art. 215. Havendo desobediência ou retardamento abusivo no cumprimento da ordem de *habeas corpus*, por parte do detentor ou carcereiro, o Presidente do Tribunal expedirá mandado de prisão contra o desobediente e oficiará ao Ministério Público, a fim de que promova a ação penal.

Parágrafo único. Na hipótese deste artigo, o Tribunal tomará as providências necessárias ao cumprimento da decisão, com emprego dos meios legais cabíveis, e determinará, se necessário, a apresentação do paciente ao Presidente ou a magistratura local por ele designado.

Art. 216. As fianças que se tiverem de prestar perante o Tribunal, em virtude de *habeas corpus*, serão processadas pelo Presidente, a menos que este delegue essa atribuição a outro magistrado.

Art. 217. Se, pendente o processo de *habeas corpus*, cessar a violência ou a coação, será julgado prejudicado o pedido, podendo, porém, o Tribunal declarar a ilegalidade do ato e tomar as providências cabíveis para punição do responsável.

## **Seção II - Do Pedido Originário de Mandado de Segurança**

Art. 218. O plenário do Tribunal de Justiça concederá mandado de segurança para proteger direito líquido e certo não amparado por *habeas corpus* quando o responsável pela ilegalidade ou abuso de poder for o Governador, a Assembleia Legislativa ou a respectiva Mesa, o próprio Tribunal de Justiça ou o Presidente, o Corregedor Geral da Justiça e o Procurador Geral de Justiça.

Parágrafo único. Quando requeridos contra atos de Juiz de Direito, os mandados de segurança serão processados e julgados por Câmara Cível.

Art. 219. Não se dará mandado de segurança quando estiver em causa:

I – ato de que caiba recurso administrativo com efeito suspensivo, independente de caução;

II – despacho ou decisão judicial, de que caiba recurso, ou que seja suscetível de correição;

III – ato disciplinar salvo se praticado por autoridade incompetente ou com inobservância de formalidade essencial.

Art. 220. A petição inicial, que deverá preencher os requisitos dos arts. 282 e 283 do Código de Processo Civil, será apresentada em duas vias, e os documentos que instruírem a primeira deverão ser reproduzidos, por cópia na segunda.

§ 1º Quando a parte não puder instruir, desde logo, a sua petição, em vista de impedimento ou demora em obter certidões ou cópias autenticadas de notas ou registros em repartições ou estabelecimento público, o Relator concederá prazo para esse fim.

§ 2º Se houver recusa de repartição, estabelecimento público ou de autoridade, de fornecer, por certidão, documento que tenha em seu poder e necessário à prova do alegado no requerimento, o relator ordenará, preliminarmente, por ofício, a exibição desse documento, em original ou cópia autêntica, e marcará, para o cumprimento da ordem, o prazo de dez dias.

§ 3º Se a recusa houver partido da autoridade coatora, a ordem será feita no próprio instrumento da notificação.

§ 4º O escrivão extrairá cópias do documento para juntá-las, à segunda via da petição.

Art. 221. O relator mandará notificar a autoridade coatora para prestar informações no prazo previsto em lei.

§ 1º Quando relevante o fundamento e do ato impugnado puder resultar a ineficácia da medida, caso deferida, o relator determinará a sua suspensão, salvo nos casos vedados em lei.

§ 2º A notificação será instruída com a segunda via da inicial e cópias dos documentos, bem como do despacho concessivo da liminar, se houver.

Art. 222. A medida liminar vigorará pelo prazo de noventa dias contado de sua efetivação.

Parágrafo único. Se o impetrante criar obstáculo no normal andamento do processo, deixar de promover, por mais de três dias, os atos e diligências que lhe cumprirem, ou abandonar a causa por mais de vinte dias, o Relator *ex-officio* ou a requerimento do Ministério Público, decretará a perempção ou a caducidade da medida liminar.

Art. 223. Recebidas as informações ou transcorrido o respectivo prazo, sem o seu oferecimento, o Relator, após a vista à Procuradoria Geral de Justiça, pedirá dia para julgamento.

Art. 224. A concessão ou a denegação de segurança na vigência de medida liminar serão imediatamente comunicadas à autoridade apontada como coatora.

### **Seção III - Da Ação Penal Originária**

#### **Subseção I - Disposições gerais**

(A Subseção I consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça)

Art. 225. Nos processos por delitos comuns e funcionais, de competência originária do Tribunal de Justiça, a denúncia, nos crimes de ação pública, a queixa nos de ação privada, bem como a representação quando indispensável ao exercício da primeira, obedecerão ao que dispõe a lei processual.

Art. 226. O Relator, a quem o feito for distribuído, funcionará como juiz de instrução do processo, com as atribuições que a lei processual confere aos juízes singulares.

Art. 227. Distribuído inquérito sobre crime de ação pública, da competência originária do Tribunal, o Relator encaminhará os autos ao Procurador Geral de Justiça, que terá quinze dias para oferecer a denúncia ou requerer o arquivamento. Esse prazo será de cinco dias, se o indiciado estiver preso.

§ 1º O Procurador Geral poderá requerer, ao Relator, diligências complementares ao inquérito, as quais não interromperão o prazo para oferecimento da denúncia, se o indiciado estiver preso.

§ 2º Estando preso o indiciado, se as diligências requeridas forem indispensáveis ao oferecimento da denúncia e implicarem em ser excedido o prazo do Ministério Público, o Relator determinará o relaxamento da prisão; se não o forem, mandará que se realizem em separado, depois de oferecida a denúncia e sem prejuízo da prisão e do processo.

Art. 228. Se o inquérito versar sobre a prática de crime de ação privada, o Relator determinará a iniciativa do ofendido ou de quem, por lei, esteja autorizada a oferecer queixa.

Parágrafo único - Verificando a extinção da punibilidade, ainda que não haja iniciativa do ofendido o Relator após ouvir o Procurador Geral, pedirá dia para o julgamento, independentemente de revisão.

Art. 229. Recebida a queixa ou a denúncia, será notificado o acusado para que, no prazo improrrogável de quinze dias, apresente resposta escrita, excetuados os seguintes casos:  
I – achar-se o acusado fora do território sujeito à jurisdição do Tribunal, ou em lugar desconhecido ou incerto;

II – ser o delito inafiançável.

§ 1º A notificação acompanhada de cópia do ato de acusação e dos documentos que o instruírem, será encaminhada ao acusado sob registro postal.

§ 2º A notificação poderá ser feita por intermédio da autoridade judiciária do lugar em que se encontrar o acusado.

§ 3º O Tribunal enviará à autoridade referida no parágrafo anterior, para entrega ao notificado, cópia autêntica da acusação, do despacho do Relator e dos documentos apresentados, peças que devem ser conferidas pela Secretaria e fornecidas pelo autor.

Art. 230. A notificação de deputado estadual não será determinada sem prévia licença da Assembleia Legislativa, na conformidade da Constituição vigente.

Art. 231. Se a resposta ou defesa prévia do acusado convencer da improcedência da acusação, o Relator proporá ao Tribunal o arquivamento do processo.

Art. 232. Não sendo vencedora a opinião do relator, ou se ele não se utilizar da faculdade que lhe confere o artigo antecedente, se procederá à instrução do processo, que obedecerá, no que couber, ao procedimento comum do Código de Processo Penal.

Art. 233. Não comparecendo o acusado, ou não constituindo advogado, o Relator lhe nomeará defensor.

Art. 234. O Relator poderá delegar o interrogatório do réu e qualquer dos atos de instrução a juiz ou a outro Tribunal, que tenha competência territorial no local onde devam ser produzidos.

Art. 235. Terminada a inquirição de testemunhas, o Relator dará vista sucessiva à acusação e à defesa, pelo prazo de cinco dias, para requererem diligências, em razão de circunstâncias ou fatos apurados na instrução.

Art. 236. Concluídas as diligências acaso deferidas, mandará o Relator dar vista às partes para alegações, pelo prazo de quinze dias, sendo comum o prazo do acusador e do assistente, bem como o dos corréus.

Art. 237. Findos os prazos do artigo anterior e após ouvir o Procurador Geral na ação penal privada, pelo prazo de quinze dias, o Relator poderá ordenar diligências para sanar nulidade ou suprir falta que prejudique apuração da verdade.

Art. 238. Observando o disposto no artigo anterior, o Relator lançará o relatório e passará os autos ao Revisor que pedirá dia para julgamento.

Art. 239. Designados dia e hora para o julgamento, da designação serão intimados as partes, as testemunhas e o Ministério Público.

Parágrafo único. A Secretaria remeterá cópia do relatório aos Desembargadores logo após o pedido de dia formulado pelo Revisor.

Art. 240. A requerimento das partes ou do Procurador Geral, o Relator poderá admitir que deponham, na sessão de julgamento, testemunhas previamente arroladas, as quais serão intimadas na forma da lei.

### **Subseção II - Do julgamento**

(A Subseção II consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça)

Art. 241. Na sessão de julgamento será observado o seguinte:

I – aberta a sessão, apregoadas as partes e as testemunhas, lançado o querelante que

deixar de comparecer, atendendo-se ao disposto no artigo 29 do Código de Processo Penal, serão realizadas as demais diligências preliminares salvo ocorrendo a hipótese prevista no art. 60, inciso III, do referido Código;

II – a seguir, o Relator apresentará minucioso relatório do feito, resumindo as principais peças dos autos e a prova produzida, lendo, outrossim, se houver o aditamento ou a retificação do Revisor;

III – se algum dos Desembargadores solicitar a leitura integral dos autos ou de partes deles o Relator poderá ordenar seja ela efetuada pelo secretário;

IV – as testemunhas arroladas, que não tiverem sido dispensadas pelas partes, serão inquiridas pelo Relator e, facultativamente, pelos demais Desembargadores; em primeiro lugar as de acusação e, depois, as de defesa;

V – serão admitidas, a seguir, perguntas do Procurador Geral e das partes;

VI – serão ouvidos os peritos para esclarecimentos previamente ordenados pelo Relator, de ofício, ou a requerimento das partes ou do Procurador Geral;

VII – findas as inquirições e efetuadas quaisquer diligências que o Tribunal houver determinado, o Presidente dará a palavra, sucessivamente, ao acusador, ao órgão do Ministério Público e ao acusado ou a seu defensor, para sustentarem oralmente a acusação e a defesa, podendo cada um ocupar a tribuna durante uma hora, prorrogável pelo Presidente;

VIII – na ação penal privada, o Procurador Geral falará por último;

IX – encerrados os debates, o Tribunal passará a funcionar em sessão reservada, sem a presença das partes e do Procurador Geral, para proferir o julgamento, que será anunciado em sessão pública;

X – o julgamento será efetuado em uma ou mais sessões, a critério do Tribunal.

Art. 242. Será nomeado defensor *ad hoc* se o advogado constituído pelo réu ou o defensor anteriormente nomeado não comparecer na sessão de julgamento, a qual será adiada se aquele o requerer para exame dos autos.

Art. 243. Logo após os pregões, o réu poderá, sem justificação, recusar um dos Desembargadores, e o acusador, outro. Havendo mais de um réu ou mais de um acusador, se não entrarem em acordo, será determinado, por sorteio, quem deva exercer o direito de recusa.

Art. 244. Caberá agravo, sem efeito suspensivo, para as Câmaras Reunidas Criminais, para o Plenário do Tribunal, do despacho do relator, que:

I – receber ou rejeitar a queixa ou a denúncia, ressalvando o disposto no art. 228;

II – conceder ou denegar fiança, ou a arbitrar;

III – recusar a produção de qualquer prova ou a realização de qualquer diligência.

Art. 245. (Revogado) (Redação dada pelo art. 7º da Resolução nº 03, de 10/06/1999)

#### **Seção IV - Da Revisão Criminal**

Art. 246. Ressalvada a competência do Supremo Tribunal Federal, será admitida a revisão, pelo Plenário do Tribunal de Justiça, nos processos criminais findos em que a condenação houver sido proferida por órgão judiciário de primeira instância ou pelo próprio Tribunal, seja originariamente, seja em grau de recurso, quando:

I – a decisão condenatória for contrária ao texto expresso da lei penal ou à evidência dos

autos;

II – a decisão condenatória se fundar em depoimentos, exames ou documentos comprovadamente falsos;

III – após decisão condenatória, se descobrirem novas provas de inocência do condenado ou de circunstância que determine ou autorize diminuição especial da pena.

Art. 247. A revisão poderá ser requerida a qualquer tempo, depois de transitada em julgado a decisão condenatória, esteja ou não extinta a pena.

Art. 248. Não é admissível reiteração do pedido, com o mesmo fundamento, salvo se fundada em novas provas.

Art. 249. A revisão poderá ser pedida pelo próprio condenado ou seu procurador legalmente habilitado, ou, falecido aquele, pelo seu cônjuge, ascendente, descendente ou irmão.

Art. 250. O pedido de revisão será sempre instruído com o inteiro teor, autenticado, da decisão condenatória, com a prova de haver esta passado em julgado e com os documentos comprobatórios das alegações em que se fundar, indicadas, igualmente, as provas que serão produzidas.

Parágrafo único. Se a decisão impugnada for confirmatória de outras, estas deverão, também, vir comprovadas no seu inteiro teor.

Art. 251. O requerimento será distribuído a um Relator e um Revisor, devendo funcionar como relator um Desembargador que não tenha pronunciado decisão em qualquer fase do processo.

Art. 252. O Relator admitirá ou não as provas requeridas e determinará a produção de outras que entender necessárias, facultando o agravo regimental.

Parágrafo único. A qualquer tempo, o Relator poderá solicitar informações ao juiz de execução e requisitar os autos do processo sob revisão.

Art. 253. Se o Relator julgar insuficientemente instruído o pedido e inconveniente ao interesse da Justiça que se apensem os autos originais, o indeferirá *in limine*, cabendo, dessa decisão, agravo regimental para o Plenário.

Parágrafo único. Interposto o agravo por petição e independentemente de termo, o Relator apresentará o processo em mesa para o julgamento e o relatará, sem tomar parte na discussão.

Art. 254. Se o requerimento não for indeferido liminarmente, instruído o processo, o Relator ouvirá o requerente e o Procurador Geral, no prazo de dez dias para cada um, e, lançado o relatório, passará os autos ao Revisor, que pedirá dia para o julgamento.

Parágrafo único. Serão de dez dias os prazos do relator e do revisor para exame do processo.

Art. 255. Se julgar procedente a revisão, o Tribunal poderá absolver o acusado, alterar a classificação da infração, modificar a pena ou anular o processo.

Parágrafo único - Não poderá ser agravada, de qualquer maneira, a pena imposta pela decisão revista.

Art. 256. A absolvição implicará o restabelecimento de todos os direitos perdidos em virtude da condenação, devendo o Tribunal, se for o caso, impor a medida de segurança cabível.

Art. 257. À vista de certidão do acórdão que houver cassado ou reformado a decisão condenatória, o juiz da execução mandará juntá-la aos autos, para seu cumprimento,

determinando desde logo o que for de sua competência.

Art. 258. O Tribunal, se o interessado o requerer, poderá reconhecer o direito a uma justa indenização pelos prejuízos sofridos.

Art. 259. A indenização a que se refere o artigo anterior não será devida:

I – se o erro ou a injustiça da condenação proceder de ato ou falta imputável ao próprio impetrante, como a confissão ou a ocultação de prova em seu poder;

II – se a acusação houver sido meramente privada.

Art. 260. Quando, no curso da revisão, falecer a pessoa, cuja condenação tiver de ser revista, o Presidente do Tribunal nomeará curador para defesa.

### **Seção V - Da Ação Rescisória**

Art. 261. Ressalvada a competência do Supremo Tribunal Federal, cabe, nos casos e pela forma prevista na lei processual:

I – ao Plenário do Tribunal de Justiça, processar e julgar as ações rescisórias de seus acórdãos;

II – às Câmaras Reunidas processar e julgar as ações rescisórias dos seus acórdãos, das Câmaras Especializadas e das decisões dos juízes singulares.

Art. 262. A petição inicial, elaborada com os requisitos a que se refere o art. 968 da Lei n. 13.105, de 16 de março de 2015 (Código de Processo Civil) e com o comprovante de recolhimento do depósito de que trata o inciso II do mencionado dispositivo processual, salvo na hipótese de não obrigatoriedade de depósito, será distribuída ao relator, que mandará citar o réu, assinando-lhe prazo nunca inferior a 15 (quinze) dias nem superior a 30 (trinta) dias, para, querendo, apresentar resposta. [\(Redação dada pelo art. 23 da Resolução nº 06/2016, de 04/04/2016\)](#)

§ 1º. A escolha do relator recairá, sempre que possível, em quem não haja participado do julgamento rescindendo. [\(Redação dada pelo art. 23 da Resolução nº 06/2016, de 04/04/2016\)](#)

§ 2º. A propositura de ação rescisória não impede o cumprimento da decisão rescindenda, ressalvada a concessão de tutela provisória. [\(Redação dada pelo art. 23 da Resolução nº 06/2016, de 04/04/2016\)](#)

Art. 263. Findo o prazo a que alude o *caput* do artigo anterior, com ou sem resposta, observar-se-á, no que couber, o procedimento comum. [\(Redação dada pelo art. 24 da Resolução nº 06/2016, de 04/04/2016\)](#)

Parágrafo único. Nas hipóteses do artigo 178 do Código de Processo Civil, o Ministério Público será intimado para intervir como fiscal da ordem jurídica, quando não for parte. [\(Redação dada pelo art. 24 da Resolução nº 06/2016, de 04/04/2016\)](#)

Art. 264. Se os fatos alegados pelas partes dependerem de prova, o relator delegará a competência ao Juiz de Direito da comarca onde deva ser produzido, fixando o prazo de quarenta e cinco a noventa dias para a devolução dos autos.

Art. 265. Concluída a instrução, o Relator abrirá vista sucessiva às partes, por dez dias, para o oferecimento de razões e, após ouvido o Procurador Geral, lançará o relatório e passará os autos ao Revisor, que pedirá dia para julgamento.

Art. 266. Julgada procedente a ação, o Tribunal rescindir a decisão, proferirá, se for o caso, novo julgamento e determinará a restituição do depósito a que se refere o inciso II

do art. 968 do Código de Processo Civil. (Redação dada pelo art. 25 da Resolução nº 06/2016, de 04/04/2016)

§ 1º. Considerada, por unanimidade, inadmissível ou improcedente a ação, o Tribunal determinará a reversão, em favor do réu, da importância do depósito, sem prejuízo do disposto no § 2º do art. 82 do Código de Processo Civil. (Redação dada pelo art. 25 da Resolução nº 06/2016, de 04/04/2016)

§ 2º. No caso de julgamento não unânime pela procedência da ação rescisória, o prosseguimento do julgamento deverá ocorrer pelo Tribunal Pleno, colhendo-se, na oportunidade, os votos dos julgadores que compõe este colegiado. (Redação dada pelo art. 25 da Resolução nº 06/2016, de 04/04/2016)

§ 3º. Os julgadores que já tiverem votado, poderão rever os seus votos por ocasião do prosseguimento do julgamento pelo Tribunal Pleno. (Redação dada pelo art. 25 da Resolução nº 06/2016, de 04/04/2016)

§ 4º. Na oportunidade desse julgamento pelo Tribunal Pleno, será assegurado às partes e eventuais terceiros o direito de sustentar novamente suas razões perante os novos julgadores. (Redação dada pelo art. 25 da Resolução nº 06/2016, de 04/04/2016)

§ 5º. A improcedência da ação rescisória, ainda que em julgamento não unânime, não ensejará a oportunidade de complementação do julgamento pelo Tribunal Pleno. (Redação dada pelo art. 25 da Resolução nº 06/2016, de 04/04/2016)

## **Seção VI - Do Conflito de Jurisdição ou Competência e de Atribuições**

Art. 267. O conflito de jurisdição ou competência poderá ocorrer entre órgãos ou autoridades judiciárias; o de atribuições, entre autoridades judiciárias e administrativas.

Art. 268. Há conflito de competência quando: (Redação dada pelo art. 26 da Resolução nº 06/2016, de 04/04/2016)

I – dois ou mais juízes se declaram competentes; (Redação dada pelo art. 26 da Resolução nº 06/2016, de 04/04/2016)

II – dois ou mais juízes se consideram incompetentes, atribuindo um ao outro a competência; (Redação dada pelo art. 26 da Resolução nº 06/2016, de 04/04/2016)

III – entre dois ou mais juízes surge controvérsia acerca da reunião ou separação de processos. (Redação dada pelo art. 26 da Resolução nº 06/2016, de 04/04/2016)

Parágrafo único. O juiz que não acolher a competência declinada deverá suscitar o conflito, salvo se a atribuir a outro juiz. (Redação dada pelo art. 26 da Resolução nº 06/2016, de 04/04/2016)

Art. 269. O conflito de competência pode ser suscitado por qualquer das partes, pelo Ministério Público e pelo juiz. (Redação dada pelo art. 27 da Resolução nº 06/2016, de 04/04/2016).

Art. 270. O Ministério Público somente será ouvido nos conflitos de competência relativos aos processos previstos no art. 178 do Código de Processo Civil, mas terá qualidade de parte naqueles que suscitar.

(Redação dada pelo art. 28 da Resolução nº 06/2016, de 04/04/2016)

Art. 271. Não pode suscitar conflito a parte que, no processo, arguiu a incompetência relativa. (Redação dada pelo art. 29 da Resolução nº 06/2016, de 04/04/2016)

Parágrafo único. O conflito de competência não obsta, porém, que a parte que não o



arguiu suscitar a incompetência. (Redação dada pelo art. 29 da Resolução nº 06/2016, de 04/04/2016)

Art. 272. O conflito será suscitado ao Presidente do Tribunal:

I – pela autoridade judiciária ou administrativa, conforme o caso, por ofício;

II – pela parte ou pelo Ministério Público, por petição.

Parágrafo único. O ofício e a petição serão instruídos com os documentos necessários à prova do conflito.

Art. 273. Após a distribuição, o Relator mandará ouvir as autoridades em conflito, ou apenas a suscitada, se uma delas for suscitante; dentro do prazo, assinado pelo Relator, caberá à autoridade ou às autoridades prestar as informações.

Art. 274. Poderá o Relator, de ofício, ou a requerimento de qualquer das partes, determinar, quando o conflito for positivo, seja sobrestado o processo e, neste caso, bem assim no de conflito negativo, designar um dos órgãos para resolver, em caráter provisório, as medidas urgentes.

Parágrafo único. O relator poderá julgar de plano o conflito de competência quando sua decisão se fundar em: (Redação dada pelo art. 30 da Resolução nº 06/2016, de 04/04/2016)

I – súmula do Supremo Tribunal Federal, do Superior Tribunal de Justiça ou do próprio Tribunal; (Redação dada pelo art. 30 da Resolução nº 06/2016, de 04/04/2016)

II – tese firmada em julgamento de casos repetitivos ou em incidente de assunção de competência. (Redação dada pelo art. 30 da Resolução nº 06/2016, de 04/04/2016)

Art. 275. Decorrido o prazo, com informações ou sem elas, será ouvido, em cinco dias, o Ministério Público; em seguida o relator apresentará o conflito em sessão de julgamento.

Art. 276. Ao decidir o conflito, o Tribunal declarará qual o juízo competente, pronunciando-se também sobre a validade dos atos praticados pelo juízo incompetente. (Redação dada pelo art. 31 da Resolução nº 06/2016, de 04/04/2016)

Parágrafo único. Os autos do processo em que se manifestou o conflito serão remetidos ao juízo declarado competente.

(Redação dada pelo art. 31 da Resolução nº 06/2016, de 04/04/2016)

Art. 277. Na decisão do conflito, será compreendido como expresso o que nela virtualmente se contenha ou dela resulte.

Art. 278. No caso de conflito positivo, o Presidente poderá determinar o imediato cumprimento da decisão, lavrando-se o acórdão posteriormente.

Art. 278-A. No conflito que envolva órgãos fracionários no Tribunal, desembargadores e juízes em exercício no Tribunal, definidos na forma do art. 81, I, “g”, deste Regimento, aplicam-se as disposições desta Seção, no que couber. (Redação dada pelo art. 32 da Resolução nº 06/2016, de 04/04/2016)

## **Seção VII - Da Perda de Cargo, da Disponibilidade e da Remoção Compulsória de Magistrados**

### **Subseção I - Do procedimento**

(A Subseção I consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça).

Art. 279. O procedimento para a decretação da perda de cargo de magistrado, subordinado ao Tribunal de Justiça ou a ele pertencente terá início por determinação do mesmo Tribunal.

Art. 280. A determinação do Tribunal poderá ser tomada de ofício ou à vista de representação fundamentada do Poder Executivo ou Legislativo, do Chefe do Ministério Público ou do Conselho Federal ou Seccional da Ordem dos Advogados do Brasil.

Art. 281. Em qualquer hipótese, a instauração do processo será precedida da defesa prévia do magistrado, no prazo de quinze dias, contado da entrega da cópia do teor da acusação e das provas existentes, que lhe remeterá o Presidente do Tribunal, mediante ofício nas quarenta e oito horas imediatamente seguintes à apresentação da acusação.

Art. 282. Findo o prazo da defesa prévia, haja ou não sido apresentada, o Presidente, no dia útil imediato, convocará o Tribunal para que, em sessão reservada, decida sobre a instauração do processo, e, caso determinada esta, no mesmo dia distribuirá o feito e fará entrega ao Relator.

### **Subseção II - Do processo**

(A Subseção II consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça).

Art. 283. O Tribunal, na sessão em que ordenar a instauração do processo, como no curso deste, poderá afastar o magistrado do exercício das suas funções, sem prejuízo dos vencimentos e das vantagens, até a decisão final.

Art. 284. As provas requeridas e deferidas, bem como as que o Relator determinar de ofício, serão produzidas no prazo de vinte dias, cientes o Ministério Público, o magistrado ou o procurador por ele constituído, a fim de que possam delas participar.

Art. 285. Finda a instrução, o Ministério Público e o magistrado ou seu procurador terão, sucessivamente, vista dos autos por dez dias, para razões.

### **Subseção III - Do julgamento**

(A Subseção III consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça).

Art. 286. O julgamento será realizado em sessão reservada do Tribunal depois de relatório oral, e a decisão no sentido da penalização do magistrado só será tomada pelo voto de dois terços dos membros do colegiado, em escrutínio secreto.

Art. 287. Da decisão será publicado somente a conclusão.

Art. 288. Se a decisão concluir pela perda do cargo, será comunicada, imediatamente, ao Poder Executivo, para a formalização do ato.

Art. 289. Quando, pela natureza ou gravidade da infração penal, se torne aconselhável o recebimento de denúncia ou de queixa contra magistrado, o Tribunal poderá, em decisão tomada pelo voto de dois terços de seus membros, determinar o afastamento do cargo do magistrado denunciado.

#### **Subseção IV - Da disponibilidade e da remoção compulsória**

(A Subseção IV consta do Regimento Interno editado pelo Tribunal em 2000, mas não no texto publicado no Diário de Justiça).

Art. 290. O Tribunal poderá determinar, por motivo de interesse público, em escrutínio secreto e pelo voto de dois terços de seus efetivos:

I – a remoção de juiz de instância inferior;

II – a disponibilidade de membro do próprio Tribunal ou de instância inferior, com vencimentos proporcionais ao tempo de serviço.

Parágrafo único. Na determinação do "quorum" de decisão será aplicado o disposto no parágrafo único do art. 24, da Lei Orgânica da Magistratura Nacional.

Art. 291. O procedimento para a decretação da remoção ou disponibilidade de magistrado obedecerá ao prescrito nos arts. 279 a 285, deste Regimento.

#### **Seção VIII - Da Imposição das Penalidades de Advertência e de Censura**

Art. 292. As penas de advertência e de censura somente são aplicáveis aos juízes de primeira instância.

Art. 293. A pena de censura será aplicada reservadamente, por escrito, no caso de reiterada negligência no cumprimento dos deveres do cargo ou no de procedimento incorreto, se a infração não justificar punição mais grave.

Parágrafo único. O juiz punido com a pena de censura não poderá figurar em lista de promoção por merecimento pelo prazo de um ano, contado da imposição da pena.

Art. 294. A pena de advertência será aplicada reservadamente, por escrito, no caso de negligência no cumprimento dos deveres do cargo.

Art. 295. Sem prejuízo da competência do Conselho da Magistratura e da Corregedoria Geral de Justiça, as penas de advertência e de censura poderão ser aplicadas, quando verificada a hipótese do respectivo cabimento, em face do que constar de autos e papéis em curso no Tribunal ou submetidos a julgamento:

I – pelo Plenário do Tribunal ou por seu Presidente;

II – pelas Câmaras Reunidas ou por seus Presidentes;

III – pelas Câmaras Especializadas ou por seus Presidentes;

IV – pelos Relatores dos feitos.

Art. 296 - Quando as faltas disciplinares, imputáveis a juízes de Direito e passíveis das penalidades de advertência e de censura, não se apresentarem manifestas, na sua autoridade ou sua configuração, a apuração respectiva será feita pelo Conselho da Magistratura ou pela Corregedoria Geral da Justiça, cabendo ao órgão apurador a aplicação originária da penalidade.

#### **Seção IX - Da Declaração da Perda de Posto e de Patente de Oficiais e da Graduação das Praças**

(Seção IX e suas Subseções I a III acrescentadas pelo art. 2º da Resolução nº 020/2006, de 09/11/2006).

## **Subseção I - Das Disposições Gerais**

Art. 296-A. A declaração de indignidade ou incompatibilidade com o oficialato e a consequente perda do posto e patente e a perda de graduação das praças, nos casos previstos em lei, será proferida pelo Tribunal: [\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

I – mediante representação do Ministério Público nos casos de condenação criminal a pena superior a dois anos e por crimes que importem indignidade ou incompatibilidade com o oficialato, na forma da legislação militar;[\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

II – no julgamento de processo oriundo de Conselho de Justificação, de que trata a Subseção III, desta Seção. [\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

## **Subseção II - Da Representação do Ministério Público**

Art. 296-B. No caso de representação do Ministério Público, a que se refere o inciso I, do artigo 296-A, o militar será citado para, no prazo de dez dias, apresentar defesa escrita, através de seu defensor. [\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

§ 1º Decorrido o prazo previsto neste artigo, sem a apresentação da defesa escrita, o relator, previamente sorteado com o revisor, nomeará defensor dativo ou solicitará, ao órgão competente, a designação de Defensor Público, para que a apresente, em igual prazo. [\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

§ 2º O relator, após manifestação do revisor, colocará o processo, com a defesa escrita, em mesa para julgamento.[\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

§ 3º A votação será processada em sessão pública, facultada a sustentação oral pelo prazo máximo de vinte minutos.[\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

## **Subseção III - Do Processo de Conselho de Justificação**

Art. 296-C. Os autos do Conselho de Justificação serão autuados e distribuídos a relator e revisor. [\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

§ 1º O relator determinará a citação do justificante para que apresente defesa escrita no prazo de dez dias.[\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

§ 2º Decorrido o prazo previsto neste artigo, sem a apresentação da defesa escrita, o relator, previamente sorteado com o revisor, nomeará defensor dativo ou solicitará, ao órgão competente, a designação de Defensor Público, para que a apresente, em igual prazo.[\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

§ 3º Com as razões de defesa, os autos serão encaminhados para vista do Procurador-Geral de Justiça.[\(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006\)](#)

Art. 296-D. Anunciado o julgamento, proceder-se-á à leitura do relatório sendo facultado à defesa usar da palavra por, no máximo, vinte minutos. [\(Incluído pelo art. 2º](#)

da Resolução nº 020/2006, de 09/11/2006).

Art. 296-E. Discutida a matéria, em sessão pública, será proferida a decisão final. (Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006).

Art. 296-F. Decidido que o justificante é incapaz de permanecer na ativa ou na inatividade, deverá, o Tribunal, conforme o caso: (Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006).

I – declará-lo indigno do oficialato ou com ele incompatível, determinando a perda de seu posto e patente; ou (Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006)

II – determinar sua reforma.(Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006)

Parágrafo único. Após publicado o Acórdão no Diário da Justiça, será ele enviado, através de cópia, por intermédio do Comandante-Geral da Corporação, ao Governador do Estado, para os fins legais. (Incluído pelo art. 2º da Resolução nº 020/2006, de 09/11/2006)

## **CAPÍTULO XII - DOS PROCESSOS INCIDENTES**

### **Seção I - Das Arguições de Incompetência, de Suspeição e de Impedimento**

Art. 297. As arguições de incompetência absoluta ou relativa, de suspeição e de impedimento serão originariamente processadas e julgadas pelo Tribunal de Justiça:

I – quando opostas em feitos de competência originária do Tribunal;

II – quando, tratando-se de arguições de suspeição ou de impedimento opostas a desembargadores ou juízes de direito, os recusados não reconheçam a suspeição ou o impedimento, com a remessa do incidente ao Tribunal. (Redação dada pelo art. 33 da Resolução nº 06/2016, de 04/04/2016).

Art. 297-A. As arguições de incompetência absoluta ou relativa serão alegadas como questão preliminar de contestação. (Redação dada pelo art. 34 da Resolução nº 06/2016, de 04/04/2016).

§ 1º. A incompetência relativa pode ser alegada pelo Ministério Público na causa em que atuar. (Redação dada pelo art. 34 da Resolução nº 06/2016, de 04/04/2016).

§ 2º. Prorrogar-se-á a competência relativa se o réu não alegar a incompetência em preliminar de contestação. (Redação dada pelo art. 34 da Resolução nº 06/2016, de 04/04/2016).

§ 3º. A incompetência absoluta pode ser alegada em qualquer tempo e grau de jurisdição e deve ser declarada de ofício. (Redação dada pelo art. 34 da Resolução nº 06/2016, de 04/04/2016).

§ 4º. O relator decidirá imediatamente a questão de incompetência absoluta ou relativa após a manifestação da parte contrária. (Redação dada pelo art. 34 da Resolução nº 06/2016, de 04/04/2016).

§ 5º. Nos casos em que a incompetência absoluta estiver sendo decretada de ofício, serão ouvidas previamente todas as partes no prazo de 15 (quinze) dias, findo o qual o relator decidirá imediatamente a questão. (Redação dada pelo art. 34 da Resolução nº 06/2016, de 04/04/2016).

§ 6º. Acolhida a alegação de incompetência, os autos serão remetidos ao juízo

competente. (Redação dada pelo art. 34 da Resolução nº 06/2016, de 04/04/2016).

Art. 298. Salvo decisão judicial em sentido contrário, conservar-se-ão os efeitos da decisão proferida pelo juízo incompetente até que outra seja proferida, se for o caso, pelo juízo competente. (Redação dada pelo art. 35 da Resolução nº 06/2016, de 04/04/2016).

Art. 299. Arguida a incompetência de qualquer dos órgãos colegiados do Tribunal, para o julgamento da causa, o relator mandará processá-la na forma do art. 297-A deste regimento, após o que, dentro de igual prazo, apresentará o processo em mesa para julgamento. (Redação dada pelo art. 36 da Resolução nº 06/2016, de 04/04/2016).

Parágrafo único - (Revogado) (Redação dada pelo art. 85 da Resolução nº 06/2016, de 04 de abril de 2016).

Art. 300. O incidente de impedimento ou de suspeição de juiz de direito que não reconheceu o impedimento ou suspeição alegada pela parte com o fim de afastá-lo do processo, será autuado em apartado, acompanhado das razões apresentadas pelo magistrado para a recusa dessas alegações, dos documentos e rol de testemunhas, se houver, bem como do despacho ordenando a remessa dos autos do incidente ao Tribunal. (Redação dada pelo art. 37 da Resolução nº 06/2016, de 04/04/2016).

§ 1º. Distribuído o incidente, o relator deverá declarar os seus efeitos, sendo que, se o incidente for recebido: (Redação dada pelo art. 37 da Resolução nº 06/2016, de 04/04/2016).

I – sem efeito suspensivo, o processo voltará a correr; (Redação dada pelo art. 37 da Resolução nº 06/2016, de 04/04/2016).

II – com efeito suspensivo, o processo permanecerá suspenso até o julgamento do incidente. (Redação dada pelo art. 37 da Resolução nº 06/2016, de 04/04/2016).

§ 2º. Enquanto não for declarado o efeito em que é recebido o incidente, ou quando este for recebido com efeito suspensivo, a tutela de urgência será requerida ao substituto legal. (Redação dada pelo art. 37 da Resolução nº 06/2016, de 04/04/2016).

§ 3º. Se houver rol de testemunhas, o relator marcará dia e hora e intimará as partes do incidente para a audiência de inquirição das testemunhas arroladas, após o que, ouvido o Ministério Público, apresentará o processo em mesa para julgamento. (Redação dada pelo art. 37 da Resolução nº 06/2016, de 04/04/2016).

§ 4º. Verificando que a alegação de impedimento ou de suspeição é improcedente, o Tribunal rejeita-la-á. (Redação dada pelo art. 37 da Resolução nº 06/2016, de 04/04/2016).

§ 5º. Reconhecido o impedimento ou a suspeição, o Tribunal fixará o momento a partir do qual o juiz não poderia ter atuado. (Redação dada pelo art. 37 da Resolução nº 06/2016, de 04/04/2016).

§ 6º. O Tribunal decretará a nulidade dos atos do juiz, se praticados quando já presente o motivo de impedimento ou de suspeição. (Redação dada pelo art. 37 da Resolução nº 06/2016, de 04/04/2016).

§ 7º. Acolhida a alegação, tratando-se de impedimento ou de manifesta suspeição, o Tribunal condenará o juiz nas custas e remeterá os autos ao seu substituto legal, podendo o juiz recorrer da decisão. (Redação dada pelo art. 37 da Resolução nº 06/2016, de 04/04/2016).

Art. 301. (Revogado) (Redação dada pelo art. 85 da Resolução nº 06/2016, de

04/04/2016).

§ 1º. (Revogado) (Redação dada pelo art. 85 da Resolução nº 06/2016, de 04/04/2016).

Art. 302. No prazo de 15 (quinze) dias, a contar do conhecimento do fato, a parte alegará o impedimento ou a suspeição do relator do processo ou de qualquer outro desembargador que participe do julgamento em petição específica que lhe for dirigida, na qual indicará o fundamento de sua recusa para a causa, podendo instruí-la com documentos em que se funda a alegação e com rol de testemunhas. (Redação dada pelo art. 38 da Resolução nº 06/2016, de 04/04/2016).

§ 1º. Se reconhecer o impedimento ou a suspeição, o relator ordenará imediatamente a remessa dos autos à secretaria para nova distribuição do processo, na forma regimental; caso contrário, determinará autuação em apartado da petição e, no prazo de 15 (quinze) dias, apresentará suas razões, acompanhadas de documentos e de rol de testemunhas, se houver, ordenando, por despacho, a remessa do incidente ao presidente do Tribunal para relatar o feito. (Redação dada pelo art. 38 da Resolução nº 06/2016, de 04/04/2016).

§ 2º. Recebido o incidente, o presidente do Tribunal procederá na forma do art. 300, §§ 1º, 2º e 3º, deste Regimento. (Redação dada pelo art. 38 da Resolução nº 06/2016, de 04/04/2016).

§ 3º. No julgamento do incidente de impedimento ou suspeição do relator do processo, o Tribunal procederá na forma do art. 300, §§ 5º, 6º e 7º deste Regimento. (Redação dada pelo art. 38 da Resolução nº 06/2016, de 04/04/2016).

§ 4º. Na hipótese de incidente de impedimento ou suspeição de outro desembargador que integre o órgão colegiado competente para o julgamento do processo, o relator ouvirá o julgador a quem se imputa a parcialidade, o qual, se a reconhecer, não participará do julgamento; porém, se a negar, apresentará as suas razões no prazo de 15 (quinze) dias. (Redação dada pelo art. 38 da Resolução nº 06/2016, de 04/04/2016).

Art. 303. A arguição de suspeição ou impedimento de Desembargador será sempre individual, não ficando os demais Desembargadores impedidos de apreciá-la, ainda que também recusados.

Art. 304. Não se fornecerá, salvo ao arguente e ao arguido, certidão de qualquer peça do processo de suspeição, antes de afirmada pelo arguido ou declarada pelo Tribunal.

Parágrafo único - Da certidão constará obrigatoriamente o nome de quem a requereu, bem assim o desfecho que houver tido a arguição.

Art. 304-A. Aplicam-se os motivos de impedimento e de suspeição: (Redação dada pelo art. 39 da Resolução nº 06/2016, de 04/04/2016).

I – ao membro do Ministério Público; (Redação dada pelo art. 39 da Resolução nº 06/2016, de 04/04/2016).

II – aos auxiliares da Justiça; (Redação dada pelo art. 39 da Resolução nº 06/2016, de 04/04/2016).

III – aos demais sujeitos imparciais do processo. (Redação dada pelo art. 39 da Resolução nº 06/2016, de 04/04/2016).

§ 1º. A parte interessada deverá arguir o impedimento ou a suspeição, em petição fundamentada e devidamente instruída, na primeira oportunidade em que lhe couber falar nos autos. (Redação dada pelo art. 39 da Resolução nº 06/2016, de 04/04/2016).

§ 2º. O Relator mandará processar o incidente em separado e sem suspensão do processo, ouvindo o arguido no prazo de 15 (quinze) dias e facultando a produção de

prova quando necessária. (Redação dada pelo art. 39 da Resolução nº 06/2016, de 04/04/2016).

§ 3º. No processamento e julgamento do incidente de impedimento ou de suspeição do *caput* deste artigo, aplicar-se-á o disposto no art. 300, §§ 3º a 7º, deste Regimento, no que couber. (Redação dada pelo art. 39 da Resolução nº 06/2016, de 04/04/2016).

## **Seção II - Da Habilitação Incidente**

Art. 305. Ocorrendo o falecimento de alguma das partes e estando a causa em curso no Tribunal de Justiça, a habilitação dos interessados que houverem de lhe suceder será processada perante o respectivo Relator.

Art. 306. Proceder-se-á à habilitação no processo principal, suspendendo-se, a partir de então, o processo. (Redação dada pelo art. 40 da Resolução nº 06/2016, de 04/04/2016).

Art. 307. Em caso de morte de alguma das partes:

I – o cônjuge, herdeiro ou legatário requererá sua habilitação, bem como a citação da outra parte para contestá-la;

II – a parte poderá requerer a habilitação dos sucessores do falecido;

III – qualquer interessado poderá requerer a citação do cônjuge, herdeiro ou legatário para providenciar sua habilitação em quinze dias.

§ 1º Recebida a petição inicial, ordenará o Relator a citação dos requeridos para contestar a ação no prazo de cinco dias.

§ 2º No caso de inciso III deste artigo, se a parte não providenciar a habilitação, o processo correrá à revelia.

§ 3º Na hipótese do parágrafo anterior, será nomeado curador ao revel, oficiando também o Procurador Geral da Justiça.

Art. 308. A citação será feita na pessoa do Procurador constituído nos autos, mediante publicação no órgão oficial, ou à parte, pessoalmente, se não estiver representada no processo.

Art. 309. Quando incertos os sucessores, a citação será feita por edital.

Art. 310. O cessionário ou sub-rogado poderá habilitar-se, apresentando o documento da cessão ou sub-rogação e pedindo a citação dos interessados.

Parágrafo único. O cessionário de herdeiro somente após a habilitação deste poderá se apresentar.

Art. 311. O relator decidirá o pedido de habilitação imediatamente nos autos do processo principal, salvo se este for impugnado e houver necessidade de dilação probatória diversa da documental, caso em que determinará que o pedido seja autuado em apartado e disporá sobre a instrução. (Redação dada pelo art. 41 da Resolução nº 06/2016, de 04/04/2016).

§ 1º. No primeiro caso, a habilitação será julgada por decisão interlocutória, e, no segundo, por sentença, finda a instrução do incidente. (Redação dada pelo art. 41 da Resolução nº 06/2016, de 04/04/2016).

§ 2º. Mesmo na ausência de impugnação, caberá ao relator, de ofício, determinar a produção das provas necessárias ao julgamento da habilitação, se não estiver convencido do preenchimento dos requisitos da habilitação. (Redação dada pelo art. 41 da Resolução nº 06/2016, de 04/04/2016).



Art. 312. Dependerá de decisão do relator, mesmo quando processado nos autos da causa principal, o pedido de habilitação. (Redação dada pelo art. 42 da Resolução nº 06/2016, de 04/04/2016).

I – do cônjuge e herdeiros necessários que provem por documento sua qualidade e o óbito do falecido;

II – fundado em sentença, com o trânsito em julgado, que atribuam ao requerente a qualidade de herdeiro ou sucessor;

III – do herdeiro que for incluído sem qualquer oposição no inventário;

IV – quando estiver declarada a ausência ou determinada a arrecadação da herança jacente;

V – quando, oferecidos os artigos de habilitação, a parte reconhecer a procedência do pedido e não houver oposição de terceiro.

Art. 313. O cessionário ou o adquirente podem prosseguir na causa juntando aos autos o respectivo título e provando a sua identidade, caso em que sucederão ao cedente ou ao credor originário que houverem falecido.

Art. 314. Já havendo pedido de dia de julgamento, não se decidirá o requerimento de pedido de habilitação de pessoa jurídica de direito público ou de direito privado que notoriamente seja conhecida como sucessora de outra. (Redação dada pelo art. 43 da Resolução nº 06/2016, de 04/04/2016).

Art. 315. Proceder-se-á à habilitação na instância em que estiverem os autos do processo principal. (Redação dada pelo art. 44 da Resolução nº 06/2016, de 04/04/2016).

Art. 316. Transitada em julgado a sentença de habilitação, cujo pedido foi autuado em apartado, cópia da sentença será juntada aos autos da ação principal. (Redação dada pelo art. 45 da Resolução nº 06/2016, de 04/04/2016).

Parágrafo único. Em qualquer hipótese, o processo principal retornará ao seu curso quando esgotados todos os prazos recursais, seja contra a sentença ou a decisão interlocutória de habilitação. (Redação dada pelo art. 45 da Resolução nº 06/2016, de 04/04/2016).

### **Seção III – Do Incidente de Arguição de Inconstitucionalidade**

Art. 317. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Art. 318. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Art. 319. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Parágrafo único. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Art. 320. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Art. 321. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Art. 322. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de

15/09/2016).

**Seção IV – Do Incidente de Assunção de Competência**  
(Revogado pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Art. 323. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

§ 1º. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

§ 2º. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

§ 3º. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

§ 4º. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Art. 324. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016)..

Art. 325. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Art. 326. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Parágrafo único. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

**Seção V - Da Suspensão de Liminar em Mandado de Segurança e da Suspensão Provisória de Execução de Sentença**

Art. 327. O Presidente do Tribunal de Justiça, a requerimento de pessoa jurídica pública interessada e para evitar grave lesão à ordem, à saúde, à segurança e à economia públicas, poderá suspender, em despacho fundamentado, a execução da liminar deferida por juiz de Direito, em processo de mandado de segurança, e, bem assim, a execução de sentença concessiva da segurança, enquanto não confirmada pela superior instância.

Art. 328. Sobre o pedido de suspensão a que se refere o artigo anterior, serão ouvidos as partes e o Ministério Público. (Redação dada pelo art. 51 da Resolução nº 06/2016, de 04/04/2016).

Art. 329. Do despacho do Presidente, ordenador da suspensão, caberá agravo para o Plenário do Tribunal, no prazo de dez dias.

**Seção VI - Do Desaforamento de Julgamento**

Art. 330. Nos processos de competência do Júri, poderão as Câmaras Reunidas Criminais, a requerimento de quaisquer das partes ou mediante representação do juiz de Direito, desaforar o julgamento se o interesse da ordem pública o reclamar, ou houver dúvida sobre a imparcialidade do júri, ou sobre a segurança pessoal do réu.

Art. 331. Autuado o pedido ou a representação, serão solicitadas informações ao juiz local, se a medida não tiver sido solicitada, de ofício, por ele próprio, e, em seguida, será ouvida a Procuradoria Geral da Justiça.

Art. 332. As Câmaras Reunidas Criminais, na hipótese de acolherem o pedido ou a representação, designarão para o julgamento, comarca próxima, onde não subsistam os

motivos determinantes do desaforamento.

Art. 333. As Câmaras Reunidas Criminais poderão ainda, a requerimento do réu ou do Ministério Público, determinar o desaforamento, se o julgamento não se realizar no período de um ano, contado do recebimento do libelo, desde que, para a demora, não haja concorrido o réu ou a defesa.

### **Seção VII - Da Impugnação ao Valor da Causa**

Art. 334. Nas causas cíveis da competência originária do Tribunal de Justiça, o réu poderá impugnar, em preliminar de contestação, o valor atribuído à causa pelo autor, sob pena de preclusão, e o relator decidirá a respeito, impondo, se for o caso, a complementação das custas, ou, ainda, a devolução delas se não forem devidas, ou se forem pagas em excesso. (Redação dada pelo art. 52 da Resolução nº 06/2016, de 04/04/2016).

Art. 335. O autor será ouvido no prazo de 15 (quinze) dias e, em seguida, o relator, sem suspender o processo, servindo-se, quando necessário, do auxílio de perito, determinará, em igual prazo, o valor da causa. (Redação dada pelo art. 53 da Resolução nº 06/2016, de 04/04/2016).

Parágrafo único. Da decisão do Relator caberá agravo regimental para as Câmaras Especializadas Cíveis, para as respectivas Câmaras Reunidas ou para o Plenário conforme o caso.

Art. 335-A. O relator corrigirá, de ofício, e por arbitramento, o valor da causa, quando verificar que não corresponde ao conteúdo patrimonial em discussão ou ao proveito econômico perseguido pelo autor, caso em que procederá ao recolhimento das custas correspondentes. (Redação dada pelo art. 54 da Resolução nº 06/2016, de 04/04/2016).

Parágrafo único. Antes da decisão, o relator ouvirá as partes, no prazo de 15 (quinze) dias. (Redação dada pelo art. 54 da Resolução nº 06/2016, de 04/04/2016).

Art. 335-B. Da decisão do relator caberá, em qualquer caso, agravo interno para o órgão competente. (Redação dada pelo art. 54 da Resolução nº 06/2016, de 04/04/2016).

### **Seção VIII - Do Incidente de Falsidade**

Art. 336. Arguida, por escrito, a falsidade de documento constante dos autos de processo penal, em curso no Tribunal de Justiça, o Relator observará o seguinte:

I – mandará autuar em apartado a impugnação e, em seguida, ouvirá a parte contrária, que, no prazo de quarenta e oito horas, poderá oferecer resposta;

II – assinará o prazo de três dias, sucessivamente, a cada uma das partes para prova de suas alegações;

III – conclusos os autos, poderá ordenar as diligências que entender necessárias;

IV – submeterá o incidente, para julgamento preliminar, quando apreciada a ação originária ou o recurso, pelo órgão julgante competente.

§ 1º Reconhecida a falsidade por decisão irrecurável, o Relator mandará desentranhar o documento e remetê-lo, com os autos do processo incidente, ao Ministério Público.

§ 2º A arguição de falsidade, feita por procurador, exige poderes especiais, ressalvada a

hipótese de procuração com a cláusula *ad juditia*.

§ 3º O Relator poderá, de ofício, proceder à verificação da falsidade.

Art. 337. No processo cível, a falsidade deve ser suscitada na contestação, na réplica ou no prazo de 15 (quinze) dias, contado a partir da intimação da juntada do documento aos autos. (Redação dada pelo art. 55 da Resolução nº 06/2016, de 04/04/2016).

Parágrafo único. Uma vez arguida, a falsidade será resolvida como questão incidental, salvo se a parte requerer que o juiz a decida como questão principal, nos termos do inciso II do art. 19 do Código de Processo Civil. (Redação dada pelo art. 55 da Resolução nº 06/2016, de 04/04/2016).

Art. 337-A. Suscitado o incidente de falsidade, na forma do artigo anterior, o relator, depois de ouvida a outra parte no prazo de 15 (quinze) dias, determinará a realização do exame pericial, salvo se a parte que produziu o documento concordar em retirá-lo. (Redação dada pelo art. 56 da Resolução nº 06/2016, de 04/04/2016).

Parágrafo único. O Tribunal Pleno, as Câmaras Reunidas ou as Câmaras Especializadas, conforme o caso, apreciando o incidente, declararão a falsidade ou a autenticidade do documento. (Redação dada pelo art. 56 da Resolução nº 06/2016, de 04/04/2016).

Art. 337-B. A declaração sobre a falsidade do documento, quando suscitada como questão principal, constará da parte dispositiva do acórdão e sobre ela incidirá também a autoridade da coisa julgada. (Redação dada pelo art. 56 da Resolução nº 06/2016, de 04/04/2016).

### **Seção IX - Da Restauração de Autos**

Art. 338. A restauração dos autos originais de processo penal extraviados ou destruídos será feita na primeira instância, ainda que o extravio, ou a destruição, tenha ocorrido na segunda instância.

Art. 339. Os autos originais de processo cível, desaparecidos no Tribunal de Justiça, terão a respectiva restauração promovida na segunda instância.

§ 1º A ação será distribuída, sempre que possível, ao Relator do Processo.

§ 2º A restauração será feita no juízo de origem quanto aos autos que neste se tenham realizado.

§ 3º Remetidos os autos ao Tribunal, aí se completará a restauração e se procederá ao julgamento.

§ 4º Na restauração dos autos serão observados as disposições das leis processuais em vigor.

### **Seção X - Da Reclamação**

Art. 340. Caberá reclamação da parte interessada ou do Ministério Público, dirigida ao presidente do Tribunal, nos casos previstos em lei. (Redação dada pelo art. 57 da Resolução nº 06/2016, de 04/04/2016).

Art. 341. A reclamação será instruída com prova documental.

Art. 341-A. Assim que recebida, a reclamação será autuada e distribuída ao relator do processo principal, sempre que possível. (Redação dada pelo art. 58 da Resolução nº 06/2016, de 04/04/2016).

Art. 342. Ao despachar a reclamação, o relator: (Redação dada pelo art. 59 da Resolução nº 06/2016, de 04/04/2016).

I – Requisitará informações da autoridade a quem for imputada a prática do ato impugnado, que as prestará no prazo de 10 (dez) dias; (Redação dada pelo art. 59 da Resolução nº 06/2016, de 04/04/2016).

II – Se necessário, ordenará a suspensão do processo ou do ato impugnado para evitar dano irreparável; (Redação dada pelo art. 59 da Resolução nº 06/2016, de 04/04/2016).

III – determinará a citação do beneficiário da decisão impugnada, que, terá prazo de 15 (quinze) dias para apresentar sua contestação. (Redação dada pelo art. 59 da Resolução nº 06/2016, de 04/04/2016).

IV – requisitará a remessa dos autos do processo principal ao Tribunal, se necessário. (Redação dada pelo art. 59 da Resolução nº 06/2016, de 04/04/2016).

Art. 343. (Revogado) (Redação dada pelo art. 85 da Resolução nº 06/2016, de 04/04/2016)

Art. 344. Qualquer interessado poderá impugnar o pedido do reclamante.

Art. 345. Decorrido o prazo para informações, será concedida vista à Procuradoria Geral da Justiça, quando a reclamação não tenha sido por ela formulada.

Art. 346. Julgada procedente a reclamação, o Plenário poderá:

I – avocar o conhecimento do processo em que se verifique usurpação de sua competência;

II – ordenar que lhe sejam remetidos, com urgência, os autos do recurso para ele interposto;

III – cassar decisão exorbitante de seu julgado, ou determinar medida adequada à observância de sua jurisdição.

Art. 346-A. O julgamento da reclamação compete ao órgão jurisdicional cuja competência se busca preservar ou cuja autoridade se pretenda garantir. (Redação dada pelo art. 60 da Resolução nº 06/2016, de 04/04/2016).

Art. 347. O presidente determinará o imediato cumprimento da decisão, lavrando-se o acórdão posteriormente.

### **Seção XI – Do Incidente de Resolução de Demandas Repetitivas**

(Redação dada pelo art. 61 da Resolução nº 06/2016, de 04/04/2016).

Art. 347-A.(Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

Art. 347-B. (Revogado) (Redação dada pelo art. 3º da Resolução nº 21/2016, de 15/09/2016).

## **CAPÍTULO XII-A PROCEDIMENTOS DE UNIFORMIZAÇÃO DE JURISPRUDÊNCIA**

(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

### **Seção I - Disposições Gerais**

(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

Art. 347-C - O Tribunal deve uniformizar a sua jurisprudência e mantê-la estável, íntegra e coerente. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 1º - A uniformização de jurisprudência neste Tribunal pode ser o resultado de um destes procedimentos: (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

I - incidente de resolução de demandas repetitivas; (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

II - incidente de assunção de competência; (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

III - incidente de arguição de inconstitucionalidade de lei ou ato normativo. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 2º - Não caberá recurso contra a decisão que admitir a instauração dos incidentes previstos no § 1º deste artigo. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

Art. 347-D - A jurisprudência firmada pelo Tribunal será compendiada em Súmula. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 1º - Será objeto de súmula a tese jurídica firmada no julgamento do respectivo Órgão Julgador, competente de acordo com este Regimento, tomada pelo voto da maioria absoluta de seus membros efetivos, no incidente de resolução de demandas repetitivas, no incidente de assunção de competência e no incidente de arguição de inconstitucionalidade de lei ou ato normativo. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 2º - Também poderão ser objeto de súmula as teses jurídicas correspondentes às decisões firmadas por unanimidade dos membros efetivos do Tribunal, no julgamento de questões administrativas. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 3º - Ao editar enunciados de súmula, o Tribunal deve ater-se às circunstâncias fáticas dos precedentes que motivaram sua criação. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

Seção II - Do Incidente de Resolução de Demandas Repetitivas e do Incidente de Assunção de Competência. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

Art. 347-E - O incidente de resolução de demandas repetitivas e o incidente de assunção de competência serão processados de acordo com as normas decorrentes do Código de Processo Civil e deste Regimento. Parágrafo único. O incidente de resolução de demandas repetitivas e o incidente de assunção de competência têm por objeto a solução de questão de direito material ou processual. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

Art. 347-F - O incidente de resolução de demandas repetitivas tem por objeto a solução de questão de direito que se repita em diversos processos individuais ou coletivos, quando houver risco de ofensa à isonomia e à segurança jurídica. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 1º - O incidente será instaurado a partir de pedido dirigido ao Presidente do Tribunal de Justiça do Estado do Piauí, por ofício ou petição, na forma do art. 977 do Código de Processo Civil, que determinará a sua devida autuação em decisão publicada no Diário da Justiça eletrônico para ciência das partes. (Incluído pelo art. 2º da Resolução nº

21/2016, de 15/09/2016).

§ 2º - Se houver mais de um pedido de instauração de incidente, tendo por objeto a mesma questão de direito, o Presidente do Tribunal escolherá o caso que mais bem represente a controvérsia, observado o disposto no § 6º do art. 1.036 do Código de Processo Civil, e determinará que os demais pedidos integrem a autuação a fim de que o Relator conheça dos argumentos levantados; os requerentes dos pedidos não escolhidos serão informados do número do incidente instaurado e as partes dos respectivos casos poderão participar do processo como intervenientes. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 3º - Determinada a autuação e distribuição do pedido selecionado, novos pedidos dirigidos ao Presidente envolvendo a mesma questão de direito serão rejeitados e devolvidos ao remetente com a informação de que já foi instaurado incidente sobre o tema e seu respectivo número a fim de que postulem eventual intervenção. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 4º - O incidente será distribuído por prevenção ao Relator do recurso, remessa necessária ou processo de competência originária do Tribunal do qual se originou ou, caso não integre o órgão competente para julgamento do incidente, por sorteio entre os seus membros efetivos. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 5º - Caso o incidente tenha sido suscitado no bojo de recurso, remessa necessária ou processo de competência originária do Tribunal, os autos deverão ser apensados em atenção ao disposto no parágrafo único do art. 978 do CPC. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 6º - Distribuído o incidente, o Relator submeterá o exame da sua admissibilidade ao órgão colegiado competente para julgá-lo na forma deste Regimento. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 7º - Caberá sustentação oral na sessão de julgamento de resolução de demandas repetitivas, observado o regramento do art. 984 do Código de Processo Civil. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 8º - Inadmitido o incidente e lavrado o respectivo acórdão, os autos permanecerão arquivados no Tribunal. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 9º - Admitido o incidente de resolução de demandas repetitivas pelo órgão colegiado, retornarão os autos conclusos ao Relator, que proferirá decisão na qual: [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

I - identificará, com precisão, a questão a ser submetida a julgamento; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

II - identificará as circunstâncias fáticas que ensejam a controvérsia em torno da questão jurídica; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

III - apresentará o índice com os fundamentos, acerca da questão jurídica, apresentados até o momento da admissão, inclusive os que constem de manifestações utilizadas para fins de instruir o pedido ou ofício de instauração, e com os dispositivos normativos relacionados à controvérsia; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

IV - determinará a suspensão do trâmite dos processos, individuais e coletivos, na primeira instância ou no Tribunal, em que se discuta a questão jurídica objeto do incidente; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

V - poderá requisitar informações sobre o objeto do incidente aos órgãos em que tramitem processos, judiciais ou administrativos, nos quais se discuta a questão objeto do incidente; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).

VI - determinará a intimação do Ministério Público para que participe do incidente, salvo quando já figurar como requerente; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).

VII - caso a questão objeto do incidente seja relativa à prestação de serviço concedido, permitido ou autorizado, comunicará ao ente público ou à agência reguladora competente para, querendo, participar do incidente, prestando informações; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).

VIII - determinará a inclusão do incidente no Cadastro de Incidentes do Tribunal e comunicará ao Conselho Nacional de Justiça a sua instauração para fim de inclusão, no Cadastro Nacional, das informações constantes dos incisos I a III do § 8º; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).

X - organizará a instrução do incidente, podendo, inclusive, designar audiência pública, nos termos deste Regimento. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).

§ 10º - A suspensão determinada deverá ser comunicada, via ofício e por meio eletrônico, aos órgãos jurisdicionais vinculados ao Tribunal e aos juizados especiais no âmbito do Estado do Piauí, bem como à Comissão que compõe o Núcleo de Gerenciamento de Precedentes (Nugep). [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).

§ 11 - As partes dos processos repetitivos deverão ser intimadas da decisão de suspensão de seus processos, a ser proferida pelo respectivo juiz ou Relator, quando informados acerca da suspensão a que se refere o inciso IV do § 8º. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).

§ 12 - O Tribunal atualizará o seu cadastro eletrônico para incluir informações relativas ao ingresso de amicus curiae, designação de audiências públicas e outras informações relevantes para a instrução e o julgamento do incidente; logo em seguida, os novos dados serão informados ao Conselho Nacional de Justiça para que proceda às alterações no cadastro nacional. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).

§ 13 - Além dos cadastros a que se refere o art. 979 do Código de Processo Civil, o Tribunal manterá os autos do incidente disponíveis para consulta pública na rede mundial de computadores. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).

Art. 347-G - O incidente de assunção de competência tem por objeto a solução de relevante questão de direito com grande repercussão social, jurídica, econômica ou política, sem repetição em múltiplos processos, a respeito da qual seja conveniente a prevenção ou a composição de divergência entre as Câmaras do Tribunal. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).

§ 1º - O Relator, de ofício ou a requerimento da parte, do Ministério Público ou da Defensoria Pública, proporá, ao órgão a que se encontre vinculado, que o recurso, a remessa necessária ou o processo de competência originária do Tribunal seja julgado pelo Tribunal Pleno, observadas as competências definidas neste Regimento. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\)](#).



§ 2º - Rejeitada a proposta pelo colegiado, será lavrado acórdão pelo Desembargador que proferir o primeiro voto divergente e os autos retornarão conclusos ao Relator originário para prosseguimento; aceita a proposta pelo colegiado, será lavrado acórdão nos autos e extraída cópia que, instruída pelo Relator com os elementos necessários à exposição da questão de direito e demonstração da sua relevância, será devidamente autuada e distribuída. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 3º - O incidente será apensado aos autos em que suscitado e ambos serão distribuídos por prevenção ao órgão e Relator originários. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 4º - Distribuído o incidente, o Relator submeterá o exame da sua admissibilidade ao Pleno do Tribunal. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 5º Caberá sustentação oral na sessão de julgamento do incidente de assunção de competência, observado o regramento do art. 984 do Código de Processo Civil. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 6º - Inadmitido o incidente e lavrado o respectivo acórdão, os autos do incidente permanecerão arquivados no Tribunal de Justiça e os do processo em que suscitado retornarão ao Relator no Órgão de origem. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 7º - Admitido o incidente de assunção de competência pelo Órgão colegiado, o Relator proferirá decisão em que: [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

I. identificará, com precisão, a questão a ser submetida a julgamento; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

II. identificará as circunstâncias fáticas que ensejam a controvérsia em torno da questão jurídica; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

III. apresentará o índice com os fundamentos, acerca da questão jurídica, apresentados até o momento da admissão, inclusive os que constem de manifestações utilizadas para fundamentar o pedido de instauração, e com os dispositivos normativos relacionados à controvérsia; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

IV. determinará a intimação do Ministério Público para que participe do incidente; [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

V. organizará a instrução do incidente, inclusive com a marcação de audiência pública, nos termos deste Regimento. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 8º - O Tribunal organizará o cadastro eletrônico dos incidentes de assunção de competência, a ser divulgado na rede mundial de computadores, observando-se o disposto no art. 979 do Código de Processo Civil. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 9º - O Tribunal manterá os autos do incidente disponíveis para consulta pública na rede mundial de computadores. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

Art. 347-H - Concluída a instrução, o Relator solicitará dia para julgamento do incidente, respeitado o prazo mínimo de 20 (vinte) dias entre a realização da sessão de julgamento e a publicação da pauta e inserção da informação nos cadastros a que se refere o art. 979 do Código de Processo Civil. [\(Incluído pelo art. 2º da Resolução nº](#)

21/2016, de 15/09/2016).

§ 1º - O Relator do incidente de resolução de demandas repetitivas ou do incidente de assunção de competência poderá, de comum acordo com todos os sujeitos do incidente, definir o calendário de instrução e julgamento, nos termos do art. 191 do Código de Processo Civil. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 2º - Cabe sustentação oral na sessão de julgamento do incidente de resolução de demandas repetitivas e do incidente de assunção de competência, observado, em ambos os casos, o art. 984 do Código de Processo Civil. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

Art. 347-I - São elementos essenciais do acórdão que julgar o incidente de resolução de demandas repetitivas ou o incidente de assunção de competência: (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

I. o relatório, que conterà os nomes das partes, a identificação do caso, com a suma do pedido e da contestação, e o registro das principais ocorrências havidas no andamento do processo; (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

II. a identificação das circunstâncias fáticas que ensejam a controvérsia em torno da questão jurídica; (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

III. o índice com todos os fundamentos favoráveis e contrários à tese jurídica discutida; (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

IV. a análise de todos os fundamentos contrários e favoráveis à tese jurídica discutida; (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

V. os dispositivos normativos relacionados à questão discutida; (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

VI. a enunciação da tese jurídica objeto do incidente; (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

VII. a fundamentação para a solução do caso; (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

VIII. o dispositivo, em que o Tribunal resolverá o caso que lhe foi submetido. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 1º - Se houver desistência ou abandono da causa, nos termos do art. 976, § 1º, do Código de Processo Civil, os elementos do acórdão serão apenas aqueles previstos nos incisos I a VI do caput deste artigo. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 2º - O incidente de resolução de demandas repetitivas suscitado por Juiz de Direito somente será admitido se houver, no Tribunal, processo de competência originária, remessa necessária ou recurso que verse sobre a questão de direito repetitiva, que será selecionado como representativo da controvérsia. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 3º - O Relator deverá, na sessão de julgamento, enunciar a tese jurídica objeto do incidente, o que constará da ata de julgamento. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 4º - Na enunciação da tese jurídica objeto do incidente, o Tribunal observará: - o fundamento determinante adotado pela unanimidade ou maioria dos membros do Órgão Julgador; - o disposto no art. 926, § 2º, do Código de Processo Civil. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 5º - A sessão de julgamento deverá ser integralmente registrada mediante gravação de áudio e vídeo, sempre que possível. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

Art. 347-J - O acórdão que julgar o incidente de resolução de demandas repetitivas ou o incidente de assunção de competência vinculará todos os órgãos jurisdicionais de primeira e segunda instância da área de jurisdição do Tribunal de Justiça, inclusive no âmbito dos Juizados Especiais Cíveis. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

Art. 347-K - O acórdão que inadmite a instauração de incidente de resolução de demandas repetitivas ou de incidente de assunção de incompetência é irrecorrível. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

Art. 347-L - O redator do acórdão que julgou o incidente de resolução de demandas repetitivas ou o incidente de assunção de competência é prevento para processar e julgar futuros incidentes em que se discuta a mesma questão jurídica, observado o Regimento Interno do TJPI na hipótese de necessária substituição do Desembargador prevento. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

Art. 347-M- A revisão da tese jurídica firmada no julgamento do incidente de resolução de demandas repetitivas ou do incidente de assunção de competência dar-se-á após instauração de novo incidente, observado o disposto nos §§ 2º, 3º e 4º do art. 927 do Código de Processo Civil. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 1º - Admitida a instauração do incidente-revisor, o Tribunal deverá registrar a informação no seu cadastro eletrônico, inserindo a informação no registro do incidente em que houver sido fixada a tese; logo em seguida, os novos dados serão informados ao Conselho Nacional de Justiça para que proceda ao registro no cadastro nacional. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 2º - O Relator do incidente-revisor deverá intimar os sujeitos do incidente em que tenha ocorrido a fixação da tese para que, querendo, manifestem-se no incidente-revisor. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 3º - Caso a tese jurídica seja revisada, o acórdão que julgar o incidente deverá conter todas as informações previstas no art. 347-I deste Regimento e, ainda, indicar expressamente os parâmetros para modulação temporal da eficácia da decisão revisora. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 4º A revisão da tese jurídica impõe que enunciado de súmula anteriormente editado a partir da sua consolidação seja revisto ou cancelado e, se for o caso, editado enunciado a partir da nova tese jurídica. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

### **Seção III - Do Incidente de Arguição de Inconstitucionalidade de Lei ou Ato Normativo**

[\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

Art. 347-N- Arguida, em controle difuso, a inconstitucionalidade de lei ou de ato normativo do poder público, a questão será submetida ao Órgão Julgador competente na forma deste Regimento, em atenção ao art. 97 da Constituição Federal, salvo quando já

houver pronunciamento do plenário do próprio Tribunal ou do Supremo Tribunal Federal sobre a questão. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 1º - O Relator, de ofício ou a requerimento, após ouvir o Ministério Público e as partes, submeterá a questão à turma ou à câmara à qual competir o conhecimento do processo. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 2º - Caberá sustentação oral na sessão de julgamento o Incidente de Arguição de Inconstitucionalidade de Lei ou Ato Normativo, observado o regramento do art. 984 do Código de Processo Civil. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 3º - Rejeitada a arguição de inconstitucionalidade pelo colegiado, os autos retornarão conclusos ao Relator para prosseguimento; acolhida a arguição pelo colegiado, será lavrado acórdão nos autos e extraída cópia que, instruída com os elementos necessários à demonstração da controvérsia, formará o incidente a ser devidamente autuado e distribuído. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 4º - O incidente será apensado aos autos em que suscitado e ambos serão distribuídos por prevenção ao órgão e Relator originários. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 5º - Os autos em que suscitado o incidente permanecerão na Secretaria do órgão fracionário competente para o conhecimento do recurso, remessa necessária ou ação de competência originária, mantendo-se o seu trâmite suspenso enquanto se aguarda o julgamento do incidente de arguição de inconstitucionalidade. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

Art. 347-O - O Relator mandará ouvir o Procurador-Geral de Justiça, no prazo de 15 (quinze) dias, bem como determinará a notificação da pessoa jurídica de direito público responsável pela edição do ato questionado para que se manifeste no prazo de 15 (quinze) dias. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 1º - O Tribunal dará publicidade à instauração do incidente de arguição de inconstitucionalidade a fim de permitir eventual intervenção dos legitimados referidos no art. 103 da Constituição Federal, como autoriza o art. 950, § 2º, do Código de Processo Civil, ou de outros órgãos ou entidades, na condição de amicus curiae, mediante inclusão em cadastro de incidentes instaurados disponível na sua página na rede mundial de computadores. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 2º - As intervenções previstas no § 1º serão permitidas dentro do período de 30 (trinta) dias, contados da publicação da decisão prevista no caput, que deverá indicar a lei ou o ato normativo objeto do incidente e a possibilidade de intervenção. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 3º - Encerrada a instrução do incidente, o Relator lançará relatório nos autos, determinando a distribuição de cópias deste, do acórdão que acolheu a arguição de inconstitucionalidade e do parecer do Ministério Público aos demais componentes do órgão julgador, com antecedência de 5 (cinco) dias da sessão de julgamento. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 4º - Cabe sustentação oral na sessão de julgamento do incidente de arguição de inconstitucionalidade, observado o regramento do art. 984 do Código de Processo Civil. (Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016).

§ 5º - Julgado o incidente, lavrado e publicado o respectivo acórdão, os autos permanecerão arquivados junto ao setor competente, procedendo-se ao registro da súmula do julgamento no cadastro indicado no § 1º e ao traslado de cópia do acórdão para os autos do feito originário. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

§ 6º - Certificado o resultado do julgamento do incidente nos autos do recurso, remessa necessária ou ação de competência originária, com a juntada de cópia do acórdão do órgão julgador, irão conclusos ao Relator para prosseguimento do seu trâmite. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

Art. 347-P - A decisão tomada pela maioria absoluta do órgão competente para julgar o incidente de arguição de inconstitucionalidade é precedente obrigatório e deve ser observada por todos os demais Órgãos Julgadores do Tribunal. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

Art. 347-Q - Aplicam-se ao incidente de arguição de inconstitucionalidade de lei ou de ato normativo, no que couber, as disposições relacionadas ao ordenamento, à instrução, ao julgamento, à publicidade e à revisão da tese jurídica previstas para os incidentes de resolução de demandas repetitivas e assunção de competência. [\(Incluído pelo art. 2º da Resolução nº 21/2016, de 15/09/2016\).](#)

## **Capítulo XIII - DOS RECURSOS PARA O TRIBUNAL DE JUSTIÇA**

### **Seção I - Dos Recursos Criminais**

Art. 348. Os recursos criminais, voluntários ou de ofício serão interpostos nos casos, pela forma e nos prazos estabelecidos na lei processual penal.

Art. 349. Não serão prejudicados os recursos que, por erro, falta ou omissão dos funcionários, não tiverem seguimento ou não forem apresentados dentro do prazo.

Art. 350. Havendo impropriedade de recurso, reconhecida desde logo pelo juiz, será o mesmo processado de acordo com o rito do recurso cabível; e, salvo a hipótese de má fé, a parte não será prejudicada pela interposição de um recurso por outro.

Art. 351. Os recursos em sentido estrito, interpostos de decisão, despacho ou sentença de juiz de Direito, serão distribuídos a Desembargadores de Câmara Criminal, salvo tratando-se de inclusão de jurado na lista geral, ou de exclusão da mesma lista, os quais serão de competência do Presidente do Tribunal.

Art. 352. As apelações, cabíveis de sentença definitivas de condenação ou de absolvição, proferidas por juiz singular ou pelo Tribunal do Júri, e de decisões definitivas ou com força de definitivas, que não comportam recurso em sentido estrito e proferidas por juiz de Direito, serão distribuídas aos desembargadores de Câmara Criminal.

Art. 353. Os recursos em sentido estrito serão apresentados à Secretaria do Tribunal de Justiça dentro do prazo de cinco dias, contado da publicação da resposta do juiz *a quo*, ou entregues ao Correio dentro do mesmo prazo.

Art. 354. Tratando-se de apelação, findos os prazos para razões, os autos serão remetidos à instância superior com as razões ou sem elas, no prazo de cinco dias, salvo no caso de ser necessária a extração de traslado (Cód. de Pr. Penal, art. 603, segunda

parte), hipótese em que o prazo será de trinta dias.

Art. 355. Nos recursos em sentido estrito e nas apelações das sentenças em processo de contravenção, ou de crime em que a lei comine pena de detenção, os autos irão imediatamente com vista ao Procurador Geral de Justiça, pelo prazo de cinco dias, e, em seguida, passarão, por igual prazo, ao Relator, que pedirá designação de dia para o julgamento.

§ 1º Anunciado o julgamento, pelo Presidente, e apregoadas as partes, com a presença destas ou à sua revelia, o Relator fará a exposição do feito e, em seguida, o Presidente concederá, pelo prazo de dez minutos, a palavra a advogados ou às partes que a solicitarem e ao Procurador Geral de Justiça, quando o requerer, por igual prazo.

§ 2º Os recursos de *habeas corpus* serão julgados na primeira sessão.

Art. 356. As apelações interpostas das sentenças proferidas em processos por crime a que a lei comine pena de reclusão, deverão ser processadas e julgadas pela forma estabelecida no art. 352, deste Regimento, com as seguintes modificações:

I – exarado o relatório nos autos, passarão estes ao revisor, que terá igual prazo o exame do processo e pedirá designação de dia para o julgamento;

II – os prazos serão ampliados ao dobro;

III – o tempo para os debates será de quinze minutos.

Art. 357. O Tribunal Pleno, as Câmaras Reunidas ou as Câmaras Especializadas decidirão por maioria de votos.

§ 1º No Tribunal Pleno, havendo empate, no julgamento de recursos, se o Presidente não houver tomado parte na votação proferirá voto de desempate; no caso contrário, prevalecerá a decisão mais favorável ao réu.

§ 2º O acórdão será apresentado à conferência na primeira sessão seguinte à do julgamento, ou no prazo de duas sessões, pelo juiz incumbido de lavrá-lo.

Art. 358. Se a sentença do Juiz Presidente do Tribunal do Júri for contrária à lei expressa ou divergir das respostas dos jurados aos quesitos, a Câmara Criminal, apreciando e julgando a apelação, fará a devida retificação.

Art. 359. Interposta apelação de decisão do Tribunal do Júri, com fundamento no art. 593, inciso III, letra "c", do Código de Processo Penal, a Câmara Criminal, se der provimento ao recurso, retificará a aplicação da pena ou da medida de segurança.

Art. 360. Se a apelação de decisão do Tribunal do Júri se fundar no art. 593, inciso III, letra "d", do Código de Processo Penal e a Câmara Criminal se convencer de que a decisão dos jurados é manifestamente contrária à prova dos autos, lhe dará provimento para sujeitar o réu a novo julgamento.

Parágrafo único. Não se admitirá, pelo mesmo motivo, segunda apelação.

Art. 361. No julgamento das apelações, poderá o Tribunal ou as Câmaras proceder a novo interrogatório do acusado, requerer testemunhas ou determinar outras diligências.

Art. 362. Quando cabível a apelação, não poderá ser usado o recurso em sentido estrito, ainda que somente de parte da decisão se recorra.

Art. 363. O Tribunal Pleno, as Câmaras Reunidas ou as Câmaras Criminais atenderão, nas suas decisões, ao disposto nos arts. 385, 386 e 387, do Código de Processo

Penal, no que for aplicável não podendo, porém, ser agravada a pena, quando somente o réu houver apelado da sentença.

Art. 364. Será dada carta testemunhável, em feitos criminais:

I – da decisão que denegar o recurso em sentido estrito ou proteste por novo júri;

II – daquela que, admitindo, embora, o recurso obstar à sua expedição e seguimento para a instância *ad quem*.

Parágrafo único. A extração do instrumento de carta testemunhável, o respectivo processamento e a apreciação pelo Tribunal de Justiça atenderão ao disposto na lei processual vigente.

## Seção II - Dos Recursos Cíveis

Art. 365. Os recursos cíveis, para o Tribunal de Justiça, serão interpostos nos casos, pela forma e nos prazos estabelecidos na lei processual civil.

§ 1º. Distribuídos, os autos serão imediatamente remetidos ao relator. (Redação dada pelo art. 62 da Resolução nº 06/2016, de 04/04/2016).

§ 2º. Aplicados os arts. 932, parágrafo único, e 933 do Código de Processo Civil, o Relator, no prazo de 30 (trinta) dias, depois de elaborar o voto, restituí-los-á, com relatório, à secretaria. (Redação dada pelo art. 62 da Resolução nº 06/2016, de 04/04/2016).

§ 3º. (Revogado) (Redação dada pelo art. 85 da Resolução nº 06/2016, de 04/04/2016).

§ 4º. Em seguida, os autos serão apresentados ao presidente, que designará dia para julgamento, ordenando, em todas as hipóteses de recursos previstos no Código de Processo Civil, a publicação da pauta no órgão oficial, devendo mediar entre a publicação e a sessão de julgamento o prazo de 5 (cinco) dias úteis. (Redação dada pelo art. 62 da Resolução nº 06/2016, de 04/04/2016).

§ 5º. No julgamento dos recursos cíveis, será observado o disposto nos arts. 937 a 941 e 946 do Código de Processo Civil. (Redação dada pelo art. 62 da Resolução nº 06/2016, de 04/04/2016).

§ 6º. O julgamento proferido pelo órgão colegiado do Tribunal de Justiça substituirá a decisão impugnada no que tiver sido objeto de recurso. (Redação dada pelo art. 63 da Resolução nº 06/2016, de 04/04/2016).

Art. 366. A apelação devolverá ao Tribunal o conhecimento da matéria impugnada.

§ 1º. Serão, porém, objeto de apreciação e julgamento pelo tribunal todas as questões suscitadas e discutidas no processo, ainda que não tenham sido solucionadas, desde que relativas ao capítulo impugnado. (Redação dada pelo art. 64 da Resolução nº 06/2016, de 04/04/2016).

§ 2º Quando o pedido ou a defesa tiver mais de um fundamento e o juiz acolher apenas um deles, a apelação devolverá ao Tribunal o conhecimento dos demais.

§ 3º. As questões resolvidas na fase de conhecimento, se a decisão a seu respeito não comportar agravo de instrumento, não são cobertas pela preclusão e devem ser suscitadas em preliminar de apelação, eventualmente interposta contra a decisão final, sendo que o recorrente será intimado para, em 15 (quinze) dias, manifestar-se a respeito delas, quando as referidas preliminares forem suscitadas nas contrarrazões do recurso. (Redação dada pelo art. 64 da Resolução nº 06/2016, de 04/04/2016).

§ 4º As questões de fato, não propostas no juízo inferior, poderão ser suscitadas na apelação, se a parte provar que deixou de fazê-lo por motivo de força maior.

§ 5º. O capítulo de sentença que confirma, concede ou revoga a tutela provisória, é impugnável na apelação. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

§ 6º. A apelação terá efeito suspensivo, salvo nos casos e formas legais. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

§ 7º. Recebido o recurso de apelação no Tribunal e distribuído imediatamente, o relator decidi-lo-á monocraticamente nas hipóteses do art. 932, III a V, do Código de Processo Civil, ou, se não for o caso de decisão monocrática, elaborará seu voto para julgamento do recurso pelo órgão colegiado. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

§ 8º. O Tribunal deve julgar desde logo o mérito da demanda se o processo estiver em condições de imediato julgamento. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

§ 9º. Quando o resultado da apelação não for unânime, seja ele de mérito ou não, e independentemente de a sentença apelada ser de mérito ou não, o presidente do órgão julgador procederá à convocação de novos julgadores, para, na mesma sessão, ou em outra a ser designada, proferirem votos para confirmar ou reverter o resultado do julgamento já iniciado, com a inclusão em pauta da apelação neste último caso. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

§ 10. A convocação dos novos julgadores deverá ser em número suficiente para confirmar ou reverter o resultado do julgamento já iniciado, incluindo-se mais um para funcionar como suplente, que apenas terá voto em substituição a julgador impedido, suspeito, ausente ou, por qualquer outro motivo, impossibilitado de atuar no novo julgamento. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

§ 11. Na hipótese de o novo julgamento, por deliberação do órgão colegiado, ficar designado para outra sessão, a convocação de novos julgadores se dará mediante sorteio, preferencialmente entre os integrantes das Câmaras Especializadas Cíveis. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

§ 12. O sorteio a que se refere o parágrafo anterior será realizado ao final de cada sessão em que houver divergência no julgamento de apelações cíveis, devendo os julgadores sorteados participarem da continuação do julgamento de todas as apelações nas quais, naquela sessão, houve a divergência. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

§ 13. Será assegurado às partes e aos eventuais terceiros o direito de sustentar oralmente suas razões perante os novos julgadores. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

§ 14. Os julgadores que já tiverem votado e aqueles que tenham sido convocados se vinculam ao processo para a continuação do julgamento, o qual apenas poderá ocorrer se todos os desembargadores votantes estiverem presentes, inclusive os julgadores convocados. Na ausência de algum deles, o julgamento ficará adiado para a sessão imediatamente subsequente. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

§ 15. Na continuação do julgamento, os julgadores que já tiverem votado poderão rever seus votos por ocasião do prosseguimento do julgamento. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).



§ 16. O disposto nos parágrafos 9º a 14 se aplicará igualmente à hipótese em que o agravo de instrumento seja provido por maioria, para reformar a decisão de primeiro grau que tiver julgado parcialmente o mérito da demanda. (Redação dada pelo art. 65 da Resolução nº 06/2016, de 04/04/2016).

Art. 367. Caberá agravo de instrumento contra as decisões interlocutórias apenas nos casos expressamente referidos em lei. (Redação dada pelo art. 66 da Resolução nº 06/2016, de 04/04/2016).

§ 1º. Recebido o agravo de instrumento no Tribunal e distribuído imediatamente, se não for o caso de aplicação do art. 932, III e IV, do Código de Processo Civil, o relator, no prazo de 5 (cinco) dias, procederá na forma do art. 1.019, I, II e III, do Código de Processo Civil. (Redação dada pelo art. 66 da Resolução nº 06/2016, de 04/04/2016).

§ 2º. O relator solicitará dia para julgamento em prazo não superior a 1 (um) mês da intimação do agravado. (Redação dada pelo art. 66 da Resolução nº 06/2016, de 04/04/2016).

§ 3º. No julgamento do agravo de instrumento, quando houver reforma, por maioria, da decisão que julgar parcialmente o mérito do recurso, proceder-se-á na forma dos §§ 9º, 10º e 12º do art. 366 deste Regimento. (Redação dada pelo art. 66 da Resolução nº 06/2016, de 04/04/2016).

§ 4º. Será assegurado às partes e a eventuais terceiros o direito de sustentar oralmente suas razões perante os novos julgadores nos casos de julgamento de agravo de instrumento contra decisões interlocutórias que versem sobre tutelas provisórias de urgência e da evidência. (Redação dada pelo art. 66 da Resolução nº 06/2016, de 04/04/2016).

§ 5º. (Revogado) (Redação dada pelo art. 85 da Resolução nº 06/2016, de 04/04/2016).

## **CAPÍTULO XIV - DOS RECURSOS DE ATOS, DELIBERAÇÃO E DECISÕES DOS ÓRGÃOS JURISDICIONAIS E ADMINISTRATIVOS DO TRIBUNAL**

### **Seção I - Dos Embargos Declaratórios**

Art. 368. Cabem embargos de declaração contra qualquer decisão judicial dos órgãos colegiados, assim como contra decisão do relator ou de outro integrante do Tribunal, nos feitos cíveis e criminais, que contenham quaisquer dos vícios ou defeitos previstos em lei. (Redação dada pelo art. 67 da Resolução nº 06/2016, de 04/04/2016).

§ 1º Os embargos declaratórios aos acórdãos proferidos em feitos cíveis deverão ser opostos dentro em cinco dias da data da publicação do acórdão; e os apostos a acórdãos proferidos em feitos criminais, no prazo de dois dias, também contado da publicação da decisão.

§ 2º. Os embargos de declaração não estão sujeitos a preparo e serão deduzidos em petição de que constem os vícios ou defeitos de que padeça a decisão embargada. (Redação dada pelo art. 67 da Resolução nº 06/2016, de 04/04/2016).

§ 3º. O relator apresentará os embargos em mesa para julgamento na sessão subsequente à sua oposição, proferindo o voto, e, não havendo julgamento nessa sessão, o recurso será incluído em pauta automaticamente. (Redação dada pelo art. 67 da Resolução nº 06/2016, de 04/04/2016).

§ 4º. O relator conhecerá dos embargos como agravo interno, quando não forem preenchidos os requisitos do recurso interposto, devendo proceder à intimação do recorrente, para, no prazo de 5 (cinco) dias, complementar as razões recursais de modo a ajustá-la às exigências do art. 1.021, § 1º, do Código de Processo Civil. (Redação dada pelo art. 67 da Resolução nº 06/2016, de 04/04/2016).

§ 5º. No caso de conversão dos embargos em agravo interno, o relator deverá intimar o agravado para manifestar-se sobre o recurso no prazo de 15 (quinze) dias. (Redação dada pelo art. 68 da Resolução nº 06/2016, de 04/04/2016).

§ 6º. Os embargos de declaração não possuem efeito suspensivo e interrompem o prazo para interposição de recurso. (Redação dada pelo art. 68 da Resolução nº 06/2016, de 04/04/2016).

§ 7º. A eficácia da decisão embargada, seja monocrática ou colegiada, poderá ser suspensa pelo relator, nas condições do art. 1.026, § 1º, do Código de Processo Civil. (Redação dada pelo art. 68 da Resolução nº 06/2016, de 04/04/2016).

Art. 369. Nos casos de embargos manifestamente protelatórios, o relator procederá, na forma do art. 1.026, §§ 2º, 3º e 4º, do Código de Processo Civil. (Redação dada pelo art. 69 da Resolução nº 06/2016, de 04/04/2016).

## **Seção II - Dos Embargos Infringentes e de Nulidade**

Art. 370. Quando, em feito criminal, não for unânime a decisão de segunda instância, desfavorável ao réu, admitem-se embargos infringentes e de nulidade, que poderão ser opostos no prazo de dez dias, a contar da publicação do acórdão, na forma do que estabelece o art. 613, do Código de Processo Penal.

Parágrafo único. Se o desacordo for parcial, os embargos serão restritos à matéria objeto de divergência.

Art. 371. Os embargos a que se refere o artigo anterior serão distribuídos a Desembargador que não tenha funcionado como Relator ou Revisor do acórdão embargado.

Parágrafo único. Serão observados no processamento dos embargos, as normas atinentes ao processamento das apelações.

Art. 372. (Revogado) (Redação dada pelo art. 85 da Resolução nº 06/2016, de 04/04/2016).

## **Seção III – Do Agravo Interno**

(Redação dada pelo art. 70 da Resolução nº 06/2016, de 04/04/2016).

Art. 373. Das decisões do presidente e do vice-presidente, dos presidentes de órgãos fracionários, dos relatores, ou de qualquer outro integrante do Tribunal de Justiça, caberá agravo interno para o respectivo órgão colegiado, na forma deste Regimento. (Redação dada pelo art. 71 da Resolução nº 06/2016, de 04/04/2016).

§ 1º. Revogado pelo art. 85 da Resolução nº 06/2016, de 04/04/2016).

§ 2º. O prazo para a interposição do agravo interno e para respondê-lo é de 15 (quinze) dias, contados na forma do art. 1.003 do Código de Processo Civil. (Redação dada pelo art. 71 da Resolução nº 06/2016, de 04/04/2016).

§ 3º. O processamento e o julgamento do agravo interno dar-se-á na forma do disposto no art. 1.021, §§ 1º, 2º, 4º e 5º, do Código de Processo Civil. (Redação dada pelo art. 72 da Resolução nº 06/2016, de 04/04/2016).

Art. 374. O agravo será protocolado e submetido imediatamente ao prolator da decisão recorrida, que procederá na forma do § 3º do art. 373 deste Regimento. (Redação dada pelo art. 73 da Resolução nº 06/2016, de 04/04/2016).

Art. 375. Provido o agravo, o órgão determinará o que for de direito. (Redação dada pelo art. 9º da Resolução nº 03/1999, de 10/06/1999).

Art. 376. O agravo regimental não terá efeito suspensivo. (Redação dada pelo art. 9º da Resolução nº 03/1999, de 10/06/1999).

#### **Seção IV - Dos Recursos Administrativos**

Art. 377. Cabe ao Plenário do Tribunal de Justiça apreciar e julgar:

I – os embargos opostos a decisões não unânimes, proferidas por Câmara Criminal, em processos oriundos do Conselho de Justificação da Polícia Militar do Estado e relativos a oficiais da mesma corporação;

II – os recursos de decisões proferidas, em instância originária, pelo Conselho da Magistratura, e que concluam pela imposição de penalidade;

III – os recursos interpostos da aplicação de pena disciplinar pelo Presidente do Tribunal de Justiça, pelos Presidentes das Câmaras Reunidas e pelos Presidentes das Câmaras Especializadas.

Art. 378. Os recursos, interpostos da imposição de pena disciplinar, terão efeito suspensivo.

Art. 379. Os recursos dos atos e decisões do Corregedor Geral da Justiça, para o Conselho da Magistratura, serão disciplinadas no Regimento Interno do referido Conselho.

Art. 380. No julgamento dos recursos administrativos, embora participe da discussão, não votará o prolator da decisão recorrida ou o Relator do acórdão, quando se tratar de decisão colegiada.

Art. 381. Os recursos administrativos serão interpostos no prazo de cinco dias, contados da publicação do ato, deliberação ou decisão, salvo nas hipóteses do art. 377, inciso III, e art. 379, deste Regimento, em que o prazo para a interposição de recurso será de dez dias, e, outrossim, nos casos em que a lei expressamente fixar prazo diverso.

#### **Seção V - Do Recurso Extraordinário e do Recurso Especial**

(Seção V com designação alterada pelo art. 10 da Resolução nº 01/1999, de 10/02/1999).

Art. 382. Das decisões proferidas pelo Tribunal de Justiça caberá, nos casos previstos na Constituição da República, recurso extraordinário para o Supremo Tribunal Federal e recurso especial para o Superior Tribunal de Justiça. (Redação dada pelo art. 10 da Resolução nº 03/1999, de 10/06/1999).

Art. 383. O recurso será interposto no prazo estabelecido na lei processual pertinente, perante o Presidente do Tribunal de Justiça, mediante petição, com precisa indicação do

dispositivo que autorize, dentre os casos previstos nos arts. 102, III e 105, III, da Constituição Federal. (Redação dada pelo art. 10 da Resolução nº 03/1999, de 10/06/1999).

Art. 384. Se na causa tiverem sido vencidos autor e réu, qualquer deles poderá aderir ao recurso da outra parte, nos termos da lei processual civil. (Redação dada pelo art. 10 da Resolução nº 03/1999, de 10/06/1999).

Art. 385. Recebida a petição pela Secretaria e aí protocolada, o recurso terá, no Tribunal de Justiça, a tramitação prescrita na legislação processual e nas disposições normativas aplicáveis à matéria. (Redação dada pelo art. 10 da Resolução nº 03/1999, de 10/06/1999).

Art. 386. Caberá ao Presidente do Tribunal de Justiça deferir ou negar o seguimento do recurso. (Redação dada pelo art. 10 da Resolução nº 03/1999, de 10/06/1999).

Art. 387. Incumbe ao Presidente do Tribunal de Justiça, a fim de fundamentar a sua decisão, examinar a ocorrência de arguição razoável de ofensa à Constituição e de manifesta divergência com súmulas do Supremo Tribunal Federal e do Superior Tribunal de Justiça, bem como, nos casos especificados nos respectivos regimentos internos, as hipóteses ressalvadas de cabimento dos recursos. (Redação dada pelo art. 10 da Resolução nº 03/1999, de 10/06/1999).

Parágrafo único. Será sempre motivado o despacho pelo qual o Presidente do Tribunal admitir o recurso ou denegar-lhe a interposição.

Art. 388. Denegada a interposição do recurso, caberá agravo de instrumento, no prazo de cinco dias, cumprindo-se, no processamento do agravo, o que dispõe a legislação processual. (Redação dada pelo art. 10 da Resolução nº 03/1999, de 10/06/1999).

Art. 389. O preparo do recurso extraordinário e do recurso especial será feito mediante guia extraída pela Secretaria do Tribunal, para recolhimento no estabelecimento bancário competente, observando-se as disposições da legislação processual civil. (Redação dada pelo art. 10 da Resolução nº 03/1999, de 10/06/1999).

Art. 390. Devidamente preparado o recurso, serão os autos remetidos, dentro de quinze dias, à Secretaria do Supremo Tribunal Federal ou do Superior Tribunal de Justiça, observado, inclusive, quando for o caso, o que dispõe o art. 543, do Código de Processo Civil. (Redação dada pelo art. 10 da Resolução nº 03/1999, de 10/06/1999).

## **CAPÍTULO XV – DA EXECUÇÃO OU DO CUMPRIMENTO DAS DECISÕES DO TRIBUNAL NAS CAUSAS DE SUA COMPETÊNCIA ORIGINÁRIA**

(Redação dada pelo art. 74 da Resolução nº 06/2016, de 04/04/2016).

### **Seção I - Disposições Gerais**

Art. 391. Compete ao Plenário do Tribunal ou às Câmaras Reunidas e às Especializadas a execução ou o cumprimento, conforme o caso, dos acórdãos que prolatarem nas causas cíveis e criminais de sua competência originária.

(Redação dada pelo art. 75 da Resolução nº 06/2016, de 04/04/2016)

Art. 392. Nos feitos cíveis, o cumprimento das decisões judiciais, nas hipóteses do artigo anterior, será de competência do relator do acórdão exequendo.

(Redação dada pelo art. 75 da Resolução nº 06/2016, de 04/04/2016)

Art. 393. Nas ações penais originárias, cabe ao Presidente da respectiva Câmara prover a execução do julgado, cumprindo-se o que dispuser a legislação processual.

(Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999).

Art. 394. Ressalvado o disposto nos arts. 392 e 393 deste Regimento, a execução, ou o cumprimento de decisão judicial, nos feitos e papéis submetidos ao Tribunal de Justiça, competirá: (Redação dada pelo art. 75 da Resolução nº 06/2016, de 04/04/2016)

I – ao Presidente do Tribunal, quanto aos seus despachos e ordens e quanto às decisões do Plenário, em matéria administrativa;

II – aos presidentes das Câmaras Reunidas ou das Câmaras Especializadas, quanto aos respectivos despachos e deliberações;

III – aos relatores, quanto aos seus despachos acautelatórios ou de instrução e direção do processo.

Art. 395. Os atos de execução, ou de cumprimento de ordem judicial, que não dependerem de sentença, serão ordenados a quem os deva praticar ou delegados a outras autoridades judiciárias. (Redação dada pelo art. 75 da Resolução nº 06/2016, de 04/04/2016)

Art. 396. Se necessário, os incidentes de execução, ou de cumprimento de decisão judicial, podem ser levados à apreciação: (Redação dada pelo art. 75 da Resolução nº 06/2016, de 04/04/2016)

I – do Presidente, por qualquer dos Desembargadores;

II – do Plenário, pelo Presidente, pelo relator, pelas Câmaras Reunidas ou pelas Câmaras Especializadas através dos seus presidentes;

III – das Câmaras Reunidas ou das Câmaras Especializadas pelo respectivo Presidente ou pelo relator.

Parágrafo único. A execução, ou o cumprimento de decisão ou ordem judicial, atenderá ao disposto na legislação processual em vigor. (Redação dada pelo art. 75 da Resolução nº 06/2016, de 04/04/2016)

## **Seção II – Do Cumprimento de Decisão Judicial e da Execução de Título Extrajudicial contra a Fazenda Pública**

(Redação dada pelo art. 76 da Resolução nº 06/2016, de 04/04/2016)

Art. 397. No cumprimento de sentença que impuser à Fazenda Pública o dever de pagar quantia certa, em ação de competência originária do Tribunal, se a devedora, intimada pessoalmente pelos meios legais para impugnar o cumprimento de sentença, não o fizer no prazo de lei, ou se forem rejeitadas as suas arguições, o presidente do Tribunal expedirá precatório em favor do requerente, observando-se o disposto na Constituição Federal. (Redação dada pelo art. 77 da Resolução nº 06/2016, de 04/04/2016)

Art. 398. Salvo na hipótese do artigo anterior, o pagamento será requisitado pelo juiz de direito competente ao presidente do Tribunal de Justiça, que expedirá o precatório em favor do requerente, a não ser nos casos de pagamento de obrigação de pequeno valor, quando, então, o juiz da causa procederá na forma do art. 535, § 3º, II, do Código de Processo Civil. (Redação dada pelo art. 77 da Resolução nº 06/2016, de 04/04/2016)

Art. 399. Das requisições de pagamento constarão expressamente: (Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999).

- I – a importância total da condenação;
- II – a quem deve ser paga a quantia requisitada;
- III – o inteiro teor do parecer proferido pelo representante judicial da Fazenda Pública, favorável ao pagamento, ou, caso contrário, com as razões de impugnação;
- IV – o reconhecimento da firma do Juiz requisitante;

Art. 400. A requisição será instruída, obrigatoriamente, com as cópias autenticadas, em duas vias, das seguintes peças, além de outras que o Juiz entender necessárias ou que as partes indicarem: [\(Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999\)](#).

- I – petição inicial da ação;
- II – procuração e substabelecimento, se houver;
- III – contestação;
- IV – sentença de primeiro grau;
- V – acórdão do Tribunal de Justiça;
- VI – acórdão do Supremo Tribunal Federal, ou do Superior Tribunal de Justiça no caso de ter havido recurso extraordinário, ou recurso especial;
- VII – petição inicial de execução;
- VIII – sentença que tenha julgado a liquidação;
- IX – conta da liquidação;
- X – manifestação do representante judicial da Fazenda Pública no sentido de estar a requisição de pagamento conforme aos autos originais.

Parágrafo único. As peças juntas por cópia deverão ser devidamente autenticadas.

Art. 401. Protocolada e autuada a requisição de pagamento, será ouvida a Procuradoria Geral de Justiça, após o que os autos serão conclusos ao Presidente, que decidirá, podendo ordenar diligências que entenda indispensáveis ao esclarecimento da matéria. [\(Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999\)](#).

§ 1º Da decisão do Presidente caberá agravo regimental.

[\(Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999\)](#).

§ 2º Deferido o pagamento, será feita a devida comunicação à autoridade fazendária competente, para o cumprimento, na conformidade da lei. [\(Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999\)](#).

Art. 402. Os pagamentos devidos pela Fazenda Pública em virtude de sentença judiciária serão feitos na ordem de apresentação dos precatórios e à conta dos créditos respectivos. [\(Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999\)](#).

§ 1º Se o credor for preterido no seu direito de preferência, o Presidente do Tribunal de Justiça poderá, depois de ouvido o Chefe do Ministério Público, ordenar o sequestro da quantia necessária para satisfazer o débito. [\(Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999\)](#).

§ 2º Além da publicação no Diário da Justiça da decisão do Presidente que houver deferido a requisição do pagamento, inteiro teor da mesma decisão será remetida ao juiz requisitante, para que a faça constar dos autos de que a requisição tenha sido extraída. [\(Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999\)](#).

Art. 403. Tratando-se de execução de sentença com origem em ação expropriatória, movida pela Fazenda Pública, a requisição de pagamento deverá ser encaminhada pelo juiz de Direito diretamente ao expropriante, para pagamento, acompanhada das peças enumeradas no art. 400 e satisfeitos os requisitos do art. 399 deste Regimento. [\(Redação](#)

dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999).

Parágrafo único. Ocorrendo pedido de seqüestro com base em alegada preterição de credor na ordem de preferência, o Presidente do Tribunal de Justiça requisitará ao Juízo de origem, para consulta, se necessário, cópia das peças que instruíram a requisição.

Art. 403-B. Na execução fundada em título extrajudicial, proceder-se-á na forma do art. 910 do Código de Processo Civil. (Redação dada pelo art. 78 da Resolução nº 06/2016, de 04/04/2016).

### **Seção III – Do Cumprimento Provisório de Decisão Judicial**

(Redação dada pelo art. 79 da Resolução nº 06/2016, de 04/04/2016).

Art. 404. O cumprimento provisório de acórdão proferido em processo de competência originária do Tribunal, impugnado por recurso desprovido de efeito suspensivo, será realizado da mesma forma que o cumprimento definitivo, sujeitando-se ao regime legal previsto no art. 520 do Código de Processo Civil. (Redação dada pelo art. 80 da Resolução nº 06/2016, de 04/04/2016).

Art. 405. O cumprimento provisório de acórdão na hipótese do artigo anterior, será requerida ao relator do processo. (Redação dada pelo art. 80 da Resolução nº 06/2016, de 04/04/2016).

Art. 406. Não sendo eletrônicos os autos, a petição de cumprimento provisório de acórdão, nos termos dos artigos anteriores, serão instruída com as peças indicadas no art. 522, parágrafo único, do Código de Processo Civil, cuja autenticidade poderá ser certificada pelo próprio advogado, sob pena de responsabilidade pessoal. (Redação dada pelo art. 80 da Resolução nº 06/2016, de 04/04/2016).

### **Seção IV - Da Intervenção do Estado em Município**

(Seção IV acrescentada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999).

Art. 407. A requisição de intervenção estadual, prevista no art. 36, IV, da Constituição Estadual, será promovida mediante representação do Procurador Geral de Justiça, para assegurar a observância de princípios indicados na Constituição Estadual, ou para prover a execução de lei, ordem ou decisão judicial. (Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999, publicada no DJE nº 4.059, de 11/06/1999).

Art. 408. O Presidente, ao receber o pedido: (Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999).

I – tomará as providências oficiais que lhe parecerem adequadas para remover, administrativamente, a causa do pedido;

II – mandará arquivá-lo, se for manifestamente infundado, cabendo do seu despacho agravo regimental.

Art. 409. Realizada a gestão prevista no inciso I do artigo precedente, solicitadas as informações à autoridade municipal e ouvido o Procurador Geral de Justiça, o pedido será relatado pelo Presidente, em sessão plenária. (Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999).

Parágrafo único. O julgamento, se não tiver sido público, será proclamado em sessão pública.

Art. 410. Julgado procedente o pedido, o Presidente do Tribunal imediatamente comunicará a decisão aos órgãos do Poder Público interessados e requisitará a intervenção ao Governador do Estado, que decidirá na forma do inciso III e § 1º, do art. 37, da Constituição Estadual. ([Redação dada pelo art. 11 da Resolução nº 03/1999, de 10/06/1999](#)).

## **CAPÍTULO XVI - DOS ATOS NORMATIVOS**

Art. 411. Compete privativamente ao Plenário do Tribunal de Justiça, mediante proposta ao Poder Legislativo, a iniciativa das leis que disponham sobre:

I – a alteração do número de membros do Tribunal;

II – a alteração da organização e da divisão judiciárias do Estado;

III – a criação de Juizados Especiais Cíveis e Criminais, de acordo com o art. 98, I, da Constituição Federal;

IV – a criação de justiça de paz remunerada e temporária, competente para habilitação de casamento;

V – a criação ou a extinção de cargos nos serviços auxiliares do Tribunal e a fixação dos respectivos vencimentos.

Parágrafo único. O Presidente do Tribunal de Justiça encaminhará à Assembleia Legislativa os projetos de leis aprovados pelo Plenário do Tribunal, através de resoluções.

Art. 412. É, outrossim, da competência privativa do Plenário do Tribunal de Justiça:

I – elaborar e aprovar por maioria absoluta de seus membros, o Regimento Interno do Tribunal e, nas mesmas condições, as emendas que, no mesmo tiverem de ser introduzidas;

II – homologar o Regimento Interno da Corregedoria Geral da Justiça, depois de aprovado pelo Conselho da Magistratura;

III – homologar o Regimento Interno do Conselho da Magistratura, elaborado e aprovado pelo mesmo órgão;

IV – aprovar e encaminhar ao órgão competente, na oportunidade própria, a proposta da despesa orçamentária do Poder Judiciário do Estado para o exercício seguinte;

V – elaborar e aprovar o regulamento para o concurso de Juiz de Direito Substituto, atendendo ao que dispõe a Constituição, a Lei Orgânica da Magistratura Nacional e a Organização Judiciária do Estado;

VI – elaborar e aprovar regulamento para as remoções e promoções de juízes de Direito, observando o disposto nos preceitos constitucionais e legais;

VII – elaborar e aprovar regulamentos e instruções de recursos para provimento de cargos na Secretaria e nos Serviços auxiliares do Tribunal de Justiça, Corregedoria da Justiça

e das comarcas do interior;

VIII – elaborar e aprovar a organização da Secretaria e dos Serviços auxiliares do Tribunal e a regulamentação de suas atividades.

Art. 413. As matérias de que tratam os arts. 411 e 412, serão apreciadas e examinadas pelo Tribunal de Justiça, à vista de proposição escrita, formulada pelo Presidente ou por qualquer Desembargador, e cientificada aos demais, inclusive aos licenciados ou



ausentes por qualquer motivo, observando-se uma antecedência mínima de quarenta e oito horas entre o recebimento da comunicação e a sessão em que a matéria houver de ser discutida e votada.

## **CAPÍTULO XVII - DOS ATOS DE PROVIMENTO E DE VACÂNCIA**

### **Seção I - Das Nomeações para os Cargos da Secretaria e dos Serviços Auxiliares**

Art. 414. Cabe ao Presidente do Tribunal de Justiça prover os cargos integrantes do quadro de sua Secretaria, da Corregedoria Geral da Justiça e dos serviços auxiliares, nomeando os que devam ocupá-los.

Art. 415. Nos casos em que o provimento requeira prévia habilitação e classificação em concurso o Tribunal de Justiça adotará as providências adequadas para isso, baixando as necessárias instruções, em que serão definidas as provas a serem realizadas e as matérias sobre que as mesmas versarão, e fixados os critérios de julgamento e de classificação.

### **Seção II - Das Secretarias dos Juizados Especiais**

(Seção II com designação conferida pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

Art. 416. Para cada unidade dos Juizados Especiais, bem como para cada Turma Recursal, funcionará uma secretaria, composta de: (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

I – um Diretor de Secretaria; (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

II – um Escrevente Judiciário; (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

III – um Escrevente Auxiliar; (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

IV – outros servidores designados. (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

Art. 417. O Diretor de Secretaria será recrutado preferencialmente dentre bacharéis em Direito, integrantes do quadro permanente de pessoal do Poder Judiciário. (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

§ 1º Para o recrutamento, o Tribunal poderá aplicar teste seletivo entre os candidatos, que, aprovados, receberão treinamento específico, consoante programa definido pela Corregedoria Geral da Justiça. (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

§ 2º A escolha dos candidatos a Diretor de Secretaria, para nomeação pelo Presidente do Tribunal, será feita: (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

I – na comarca da capital, mediante indicação do Tribunal, em lista tríplice, observado o disposto nos arts. 173, última parte, 174 e 180; (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

II – nas comarcas do interior, mediante indicação do juiz de direito titular, ou daquele que a este estiver substituindo. (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

Art. 418. As disposições do artigo anterior, ressalvadas as exigências específicas quanto a profissão e formação, são aplicáveis ao recrutamento de juizes leigos e de conciliadores. (Redação dada pelo art. 12 da Resolução nº 03/1999, de 10/06/1999).

## **CAPÍTULO XVIII - DAS COMISSÕES**

Art. 419. O Tribunal manterá as seguintes comissões permanentes:

- a) Comissão de Organização Judiciária, Regimento Interno e Jurisprudência;
- b) Comissão de Biblioteca e Publicações.

§ 1º Cada comissão permanente compõe-se de pelo menos dois (2) Desembargadores, escolhidos entre os que não exercerem cargo de direção no Tribunal de Justiça. (Redação dada pelo art. 1º da Resolução nº 02, de 23/06/1988).

§ 2º O Tribunal e seu Presidente podem criar comissões temporárias com qualquer número de membros.

Art. 420. Será Presidente das Comissões permanentes o integrante que contar maior tempo no Tribunal.

Art. 421. Compete às Comissões permanentes ou temporárias:

I – expedir normas de serviço e sugerir ao Presidente do Tribunal as que envolvam matéria de sua competência;

II – requisitar ao Presidente do Tribunal os servidores de que necessitar para os serviços auxiliares.

Art. 422. São atribuições especiais da Comissão de Organização Judiciária, Regimento Interno e Jurisprudência:

I – velar pela complementação da organização judiciária, Regimento Interno, propondo emendas aos textos em vigor, a fim de supri-lhes as omissões e corrigir as faltas, emitindo parecer sobre as emendas de iniciativas de outros Desembargadores;

II – opinar em processos administrativos, quando consultada pelo Presidente ou pelo Corregedor Geral;

III – selecionar os acórdãos que devam ser publicados em seu inteiro teor na Revista "Piauí Judiciário", preferindo os indicados pelos relatores;

IV – encaminhar para publicação no Diário da Justiça, em sumário, as decisões não publicadas na íntegra.

Parágrafo único. As alterações a estas normas internas, observado o procedimento constante do inciso I, serão feitas através de emendas regimentais, numeradas em ordem crescente. (Parágrafo único acrescentado pelo art. 13 da Resolução nº 03/1999, de 10/06/1999).

Art. 423. São atribuições especiais da Comissão de Biblioteca e Publicações:

I – orientar os serviços da biblioteca;

II – velar pela expansão, atualização e publicação da jurisprudência do Tribunal, mantendo a regularidade de edição da Revista Piauí Judiciário;

III – cuidar da publicação do Diário da Justiça;

IV – encarregar-se do Plano Editorial do Judiciário.

Art. 424. As Comissões serão eleitas pelo período de dois (02) anos, na ocasião da escolha dos dirigentes do Tribunal, tendo seus membros a gratificação que a lei fixar.

## **DISPOSIÇÕES FINAIS**

Art. 425. Os desembargadores ausentes por motivo de férias, licença ou outro afastamento justo, terão direito a voto, oferecido em sobrecarta opaca, autenticada com a rubrica do votante, encerrada em outra que será encaminhada ao Presidente do Tribunal.

§ 1º A sobrecarta somente será aberta no ato da apuração.

§ 2º Esse direito de voto prevalece tanto nas eleições para os cargos de direção do Tribunal, ou outros, quanto na formação de listas para acesso, promoção ou remoção de Juízes.

Art. 426. O Desembargador licenciado ou em férias pode ser votado para cargo ou comissão, desde que esteja desimpedido.

Art. 427. Os Desembargadores são obrigados a residir em Teresina, podendo gozar férias e licenças onde lhes convier, comunicando seu endereço, por escrito, ao Presidente do Tribunal.

Art. 428. Revogadas as disposições em contrário, este Regimento Interno entra em vigor na data de sua publicação.

Este **REGIMENTO INTERNO** foi aprovado pelo Egrégio TRIBUNAL PLENO, em sua sessão de 12 de novembro de 1987, constituído dos Exmos. Srs. Desembargadores **PAULO DE TARSO MELLO E FREITAS** (PRESIDENTE), **MANFREDI MENDES DE CERQUEIRA** (VICE-PRESIDENTE), **ANTONIO RIBEIRO DE ALMEIDA** (CORREGEDOR DA JUSTIÇA), **ALUÍSIO SOARES RIBEIRO**, **ADOLFO UCHÔA FILHO**, **RAIMUNDO BARBOSA DE CARVALHO BAPTISTA**, **MILTON NUNES CHAVES**, **ÁLVARO BRANDÃO FILHO**, **WALTER DE CARVALHO MIRANDA** e **TOMAZ GOMES CAMPELO**, presente o Exmo. Sr. Procurador Geral de Justiça, Dr. **JOSINO RIBEIRO NETO**.

Teresina, 15 de março de 1988

Des. **RAIMUNDO BARBOSA DE CARVALHO BAPTISTA**  
PRESIDENTE

*Este texto não substitui o publicado no Diário de Justiça do Estado nº 1.489, SUPLEMENTO ESPECIAL, de 22/03/1988, pp. 1/102, e nem o publicado no Diário de Justiça Eletrônico nº 7.951, de 06/04/2016, considerado publicado em 07/04/2016, pp. 02/12.*

## **RESOLUÇÕES QUE ALTERARAM O REGIMENTO INTERNO**

- 1) Resolução nº 02/1988, de 23/06/1988;
- 2) Resolução nº 01/1990, de 08/02/1990;
- 3) Resolução nº 02/1995, de 24/08/1995;
- 4) Resolução nº 01/1996, de 07/03/1996;
- 5) Resolução nº 01/1997, de 15/05/1997;
- 6) Resolução nº 02/1997, de 15/05/1997;
- 7) Resolução nº 05/1997, de 1º/10/1997;
- 8) Resolução nº 01/1999, de 10/02/1999;
- 9) Resolução nº 03/1999, de 10/06/1999;
- 10) Resolução nº 02/2003, de 13/02/2003;
- 11) Resolução nº 11/2005, de 15/09/2005;
- 12) Resolução nº 13/2005, de 29/09/2005;
- 13) Resolução nº 14/2005, de 20/10/2005;
- 14) Resolução nº 20/2005, de 16/12/2005;
- 15) Resolução nº 20/2006, de 09/11/2006;
- 16) Resolução nº 05/2007, de 10/04/2007;
- 17) Resolução nº 04/2008, de 27/03/2008;
- 18) Resolução nº 16/2010, de 16/07/2010;
- 19) Resolução nº 17/2010, de 24/03/2011;
- 20) Resolução nº 30/2010, de 16/09/2011;
- 21) Resolução nº 42/2011, de 24/11/2011;
- 22) Resolução nº 01/2012, de 12/01/2012;
- 23) Resolução nº 02/2012, de 12/01/2012;
- 24) Resolução nº 04/2012, de 26/01/2012;
- 25) Resolução nº 21/2012, de 19/04/2012;
- 26) Resolução nº 09/2013, de 27/06/2013;
- 27) Resolução nº 10/2013, de 07/08/2013;
- 28) Resolução nº 18/2013, de 28/11/2013;
- 29) Resolução nº 04/2014, de 27/02/2014;
- 30) Resolução nº 11/2014, de 22/05/2014;
- 31) Resolução nº 11/2014, de 22/05/2014;
- 32) Resolução nº 28/2014, de 27/11/2014;
- 33) Resolução nº 22/2015, de 24/09/2015;
- 34) Resolução nº 06/2016, de 04/04/2016;
- 35) Resolução nº 21/2016, de 15/09/2016;
- 36) Resolução nº 24/2016, de 15/09/2016;
- 37) Resolução nº 35/2016, de 29/09/2016;
- 38) Resolução nº 43/2016, de 24/12/2016.

LEI nº 3.716, de 12 de dezembro de 1979.

Dispõe sobre a Organização Judiciária do Estado do Piauí e dá outras providências.

- Lei recepcionada como lei complementar, por força do art. 77, parágrafo único, II, da Constituição Estadual, de 05 de outubro de 1989.
- Na medida do possível, o texto encontra-se ajustado com as regras de redação legislativa estabelecidas pela Lei Complementar n. 95, de 26 de fevereiro de 1998.

O GOVERNADOR DO ESTADO DO PIAUÍ:

Faço saber que a Assembléia Legislativa decreta e eu sanciono a seguinte Lei:

## LIVRO I DA ORGANIZAÇÃO E DA DIVISÃO JUDICIÁRIA

### TÍTULO I DISPOSIÇÕES PRELIMINARES

#### CAPÍTULO I

Art. 1º O território do Estado do Piauí, para fins de Administração da Justiça, divide-se em comarcas e termos judiciários, constituindo-se numa só circunscrição para os atos da competência do Tribunal de Justiça.

- Artigo com redação dada pelo art. 1º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.

Art. 2º *(Revogado pelo art. 6º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.)*

Art. 3º Classificam-se as comarcas em três categorias ou entrâncias.

- Artigo com redação dada pelo art. 1º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.

Art. 4º A comarca constitui-se de um ou mais municípios.

Parágrafo único. A sede da comarca é a do município que lhe dá o nome.

#### CAPÍTULO II DA DIVISÃO JUDICIÁRIA

Art. 5º A divisão judiciária do Estado do Piauí compreende:

I – *(Revogado pelo art. 6º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.)*

II – **V E T A D O:**

- a) Teresina, com trinta e quatro Varas e dez Juizados especiais cíveis e criminais;
- b) Parnaíba, com seis Varas e dois Juizados especiais cíveis e criminais;
- c) Picos, com cinco Varas e um Juizado especial cível e criminal;
- d) Floriano, com três Varas e um Juizado especial cível e criminal;

e) Campo Maior, com três Varas e um Juizado especial cível e criminal;

f) **V E T A D O**;

g) **V E T A D O**.

III – quarenta e duas Comarcas de entrância intermediária, sendo:

a) União e Uruçuí, com duas Varas e um Juizado especial cível e criminal;

b) Batalha, Barras, Bom Jesus, Canto do Buriti, Paulistana, Piracuruca, Pedro II, São João do Piauí, com uma Vara e um Juizado especial Cível e Criminal;

c) Regeneração, com duas Varas;

d) Água Branca, Alto Longá, Amarante, Avelino Lopes, Beneditinos, Buriti dos Lopes, Castelo do Piauí, Cocal, Cristino Castro, Demerval Lobão, Elesbão Veloso, Esperantina, Fronteiras, Guadalupe, Gilbués, Inhumas, Itainópolis, Itaueira, Jaicós, Jerumenha, Luiz Correia, Luzilândia, Miguel Alves, Padre Marcos, Palmeirais, Pio IX, Porto, São Miguel do Tapuio, São Pedro do Piauí, Simões e Simplício Mendes, com uma Vara.

IV – quarenta e cinco Comarcas de entrância inicial, com sede em Angical do Piauí, Anísio de Abreu, Antônio Almeida, Aroazes, Arraial do Piauí, Barro Duro, Bertolínea, Bocaina, Brasileira, Campinas do Piauí, Capitão de Campos, Caracol, Conceição do Canindé, Cristalândia do Piauí, Curimatá, Domingos Mourão, Elizeu Martins, Flores do Piauí, Francinópolis, Francisco Santos, Ipiranga do Piauí, Isaías Coelho, Joaquim Pires, Landri Sales, Manoel Emídio, Marcolândia, Marcos Parente, Matias Olímpio, Monsenhor Gil, Monsenhor Hipólito, Monte Alegre do Piauí, Nazaré do Piauí, Nossa Senhora dos Remédios, Paes Landim, Parnaguá, Pimenteiras, Redenção do Gurguéia, Ribeiro Gonçalves, Santa Cruz do Piauí, Santa Filomena, São Félix do Piauí, São Gonçalo do Piauí, São Julião, Socorro do Piauí e Várzea Grande.

V – *(Revogado pelo art. 6º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.)*

VI – *(Revogado pelo art. 6º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.)*

Parágrafo único. *(Revogado pelo art. 6º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.)*

### **CAPÍTULO III**

#### **CRIAÇÃO, ELEVAÇÃO, REBAIXAMENTO E EXTINÇÃO DE COMARCA**

Art. 6º São requisitados para a criação de Comarca:

a) população mínima de dez mil habitantes no município, com, pelo menos, dois mil na sede;

b) território de área superior a quarenta quilômetros quadrados;

c) serviços forenses, apurados na Comarca que tiver de sofrer desdobramento de no mínimo, sessenta (60) processos anuais, de qualquer natureza;

d) receita tributária federal, estadual, municipal superior a mil vezes o salário-mínimo regional, em sua totalidade;

e) prédios apropriados de domínio do Estado ou do Município, para:

1. todas as necessidades dos serviços forenses, inclusive edifício para a Cadeia Pública, com a devida segurança e em condições de regularidade de regime de prisão provisória;

2. residência condigna do Juiz e Promotor;

3. provimento de todos os cargos judiciais e do Ministério Público.

f) mil eleitores regularmente inscritos.

§ 1º Para que se apurem os requisitos de que tratam as alíneas a e e deste artigo, recorre-se a informações do Prefeito Municipal, do Juiz de Direito e do Promotor Público,

assim como a subsídios de geografia e estatística dos órgãos técnicos competentes da Comarca que tiver de ser desdobrada.

§ 2º O município interessada em elevar-se a Comarca poderá concorrer com recursos próprios para que os efeitos se efetivem as condições exigidas neste artigo.

§ 3º Criada a Comarca, a instalação dar-se-á em data fixada pelo Tribunal de Justiça e em solenidade dirigida pelo seu Presidente ou Desembargador por ele designado.

Art. 7º Para elevação de entrância, o Tribunal de Justiça observará o desenvolvimento de serviços judiciários, o interesse público e as condições sociais da sede da Comarca.

Art. 8º A redução ou supressão dos requisitos exigidos para que se crie Comarca ou se eleve entrância poderá ter como consequência que se extinga aquela e se rebaixe esta, conforme for o caso.

Art. 9º Os Termos Judiciários devem ser instalados pelo Juiz de Direito da Comarca.

## **TÍTULO II DOS ÓRGÃOS DO PODER JUDICIÁRIO**

### **CAPÍTULO I**

Art. 10. São órgãos do Poder Judiciário do Estado:

- I – o Tribunal de Justiça,
- II – o Conselho da Magistratura;
- III – a Corregedoria da Justiça;
- IV – os Juízes de Direito;
- V – o Tribunal do Júri;
- VI – a Auditoria Militar e o Conselho de Justiça Militar;
- VII – os Juízes de Direito Adjunto;
- VIII – os Juízes de Paz.

Art. 11. Consideram-se órgãos auxiliares do Poder Judiciário:

- I – Ministério Público;
- II – a Assistência Judiciária;
- III – os Advogados, os Provisionados e os Estagiários;
- IV – os Serventuários dos Ofícios de Justiça;
- V – os Funcionários da Justiça.

## **CAPÍTULO II DO TRIBUNAL DE JUSTIÇA E DAS CÂMARAS COMPOSIÇÃO E COMPETÊNCIA**

### **Seção I Disposições Preliminares**

Art. 12. O Tribunal de Justiça, com sede na Capital e jurisdição em todo o território do Estado, compõe-se de dez Desembargadores e constitui-se em Tribunal Pleno, em Câmaras Reunidas e em Câmaras Especializadas.

Art. 13. O Tribunal de Justiça declara a inconstitucionalidade de lei, ou de ato de poder público, somente pelo voto da maioria absoluta dos seus membros.

Art. 14. O Regimento Interno do Tribunal e dos demais órgãos do Poder Judiciário disporá sobre o funcionamento, processamento e julgamento dos feitos de sua competência, das Câmaras Reunidas e Especializadas.

## Seção II Da Competência

Art. 15. Compete ao Tribunal Pleno:

I – processar e julgar originariamente:

a) o Governador do Estado, Vice-Governador, Prefeito da Capital e os deputados Estaduais, nos crimes comuns;

b) os Secretários de Estado, nos crimes comuns e de responsabilidade;

c) os Juízes de Direito, Juízes de Direito Adjunto e os membros do Ministério Público, nos crimes comuns e de responsabilidade;

d) o Procurador Geral do Estado e o Procurador Geral da Justiça, nos crimes comuns e de responsabilidade;

e) o Comandante Geral da Polícia Militar, nos crimes comuns e de responsabilidade;

f) os conflitos de competência entre as Câmaras, Conselho da Magistratura, Desembargador ou entre autoridades judiciárias e administrativas, quando participarem neles o Governador, Secretário de Estado, Magistrados ou o Procurador Geral da Justiça;

g) os conflitos de competência dos Juízes de Direito entre si e com o Conselho da Justiça Militar;

h) os mandados de segurança contra os atos do Governador, dos Secretários de Estado, da Assembléia Legislativa, sua Mesa e seu Presidente, do Tribunal de Justiça e seu Presidente, das Câmaras e seus Presidentes, do Conselho da Magistratura, do Corregedor da Justiça, dos Procuradores Gerais da Justiça e do Estado, do Tribunal de Contas e seu Presidente, dos Juízes de Direito e dos Juízes de Direito Adjunto;

i) os *habeas corpus*, quando o alegado constrangimento partir de autoridade diretamente subordinada ao Tribunal de Justiça; quando se tratar de crimes sujeito a esta mesma jurisdição, em única instância; e quando houver perigo de consumir-se a violência, antes que outro juízo possa conhecer da espécie;

j) a execução de sentença proferida em causa de sua competência facultada a delegação de atos do processo a Juiz do primeiro grau de jurisdição ou de primeira instância;

l) as habilitações incidentes nas causas de sua competência;

m) as ações recisórias de seus acórdãos;

n) a representação do Procurador Geral da Justiça visando à intervenção em Município;

o) a restauração de autos extraviados ou destruídos e outros incidentes que ocorram em processo de sua competência;

p) as revisões e reabilitações, quando as condenações a ele competirem.

II – julgar:

a) os crimes contra a honra em que forem quereladas as pessoas designadas nas letra a e b do inciso I deste artigo, bem como avocar o processo de outros indicados no caso do art. 85 do Código de Processo Civil;

b) a suspeição, não reconhecida, que se arguiu contra Desembargadores ou contra o Procurador Geral da Justiça;



c) os recursos de despacho do Presidente do Tribunal de Justiça e do Relator nos feitos de sua competência;

d) o recurso previsto no Parágrafo único do art. 557 do Código de Processo Penal;

e) os recursos e feitos em que houver arguição de inconstitucionalidade de lei, assim como de ato do poder público estadual ou municipal;

f) os recursos contra os despachos do Presidente do Tribunal, determinando que se suspenda a execução da medida liminar em mandado de segurança, ou de sentença que houver concedido;

g) os embargos de declaração, os infringentes dos seus julgados e os apostos na execução dos seus acórdãos;

h) os pedidos de arquivamento de inquérito, feitos pelo Procurador Geral da Justiça;

i) os recursos, interpostos pelos interessados, contra ato decisório das Comissões examinadoras de concurso de provas para o cargo de Juiz de Direito Adjunto;

j) os recursos contra as decisões do Conselho da Magistratura;

l) o agravo regimental do ato do Presidente contra despacho do Relator, nos processos de sua competência;

m) os pedidos de revogação de medidas de segurança que tiver aplicado.

III – adotar:

a) medidas cautelares e de segurança nos feitos de sua competência;

b) penas disciplinares, impondo-as aos Juízes, ou representação, para o mesmo fim, ao Conselho Seccional da Ordem dos Advogados e ao Conselho Superior do Ministério Público, nos casos de advogado, promotor ou procurador, respectivamente;

c) a remoção ou disponibilidade do magistrado, nos termos do art. 45 e seus incisos da Lei Complementar nº 35, de 14 de março de 1979.

IV – conhecer:

a) os incidentes de falsidade de documentos ou de insanidade mental de acusados, nos processos de sua competência;

b) o pedido de livramento condicional ou de suspensão condicional de pena, nas condenações que houver proferido.

V – elaborar, por intermédio de comissão eleita o seu regimento interno, interpretá-lo e modificá-lo;

VI – declarar a inconstitucionalidade de lei ou de ato do Poder Público;

VII – requerer a intervenção federal no Estado ao Supremo Tribunal Federal, nos termos do art. 11, § 1º, alínea a, b e c, da Constituição Federal;

VIII – conceder aposentadoria aos funcionários da sua Secretaria, de acordo com as leis sem vigor;

IX – propor à Assembléia Legislativa alterações da divisão e organização judiciária sempre que sejam necessárias;

X – organizar os Serviços da Secretaria e seus órgãos auxiliares, provendo-lhes os cargos, na forma da lei;

XI – indicar ao Governador do Estado, para nomeação, com fundamento na Lei Orgânica da Magistratura Nacional, de 14.03.79, art. 78, § 3º, os candidatos aprovados nos concursos de Juiz de Direito Adjunto, observando-se a ordem classificatória;

XII – efetuar, em sessão e escrutínio secretos as listas para que se removam e promovam Juízes, depois de cumpridas as determinações do art. 81 da Lei Orgânica da Magistratura Nacional;

XIII – compor em sessão e escrutínio secretos, dependentes de inscrição, as listas tríplices para acesso, por merecimento, de Juízes do Tribunal de Justiça, bem assim as relações para que sejam providas as vagas reservadas a advogado e membro do Ministério Público;

XIV – eleger, na segunda quinta-feira de dezembro, dos anos ímpares, pela maioria dos seus membros efetivos, por votação secreta, dentre seus Juízes mais antigos, em número correspondente aos dos cargos de direção, os titulares destes, com mandato por dois anos, proibida a reeleição. Quem tiver exercido quaisquer cargos de direção por quatro anos, ou o de presidente, não figurará mais entre os elegíveis, até se esgotarem todos os nomes, na ordem de antiguidade. É obrigatória a aceitação do cargo, salvo recusa manifestada e aceita antes da eleição;

XV – eleger por maioria de seus membros em sessão e escrutínio secretos, mediante solicitação do Tribunal Regional Eleitoral, os Desembargadores e Juízes de Direito, que devem integrá-lo, bem como os respectivos suplentes, e indicar, no mesmo caso, as listas tríplices de juristas e seus substitutos;

XVI – determinar, pelo voto de dois terços, no mínimo, de seus membros efetivos, a remoção ou a disponibilidade de Juiz de Direito, quando assim exigir o interesse público, e proceder da mesma forma relativamente a seus próprios membros, nos termos do art. 45 e seus incisos da Lei Orgânica da Magistratura Nacional;

XVII – decidir sobre pedido de permuta de Juízes de Direito e Juízes de Direito Adjunto;

XVIII – providenciar a aposentadoria compulsória de magistrados ou servidor judicial por implemento de idade ou invalidez compulsória;

XIX – licenciar, de ofício, magistrado ou servidor judicial em caso de invalidez ou incapacidade comprovadas;

XX – declarar, nos casos em que ocorrer o abandono ou a perda de cargo de magistrado ou servidor da justiça;

XXI – afastar do exercício do cargo o Juiz de Direito que, submetido a processo criminal ou administrativo, esteja removido compulsoriamente nos termos do inciso XVI deste artigo;

XXII – decidir as reclamações sobre antiguidade de Juiz de Direito contra a lista respectiva, organizada e publicada de ordem do Presidente do Tribunal;

XXIII – propor, no interesse da Justiça, o aproveitamento de Juiz em disponibilidade;

XXIV – elaborar súmulas de jurisprudência do Tribunal e publicá-las no Diário da Justiça;

XXV – regulamentar os concursos para ingresso na Magistratura, nos termos da lei (art. 78 da Lei Orgânica da Magistratura Nacional);

XVI – representar à autoridade competente, quando, em autos ou documentos de que conhecer, houver indícios de crime de ação pública;

XXVII – conceder licença e férias a seu Presidente e demais membros do Tribunal, aos Juízes de Direito, Juízes de Direito Adjunto, aos servidores e serventuários que lhe são imediatamente subordinados.

Art. 16. O Regimento Interno além dos casos ora previstos e respeitadas as leis federais, estabelecerá:

a) a organização do Tribunal Pleno, do Conselho da Magistratura, das Câmaras Reunidas e das Câmaras Especializadas, da Presidência e da Vice-Presidência do Tribunal e da Corregedoria da Justiça;

b) a estrutura e funcionamento da Secretaria do Tribunal;

c) a ordem dos serviços do Tribunal;

d) os assuntos administrativos e de ordem interna;

e) as alterações e aplicações do próprio Regimento.

Art. 17. Compete às Câmaras Reunidas:

I – processar e julgar em matéria cível:

a) os embargos infringentes dos julgados das Câmaras Especializadas e de suas decisões;

b) as ações rescisórias de seus acórdãos, das Câmaras Especializadas e das decisões dos Juízes singulares;

c) a restauração dos autos extraviados ou destruídos, em feitos de sua competência;

d) as habilitações nas causas sujeitas a seu julgamento.

II – julgar em matéria cível:

a) os embargos de declarações opostos aos seus acórdãos;

b) o recurso de despacho denegatório de embargos infringentes de sua competência;

c) os recursos, quando cabíveis, das decisões de seu Presidente;

d) as suspeições e impedimentos, nos casos que lhe competirem;

e) os recursos das decisões do Relator, em feitos de sua competência, nos casos previstos no Regimento Interno.

III – processar e Julgar em matéria criminal:

a) os pedidos de revisão;

b) os recusos das decisões do seu Presidente, na forma do Regimento Interno;

c) os pedidos de desaforamento;

d) os conflitos de competência entre as Câmaras e o Conselho de Justiça Militar do Estado.

IV – julgar em matéria criminal:

a) os embargos e declaração opostos aos seus acórdãos;

b) os recursos de decisão do Relator, quando este indeferir, liminarmente, o pedido de revisão criminal ou de interposição de embargos infringentes;

c) as suspeições e impedimentos, nos feitos de sua competência, dos membros das Câmara e do Procurador Geral da Justiça;

d) os pedidos de habeas-corpus, nos feitos submetidos ao seu julgamento, condedendo-os de ofícios, nos casos previstos em lei.

V – aplicar medidas de segurança, em decorrência de decisões proferidas em revisão criminal;

VI – executar o que for decidido nos feitos de sua competência.

VII – assentar prejulgados.

VIII – delegar poderes, quando for conveniente, nas ações rescisórias e executórias, a Juízes de Direito e Juízes de Direito Adjunto para a prática de atos que não envolvam decisão.

IX – impor penas disciplinares ao seus funcionários ou representar para idêntico fim ao Procurador Geral da Justiça e à Ordem dos Advogados, Secção do Piauí, quando se tratar de membro do Ministério Público ou de advogado, respectivamente.

X – uniformizar a jurisprudência, editando súmulas, quando possível.

XI – resolver as dúvidas de sua competência e das Câmaras Especializadas, na forma do Regimento Interno.

XII – declarar extinto o processo nos casos previstos em lei.

Art. 18. Compete a cada Câmara de acordo com sua Especialização:

I – julgar:

a) os recursos das sentenças e despachos dos Juízes de primeiro grau de jurisdição e primeira instância;

b) os recursos das decisões do Tribunal de Jùri;

c) originariamente, o habeas-corpus, quando o constrangimento provier de autoridade judiciária de primeira instância ou de Promotor Público;

d) as reclamações contra aplicação das penalidades previstas nos arts. 801 e 802 do Código de Processo Penal.

II – conhecer, em grau de recurso, dos habeas-corpus julgados pelos Juízes de primeira instância.

III – pronunciar-se e decidir sobre o despacho do Presidente da Câmara que indeferir in limine o pedido de habeas-corpus.

IV – ordenar o exame a que se refere o art. 177 do Código de Processo Penal.

V – executar, no que couber, as suas decisões, podendo delegar a Juízes de Direito a prática de atos não decisórios.

Art. 19. Adotam-se decisões das Câmaras Especializadas sempre pelo voto de três Juízes, na forma do Regimento Interno.

Art. 20. O Relator do acórdão, em caso de embargos infringentes, deve decidir se os recebe e processa, cabendo agravo do despacho denegatório.

### **Seção III Do Presidente do Tribunal**

Art. 21. Ao Presidente do Tribunal compete:

I – dirigir os trabalhos do Colegiado e presidir-lhe as sessões plenárias, fazendo cumprir o seu Regimento Interno.

II – prover o cumprimento imediato das decisões do Tribunal.

III – corresponder-se com as autoridades públicas sobre assuntos relacionados com a administração da Justiça.

IV – representar o Tribunal nas solenidades e atos oficiais, podendo, quando entender conveniente, delegar a incumbência a um ou mais Desembargadores.

V – presidir o Conselho da Magistratura.

VI – dar posse aos Desembargadores, Juízes e aos Servidores da Secretaria do Tribunal.

VII – convocar, na hipótese de falta ou impedimento de Desembargador, os respectivos substitutos dentre os Juízes da Capital, mediante sorteio público.

VIII – conhecer do pedido do recurso extraordinário, se o julgar relativamente amparado, mandar processá-lo resolvendo os incidentes suscitados.

IX – funcionar como Relator privativo, com direito a voto, nos seguintes feitos:

a) habeas corpus de julgamento da competência originária do Tribunal Pleno;

b) suspeição de Desembargador, inclusive no caso do art. 135 do Código de Processo Civil;

c) reclamação sobre antiguidade dos magistrados, apurada pelo Conselho da Magistratura;

d) os conflitos de competência entre as Câmaras Especializadas, entre as Câmaras Reunidas e o Tribunal Pleno;

e) remoção, disponibilidade, aposentadoria compulsória de magistrados serventuários e funcionários da Justiça, e de funcionário da Diretoria Geral da Secretaria do Tribunal de Justiça;

f) reversão ou aproveitamento de magistrados e demais servidores referidos na letra anterior;

g) funcionar nos pedidos de licença e férias dos Magistrados.

X – conceder prorrogação de prazo para que magistrados e demais servidores da Justiça tomem posse e entrem em exercício dos cargos;

XI – ordenar a suspensão de liminar e a execução da sentença concessiva de mandado de segurança (art.4º da Lei N° 4.348, de 26 de junho de 1954);

XII – assinar os acórdãos do Tribunal Pleno e do Conselho da Magistratura com os Juízes-Relatores e com os que expressamente tenham requerido declaração de voto;

XIII – expedir ordens que não dependem de acórdãos ou não sejam privativa competência dos Relatores;

XIV – ordenar o pagamento dos precatórios em virtude de sentença proferida contra a Fazenda Pública, estadual, municipal, nos termos do art. 117 da Constituição do Brasil e dos arts. 730, inciso I e 731 do Código de Processo Civil;

- No texto publicado, por equívoco, está grafado “das precatórias” em vez de “dos precatórios”.

XV – determinar a restauração dos feitos perdidos na Diretoria Geral da Secretaria do Tribunal;

XVI – julgar os recursos das decisões que incluam jurados na lista geral ou dela excluam;

XVII – conceder licença para casamento nos casos do art. 183, n° XVI, do Código Civil;

XVIII – encaminhar ao Governador do Estado, depois de aprovados pelo Tribunal, os pedidos de permutas de Juiz e serventuários da Justiça, quando for o caso;

XIX – comunicar à Ordem dos Advogados as faltas cometidas por advogados, provisionados e estagiários;

XX – conhecer e julgar as suspeições opostas ao Diretor Geral, Diretores e demais funcionários da Diretoria Geral da Secretaria do Tribunal;

XXI – nomear, demitir, exonerar, admitir, dispensar, transferir e aposentar os funcionários a que se faz referência no inciso anterior, inclusive preenchimento de função gratificada.

#### **Seção IV Do Vice-Presidente do Tribunal**

Art. 22. Ao Vice-Presidente do Tribunal compete:

I – presidir as sessões das Câmaras Reunidas e da Câmara Especializada de que fizer parte;

II – assinar os acórdãos com o Relator e os Juízes que requeiram declaração de voto;

III – distribuir em audiência pública, os feitos que não sejam da competência do Tribunal Pleno e do Conselho da Magistratura, inclusive os embargos, aos Relatores, na forma das leis processuais, salvo os *habeas corpus* e seus recursos, que tem distribuição imediata pelo Diretor Geral da Secretaria do Tribunal;

IV – substituir o Presidente nas faltas, férias, licenças e impedimentos.

#### **Seção V Dos Presidentes das Câmaras Especializadas**

Art. 23. A Câmara Especializada de que não faça parte o Vice-Presidente será presidida pelo Desembargador eleito dentre os seus membros.

Art. 24. Aos Presidentes das Câmaras Especializadas compete:

I – dirigir e manter a regularidade dos trabalhos e a ordem das sessões, pela forma determinada no Regimento Interno;

II – sustar a decisão em que o Juiz concluir pela inconstitucionalidade de lei ou de ato do poder público, encaminhando o processo ao Presidente do Tribunal de Justiça para julgamento pelo Colegiado;

III – redigir os resumos de julgamentos e assinar os acórdãos com os Relatores e com os Juízes que tenham feito declaração de voto;

IV – marcar dia para julgamento das causas e organizar a pauta das sessões.

## **Seção VI**

### **Do Desembargador Relator**

Art. 25. Compete ao Relator, nos feitos que lhe forem distribuídos além de outros deveres legais e do Regimento Interno:

I – processar os feitos e relatá-los;

II – resolver os incidentes relativos de acórdãos, e executar as diligências necessárias ao julgamento;

III – fazer cumprir as decisões de sua competência;

IV – lavrar o acórdão, quando não for o voto vencido, e assiná-lo juntamente com o Desembargador que houver presidido a decisão;

V – proceder ao interrogatório do acusado, reinquirir testemunhas ou determinar outras diligências, na hipótese do art. 616 do Código do Processo Penal;

VI – admitir ou negar os recursos legais das decisões que lavrar, salvo os casos de recurso extraordinário, ainda que seja voto vencido;

VII – assinar as cartas ou títulos executivos de sentença;

VIII – expedir alvará de soltura, dando imediato conhecimento ao Juiz de primeira instância no caso de decisão absolutória ou proferida em grau de recurso;

IX – denegar ou decretar prisão preventiva nos processos criminais;

X – conceder ou recusar fiança nos processos-crime;

XI – apresentar em mesa e relatar, sem direito a voto, os agravos dos seus despachos, levando os acórdãos, qualquer que seja a decisão de recurso;

XII – lançar da acusação o queixoso que deixar de comparecer (art. 561, inciso II do Código do Processo Penal) nos crimes de competência originária do Tribunal;

XIII – processar as habilitações requeridas e outros incidentes;

XIV – homologar por despacho o pedido de desistência dos recursos que lhe sejam distribuídos;

XV – homologar desistência nas ações rescisórias;

XVI – promover as diligências de atos que não dependem de julgamento, nos feitos que lhe sejam distribuídos;

XVII – decidir os pedidos originários de benefícios de justiça gratuita, nos feitos que lhe couberem por distribuição;

XVIII – encaminhar os pedidos de mandado de segurança à autoridade legítima para julgamento, quando for incompetente o Tribunal de Justiça, nos termos da legislação processual civil;

XIX – negar, liminarmente, os pedidos de revisão criminal, quando se verificar a incompetência do Tribunal ou da Câmara Especializada, ou não estiver instruído o processo se for desaconselhável aos interessados da justiça que se apensem os autos originais;

XX – lavrar, em forma de acórdãos, as decisões tomadas nos processos administrativos.

## **Seção VII**

### **Do Conselho da Magistratura**

Art. 26. O Conselho da Magistratura, órgão disciplinar, composto de três membros, o Presidente, o Vice-Presidente do Tribunal e o Corregedor da Justiça, tem como órgão superior o Tribunal Pleno e as suas atribuições serão estabelecidas no Regimento Interno.

### **Seção VIII** **Da Corregedoria da Justiça**

Art. 27. A Corregedoria Geral da Justiça, que funciona na sede do Tribunal, órgão de fiscalização disciplinar, orientação, controle e instrução dos serviços forenses e administrativos da justiça de primeiro grau, tem competência em todo o Estado e é exercido por Desembargador.

§ 1º O Desembargador, no exercício do mandato de Corregedor Geral da Justiça, será dispensado de sua função judicante normal, obrigando-se a comparecer às sessões plenárias do Colegiado, para decidir sobre a nomeação, promoção, permuta, disponibilidade de juizes e sobre matéria de natureza administrativa e constitucional.

§ 2º Faz-se a escolha do Corregedor Geral e do Vice-Corregedor Geral da Justiça juntamente com as dos demais titulares de funções de direção do Poder Judiciário.

§ 3º O Vice-Corregedor Geral da Justiça só se afastará de suas funções ordinárias pelo período que estiver substituindo o Corregedor Geral da Justiça.

§ 4º O Vice-Corregedor Geral da Justiça presidirá as sessões da Câmara que integrar se dela não participar o Vice-Presidente do Tribunal de Justiça.

§ 5º Ocorrendo as vagas de Corregedor Geral e Vice-Corregedor Geral da Justiça, far-se-á eleição dos novos titulares, completarão o período.

§ 6º Se o prazo que faltar para completar o mandato for inferior a um ano, o novo Corregedor Geral ou o Vice-Corregedor Geral da Justiça poderão concorrer para o período seguinte.

- [Artigo com redação dada pelo art. 1º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.](#)

Art. 28. Sem prejuízo das correições ordinárias e anuais, que os Juizes se obrigam a fazer nas comarcas, o Corregedor Geral da Justiça deve realizar uma de caráter geral, anualmente, em pelo menos dez comarcas, sem que se contem as correições extraordinárias determinadas pelo Conselho da Magistratura ou pelo Tribunal Pleno.

§ 1º As correições ordinárias e anuais, de realização obrigatória pelos Juizes, nas respectivas comarcas ou varas, consistirão na inspeção assídua e severa dos cartórios, delegacias de polícia, estabelecimentos penais e demais repartições que tenham relação direta com os serviços judiciais e sobre a atividade dos auxiliares e funcionários da Justiça que lhes sejam subordinados.

§ 2º Sujeitam-se à correição os atos dos Juizes, serventuários e funcionários da Justiça, inclusive as escritanias das Varas da Fazenda Pública.

- [Artigo com redação dada pelo art. 1º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.](#)

Art. 29. O Corregedor Geral da Justiça será substituído, quando o prazo de afastamento for superior a trinta dias, em suas férias, licenças e impedimentos, pelo Vice-Corregedor Geral da Justiça, enquanto este terá como substituto o Desembargador que lhe seguir em ordem de antiguidade, excluídos os que exercem mandatos no Tribunal Regional Eleitoral.

Parágrafo único. O Regimento Interno do Tribunal de Justiça, o Regimento Interno da Corregedoria Geral da Justiça e o Código de Normas da Corregedoria Geral da Justiça poderão conferir ao Vice-Corregedor Geral da Justiça outras atribuições específicas entre aquelas conferidas ao Corregedor Geral da Justiça.

- Artigo com redação dada pelo art. 1º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.

Art. 30. O Corregedor Geral da Justiça e o Vice-Corregedor Geral da Justiça serão auxiliados por dois Juízes-Corregedores, na forma do § 2º, do art. 41, da Lei nº 3.716, de 12 de dezembro de 1979.

§ 1º Os atos do Corregedor Geral da Justiça são expressos por despacho, ofício, portaria, circular, provimento e cota marginal nos autos, definidos no Regimento Interno do Tribunal de Justiça e no Código de Normas da Corregedoria Geral da Justiça.

§ 2º Os Juízes-Corregedores terão suas atribuições definidas no Regimento Interno do Tribunal de Justiça.

- Artigo com redação dada pelo art. 1º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.

Art. 31. Qualquer pessoa pode denunciar, por escrito, ao Corregedor Geral da Justiça, excessos, irregularidades ou omissões das autoridades judiciárias, seus auxiliares, serventuários e funcionários da Justiça, competindo-lhe encaminhar ao Conselho da Magistratura os processos respectivos, quando estes não estiverem submetidos ao seu julgamento.

§ 1º Se o ato, por sua gravidade, for praticado por membro ou funcionário do Ministério Público ou da Polícia Civil, o Corregedor Geral da Justiça dele dará ciência ao Procurador Geral de Justiça ou ao Secretário da Segurança Pública, conforme o caso, bem assim ao Presidente do Tribunal.

§ 2º Após a apuração de denúncia, prevista no *caput* deste artigo, e sem prejuízo da pena disciplinar que houver aplicado, o Corregedor Geral da Justiça encaminhará ao Procurador Geral de Justiça as provas ou indícios que coligir sobre a existência de crime ou contravenção, para que se positivem responsabilidades.

- Artigo com redação dada pelo art. 1º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.

Art. 32. Sujeitam-se à correição os atos dos Juízes, serventuários e funcionários da Justiça, inclusive as escritanias das Varas da Fazenda Pública.

- Artigo com redação dada pelo art. 1º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.

Art. 33. O Corregedor Geral da Justiça, nos exames que fizer, verificará se as determinações dos Juízes locais foram cumpridas e aplicará, em caso negativo, as penas disciplinares cabíveis ou promoverá a responsabilidade dos culpados.

Parágrafo único. Para esta verificação, o Corregedor Geral da Justiça solicitará informações ao Juiz local.

- Artigo com redação dada pelo art. 1º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.



Art. 34. Finda a correição, o Corregedor Geral da Justiça, presentes os Juízes, membros do Ministério Público, serventuários e funcionários da Justiça convocados, dará conhecimento das cotas e despachos proferidos nos autos, nos livros e nos papéis examinados, fará a leitura dos provimentos expedidos e, ainda, determinará a lavratura, pelo Secretário designado, de ata constando as ocorrências, exames, irregularidades, medidas impostas, com transcrição das adotadas instruções e determinações administrativas e a assinará com as autoridades presentes.

Parágrafo único. Os provimentos relativos a atos praticados por Juiz não devem constar de ata e lhe são transmitidos em caráter reservado pelo Corregedor Geral da Justiça.

- Artigo com redação dada pelo art. 1º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.

Art. 35. As correições abrangem, também, sindicâncias, sob reserva, a respeito da conduta funcional e moral das autoridades judiciárias, membros do Ministério Público, advogados, serventuários e funcionários da Justiça.

Parágrafo único. As faltas dos membros do Ministério Público e dos advogados serão comunicadas aos órgãos respectivos a que, por lei, se subordinam disciplinarmente.

- Artigo com redação dada pelo art. 1º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.

Art. 36. As cotas escritas, pelo Corregedor Geral da Justiça, nos livros, autos e papéis, constituem advertência para as emendas e ressalvas feitas; e despachos que ordenarem diligência e provimentos serão, dados para os casos futuros, tendo em vista evitar a prática abusiva ou ilegal, com a cominação de pena se houver.

- Artigo com redação dada pelo art. 1º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.

## **Seção IX Dos Juízes de Direito**

Art. 37. Cada Comarca tem pelo menos, um Juíz de Direito.

Art. 38. Nas Comarcas onde houver mais de um Juiz de Direito, eles se substituem, em ordem numérica, nas suas faltas e impedimentos.

Art. 39. A competência dos Juízes na Comarca onde houver mais de um fixa-se em cada processo pela distribuição, respeitadas as privatividades e a prevenção inscrita nos art. 106 e 219, do Código do Processo Civil.

Art. 40. Compete ao Juíz de Direito:

I – processar e julgar, na comarca:

- a) as causas cíveis;
- b) medidas cautelares para que sirva de documento;
- c) as falências, concordatas e demais processos destas resultantes e derivados;
- d) as causas de dissolução e liquidação das sociedades cíveis e comerciais bem como a verificação de haveres, não se tratando de firma individual, em caso de morte do comerciante;
- e) os executivos fiscais e os processos de curadoria;

f) as causas de separação judicial, divórcio, nulidade e de anulação de casamento;  
g) os feitos de crimes comuns e de contravenção não expressamente não atribuídos a outra jurisdição e os referentes a funcionários públicos que não tenham foro privativo, nos crimes de responsabilidade ou com estes conexos;

h) os crimes cometidos com abuso de liberdade de imprensa, com obediência à lei respectiva;

i) restauração de autos extraviados ou distribuídos quando afetos ao seu Juízo;

j) os crimes de falência e os que lhe são equiparados;

l) as execuções criminais, decidindo os incidentes salvo quando à graça, ao indulto e à anistia.

II – processar:

a) as deserções de apelação nas causas de sua competência;

b) os crimes de competência do Júri, proferindo nos feitos, quando for o caso, despachos de pronúncia ou impronúncia, ou sentença absolutória liminar, quando exclua o crime ou isente da pena, o réu, com recurso de ofício, nesta última hipótese, para a instância superior.

III – julgar:

a) as execuções de incompetência.

b) as causas que, resultantes direta ou indiretamente das compreendidas em sua competência, não estejam por esta lei reservadas a jurisdição de outro Juízo.

c) as suspeições declaradas pelos Promotores, serventuários e funcionários da Justiça, ou contra eles arguidas e que não tenham sido reconhecidas, nos feitos, em que lhe couber o processo e julgamento;

d) os embargos de declaração opostos às sentenças que proferir.

IV Homologar as sentenças arbitrais, no âmbito de sua alçada.

V – cumprir as precatórias procedentes de outras comarcas do Estado ou do País e dos Juízes Federais; as determinações do Supremo Tribunal Federal; do Conselho Nacional da Magistratura; do Tribunal Federal de Recursos; do Tribunal de Justiça e do Conselho da Magistratura, bem assim as requisições legais, ressalvada a competência do Diretor do Forum, onde houver.

VI – rubricar os balanços comerciais.

VII – exercer:

a) as atribuições relativas à habilitação e celebração de casamento;

b) as prerrogativas da ordem administrativa que lhe são conferidas pelo Código de Processo Civil e pelas leis federais e estaduais.

VIII – interpor, quando for o caso, recurso de ofício ou pedir julgamento em segundo grau de jurisdição.

IX – executar as suas próprias sentenças e as proferidas por Juiz superior, quando lhe forem delegados os necessários poderes.

X – suprir:

a) a outorga conjugal, nos casos da lei;

b) a denegação de consentimento para núpcias do menor, quando julgar conveniente e legítimo, com recursos para a instância superior.

XI – resolver as dúvidas suscitadas pelo escrivães, tabeliães, oficiais de registro geral, especial e cível, e dos demais funcionários do foro.

XII – requisitar:

a) a força necessária à autoridade competente, para o cumprimento das sentenças judiciais e realização de diligência, na conformidade da lei;

b) informações e certidões aos órgãos públicos julgados necessários para instruir processos ou constituir provas de que as partes alegarem.

## XIII – conceder:

- a) suprimimento de idade a menor e licença para venda de bens imóveis que a ele pertençam;
- b) prorrogação de prazo para que se ultime inventário;
- c) habeas-corpus contra auto de autoridade administrativa ou policial, salvo quando coautora for qualquer das mencionadas no artigo 15, desta lei;
- d) fiança, julgando os recursos interpostos do respectivo arbitramento feito por autoridade policial;
- e) liberdade provisória, nos termos do art. 350, do Código de Processo Penal;
- f) o benefício da Justiça gratuita, de plano no curso da lide, nomeando advogado para o beneficiário;
- g) - licença, salvo a especial, e férias aos Juízes de Paz, serventuários e funcionários da sua jurisdição, passando a competência para o Diretor do Forum, onde houver.

## XIV – determinar:

- a) de ofício ou a pedido da parte ofendida, a supressão de injúria, calúnia e difamação nos autos que lhe estejam sujeitos, adotando as providências punitivas contra o autor ou comunicando o fato à Ordem dos Advogados, quando for o caso;
- b) o fornecimento deste que requerido, de certidão de processo que deve correr ou tenha corrido em segredo da justiça;
- c) a entrega de bens de órgãos e ausentes;
- d) a lavratura de auto de prisão em flagrante, e exame de corpo de delito e o mandado de busca e apreensão;
- e) o aditamento da denúncia, nos crimes de ação pública, quando for o caso;
- f) a entrega ao membro do Ministério Público de certidões de documentos necessários à promoção de responsabilidade, quando em autos ou papéis sujeitos à sua competência, se positivar crime, inicio deste, ou ato ilegítimo, de que caiba ação pública.

XV – policiar o Forum, quando não houver Diretor, mandando prender em flagrante qualquer pessoa, que falte com respeito à sua autoridade de magistrado no exercício do cargo, ou que trave luta corporal com outrem, ou que desacate autoridade ou servidor da justiça, com a lavratura de auto de flagrante respectivo, para o competente procedimento legal.

## XVI – conhecer:

- a) das reclamações contra a cobrança de custas indevidas, mandando que se restituem os excessos, sem prejuízo das penalidades contra o culpado;
- b) das causas extintivas da punibilidade nos crimes que processar.

## XVII – nomear:

- a) tutor ou curador aos órgãos, ausentes e interditos, bem assim testamentários e inventariantes, e removê-los nos termos da lei;
- b) representante do Ministério Público, serventuário ou funcionário da Justiça ad hoc, nas faltas ou impedimentos eventuais dos titulares, e no caso de vacância do ofício ou cargo (competência do Diretor do Forum onde houver);
- c) escreventes juramentados com cartórios, compromissando-os mediante indicação do serventuário e atributos de idoneidade e habilitação;
- d) curador à lide, nos casos expressos no Código de Processo Civil (competência do Diretor do Forum, onde houver).

## XVIII – decretar:

- a) a suspensão e a perda do pátrio poder;
- b) a prisão preventiva.

XIX – proferir sentença sobre fiança ou cauções prestadas pelas partes.

XX – celebrar os casamentos na sede da Comarca.

XXI – decidir:

- a) sobre o Mandado de Segurança, nos casos de sua competência;
- b) sobre a posse e guarda de filhos menores, no caso de separação judicial, divórcio, nulidade ou anulação de casamento;
- c) sobre justificação, perícia e outras medidas necessárias, nos processos que tiver de julgar;
- d) as reclamações das partes a respeito de atos de competência de serventuários e funcionários da Justiça na sua jurisdição.

XXII – abrir:

- a) testamento dando-lhe compromisso e tomando conta dos testamenteiros;
- b) os livros dos serventuários da Justiça, nas comarcas do interior, rubricando-os e encerrando-os, quando for o caso;
- c) correição, ao menos uma vez por ano, nos cartórios da Comarca, do que enviará relatório circunstanciado das medidas que adotar ao Corregedor da Justiça.

XXIII – fazer passar de ofício ordem de habeas-corpus, quando tiver conhecimento, em feito pedente, de que alguém sofra, ou se ache ameaçado de sofrer violência ou coação em sua liberdade de locomover-se.

XXIV – relaxar prisão ou detenção de qualquer pessoa, quando ilegal, e promover a responsabilidade do autor do abuso.

XXV – qualificar os jurados, procedendo o sorteio dos que tenham de servir nas sessões do Júri.

XXVI – aplicar:

- a) medidas de segurança;
- b) a lei nova, por simples despacho, a requerimento da parte ou Ministério Público, quando o processo lhe estiver afeto, ou a última sentença for por ele proferida ou pelo Tribunal de Justiça;
- c) penas disciplinares aos serventuários e funcionários da Justiça de seu Juízo, passando a competência ao Diretor do Forum, onde houver.

XXVII – lançar o queixoso da acusação, nos crimes de sua competência ou do Tribunal do Júri.

XXVIII – presidir as sessões do Tribunal do Júri e receber os recursos das suas decisões.

XXIX – visitar as prisões para dar audiência a presos, informar-se do estado deles e pedir às autoridades as providências que se fizerem necessárias.

XXX – impor multa ao Promotor Público, quando oferecer libelo fora do prazo legal, observando o disposto no artigo 419, do Código de Processo Penal.

XXXI – aprovar os estatutos das fundações ou a sua reforma, quando denegada pelo Ministério Público, se a medida for legal.

XXXII – punir as testemunhas faltosas ou desobedientes.

XXXIII – realizar visita de inspeção, pelo menos anualmente, aos termos judiciários da sua comarca, fiscalizando o cumprimento das leis e das recomendações superiores por parte dos Juizes de Paz, serventuários e funcionários da Justiça, e ainda verificando as condições da Cadeia Pública.

XXXIV – cabe ainda ao Juíz o exercício de qualquer atribuição que lhe seja cometida pelas leis em vigor.

Art. 41. As trinta e quatro Varas da Comarca de Teresina, de entrância final, cada uma com um Juiz de Direito, repartem-se em:

- *Caput* com redação dada pelo art. 1º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.

- A mesma redação do *caput* foi repetida pelo art. 1º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.

I – dez Varas Cíveis, por distribuição, denominadas numericamente de 1ª a 10ª;

- Incisos I com redação dada pelo art. 1º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.

II – quatro Varas da Fazenda Pública, sendo duas por distribuição, denominadas, numericamente, de 1ª e 2ª, e as 3ª e 4ª Varas, também por distribuição, exclusivas de Execuções Fiscais e demais ações de natureza tributária com a seguinte competência:

- Inciso II com redação dada pelo art. 1º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.

a) a 3ª Vara da Fazenda Pública possui competência privativa para as execuções e ações de natureza tributária referentes ao Município de Teresina;

b) a 4ª Vara da Fazenda Pública possui competência privativa para as execuções e ações de natureza tributária referentes ao Estado do Piauí.

III – uma Vara de Registros Públicos e para dirimir conflitos fundiários e questões agrárias no Município de Teresina;

- Inciso III com redação dada pelo art. 1º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.

III-A – *(Revogado pelo art. 6º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.)*

IV – seis Varas de Família e Sucessões, competentes privativamente, por distribuição entre elas, para presidir a celebração de casamento;

- Inciso IV com redação dada pelo art. 1º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.

IV-A – *(Revogado pelo art. 6º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.)*

V – duas Varas da Infância e da Juventude, sendo a 1ª Vara exclusiva para os processos de natureza cível e a 2ª, para os processos de natureza criminal;

- Inciso V com redação dada pelo art. 1º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.

VI – nove Varas Criminais, por distribuição, conforme especificação abaixo:

- Inciso I com redação dada pelo art. 1º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.

a) 1ª Vara Criminal, de competência genérica;

b) 2ª Vara Criminal, privativa de execuções criminais e corregedoria de presídios;

c) 3ª Vara Criminal, de competência genérica;

d) 4ª Vara Criminal, de competência genérica;

e) 5ª Vara Criminal, de competência exclusiva para o cumprimento de cartas precatórias, cartas rogatórias, cartas de ordem e para o processo e julgamento das causas decorrentes de violência doméstica e familiar contra a mulher;

f) 6ª Vara Criminal dos crimes de trânsito, de imprensa, crimes contra a ordem tributária, contra a economia popular, ordem econômica e consumidor;

g) 7ª Vara Criminal, privativa de delitos sobre tráfico de drogas e de crimes cometidos contra crianças e adolescentes;

h) 8ª Vara Criminal, privativa de delitos praticados contra idosos e portadores de deficiência;

i) 9ª Vara Criminal, com competência para o julgamento de crimes militares cometidos contra civis e de ações judiciais contra atos disciplinares militares.

VII – duas Varas Criminais, denominadas de 1ª e 2ª Varas do Tribunal Popular do Júri, de competência exclusiva, por distribuição, para julgamento dos crimes dolosos contra a vida.

- Inciso VII com redação dada pelo art. 1º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.

VIII – *(Revogado pelo art. 6º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.)*

IX – *(Revogado pelo art. 6º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.)*

X – *(Revogado pelo art. 6º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.)*

§ 1º Haverá, ainda, em Teresina, dez Juizados Especiais Cíveis e Criminais, que terão como titulares Juízes de Direito de entrância final, com atribuições definidas nesta Lei e legislação especializada.

- § 1º acrescentado pelo art. 2º da Lei n. 5.204, de 07/08/2001, publicada no DOE nº 152, de 08/08/2001, pp. 5/7 e com redação dada pelo art. 1º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.

§ 2º Haverá, ainda, em Teresina, dois Juízes Corregedores que terão como titulares Juízes de entrância final, convocados pela Douta Corregedoria Geral da Justiça e homologados pelo Tribunal de Justiça do Estado do Piauí.

- § 2º acrescentado pelo art. 2º da Lei n. 5.204, de 07/08/2001, publicada no DOE nº 152, de 08/08/2001, pp. 5/7 e com redação dada pelo art. 1º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.

§ 3º Para compor a equipe multidisciplinar, com atuação na 5ª Vara Criminal, prevista na Lei Federal nº 11.340/2006, de 07 de agosto de 2006 (Lei Maria da Penha), ficam igualmente criados os seguintes cargos na estrutura organizacional do Poder Judiciário do Piauí:

- § 3º acrescentado pelo art. 1º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.

I – 02 (dois) cargos de assistente social, de provimento efetivo;

II – 02 (dois) cargos de psicólogo, de provimento efetivo;  
III – 02 (dois) cargos de médico, com especialização em psiquiatria, de provimento efetivo.

Art. 42. *(Revogado pelo art. 6º da Lei Complementar n. 97, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 1/2.)*

Art. 43. Na Comarca de Parnaíba, a competência das Varas determina-se por distribuição, tanto no cível como no crime, excetuando os processos de Registros Públicos, execuções criminais, acidentes do trabalho, de competência do Tribunal do Júri e da Fazenda Pública, privativo da 1ª., os relativos a menores e família, privativos da 2ª., os comerciais, os de crime previstos na lei de imprensa, os contra a economia popular e os de precatórias em geral, privativo da 3ª., e os interditos, ausentes, provedoria, resíduos e os derivados destas matérias, privativos da 4ª.

Art. 44. Nas Comarcas de Floriano, Picos e Campo Maior, a competência fixa-se por distribuição, no cível e no crime, com estas indicações privativas; a 1ª., nos Registros Públicos, Acidentes do Trabalho, processos de competência do Tribunal do Júri, Execuções Criminais e Feitos da Fazenda Pública Municipal e de entidade autárquica respectivas a 2ª., nos casos de menores, casamento, interditos, ausentes, provedoria e resíduos e nos derivados de tais matérias nas lides comerciais e precatórias em geral.

Art. 45. Nas Comarcas onde houver mais de uma vara excetuada a capital a competência para aplicação de penas disciplinares aos serventuários e funcionários da Justiça cabe ao Diretor do Forum, ressalvados do os casos em que os outros Juízes possam exercê-la, nos feitos correntes nas respectivas varas.

Art. 46. Os Juízes de Menores nas Comarcas da Capital, Parnaíba, Floriano, Picos e Campo Maior, a lém das atribuições que lhe são conferidas pelo Código respectivo e leis de amparo a menores, têm jurisdição privativa sobre os menores abandonados, desassistidos e de conduto anti-social, com menos de 18 anos de idade, para a sua assistência, proteção e segurança.

## **Seção X**

### **Dos Juízes de Direito Adjunto**

Art. 47. Os Juizes de Direito Adjunto, tem a incumbencia especial de substituir os Juizes de Direito, nas suas férias, impedimentos, faltas e suspeições. Os adjuntos são numerados ordinalmente e nomeados com as mesmas garantias dos magistrados, ressalvadas as restições legais.

Art. 48. Os Juizes de Direito Adjunto destinam-se aos serviços de interior funcionando nas zonas de igual numeração e com residência na Comarca-sede. Art.49-Atendida a conveniência do serviço declarado pelo Tribunal e por designação do Presidente, poderá o Juiz ter serventia em qualquer zona ou Comarca.

Parágrafo único. No ato designatória deferem-se as atribuições do Juiz.

Art. 50. O Juiz de Direito Adjunto, como auxiliar, nos termos do artigo 144, § 1º, letra b da Constituição da República Federativa do Brasil, tem competência para decidir as questões de pequeno valor e as criminais em que não seja cominada pena de reclusão.

§ 1º Para os efeitos deste artigo não se considerem de pequeno valor:

- a) as ações de estado e capacidade;
- b) as causas que excedam 50 vezes o maior valor de referência;
- c) as ações expropriatórias;
- d) as questões trabalhistas;
- e) o processo de insolvência.

§ 2º Nos processos por crime punidos com pena de reclusão, o Juiz de Direito Adjunto não pode decidir, mas lhe é lícito presidi-lo e instruí-lo.

## **Seção XI Do Tribunal do Júri**

Art. 51. O Tribunal do Júri, cuja organização e competência são as definidas na legislação processual penal, funciona na sede das comarcas.

§ 1º Reúne-se o Tribunal do Júri, obrigatoriamente no primeiro dia útil dos meses de março, junho, setembro e dezembro nas comarcas do interior e na Capital nos meses de fevereiro, abril, junho, agosto, outubro e dezembro, na primeira quinzena.

§ 2º Quando por motivo de força maior, ou falta de processos preparados, dada ciência ao Conselho da Magistratura, não se realizar na data fixada, a reunião deve efetuar-se no primeiro dia útil do mês seguinte.

Art. 52. Os Juízes de Direito Adjunto não podem presidir o Tribunal do Júri, quando estiverem como auxiliares.

## **Seção XII Da Justiça Militar**

Art. 53. *(Revogado pelo art. 21 da Lei Complementar n. 98, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, pp. 2/3.)*

## **Seção XIII Dos Juízes de Paz**

Art. 54. Os Termos Judiciários tem um Juiz de Paz com investidura limitada a quatro anos e competência para habilitação e celebração de casamentos. Substituindo-o haverá suplentes, nomeados por igual período.

§ 1º O Juiz de Paz deve residir na sede do Município, ou povoado e perceberá a remuneração que a lei fixar.

§ 2º Para officiar nas habilitações de casamento há um representante do Ministério Público e um escrivão do Registro Civil.

§ 3º O Juiz de Paz e seus suplentes serão nomeados pelo Governador do Estado, dentre cidadãos de reconhecidas idoneidade, indicados em lista tríplice pelo Presidente do Tribunal de Justiça, ouvido o Juiz de Direito da Comarca. Nomeado o Juiz, os demais integrantes da lista passam a primeiro e segundo suplentes, na ordem de colocação.

Art. 55. São requisitados para provimento do cargo de Juiz de Paz e seus suplentes:

- a) cidadania brasileira;
- b) idade superior a 25 anos;
- c) idoneidade moral comprovada;
- d) aptidão intelectual;



- e) gozo dos direitos políticos e civis e quitação com o serviço militar;
- f) inscrição na Zona Eleitoral;
- g) residêndia superior a um ano no município;
- h) não pertencer a órgão de Direção ou ação de Partido Político.

Art. 56. Findo o quadriênio de serventia, o Juiz de Paz e seus suplentes devem considerar-se como reconduzido aos cargos por igual período, caso não tenha m sido nomeados os substitutos, no prazo de trinta dias; em caso de substituição, aguardam, nas funções, a posse dos sucessores.

Art. 57. Compete ao Juiz de Paz:

- a) remeter ao Juiz de Direito, para exame de sua regularidade, os processos de habilitação de casamento, depois de decorrido o prazo do edital, expedindo-se a respectiva certidão;
- b) designar dias, hora, e lugar para que se celebre o casamento;
- c) informar ao Juiz de Direito a vaga de Oficial de Registro Civil que ocorra ou exista, para as providências cabíveis;
- d) nomear oficial do Registro Civil ad hoc nos impedimentos e faltas ocasionais do efetivo e seus substitutos legais;
- e) fixar dia, hora e lugar para audiência semanal do casamento.

#### **Seção XIV Do Conselho Penitenciário**

Art. 58. O Conselho Penitenciário rege-se pelo disposto no Decreto Federal n° 16.665, de 06 de novembro de 1924, pelo Decreto Estadual n°530, de 11 de novembro de 1963 e pelas leis posteriores.

### **LIVRO II TÍTULO III DISPOSIÇÕES COMPLEMENTARES**

#### **Seção XV Das Nomeações e Promoções**

Art. 59. A nomeações correspondentes ao provimento inicial em cargo de carreira ou isolados.

Art. 60. Faz-se a nomeação:

- I – vitaliciamente para os cargos de magistratura;
  - II – *(Revogado pelo art. 10 da Lei Complementar n. 88, de 05/09/2007, publicada no DOE n° 170, de 06/09/2007, pp. 1/2.)*
  - III – de forma temporária, para o cargo de Juíz de Paz;
  - IV – em substituição, no afastamento por prazo certo de ocupante de cargo de serventuário ou de cargo isolado e de nomeação efetivas;
  - V – a título de precariedade ou *ad hoc*, quando a lei autorizar.
- Parágrafo único. Veda-se nomeação em caráter interino.

Art. 61. O Juiz de Direito Adjunto, adquire vitaliciedade após dois anos de efetico exercício, nos termos do artigo 22, inciso II, letra c da lei complementar n° 35 de 14.03.1979.

Art. 62. O ingresso na Magistratura de carreira verifica-se por nomeação, após concurso de provas e títulos, de que participe o órgão seccional da Ordem dos Advogados do Brasil.

§ 1º É legítimo que se exija dos candidatos, quando oportuno, provas de haver feito o curso oficial de preparação para a Magistratura.

§ 2º Os candidatos devem ser submetidos a severa sindicância nos aspectos moral e social de sua personalidade e exame físico de saúde, conforme o dispuser a lei.

§ 3º Indica-se para nomeação pela ordem classificatória, candidato em número correspondente às vagas.

Art. 63. O Juiz, no ato da posse, apresenta declaração pública de seus bens, e presta compromisso de desempenhar com integridade as funções do cargo.

Art. 64. As promoções obedecem aos seguintes critérios:

a) apura-se na entrância a antiguidade e o merecimento, tornando-se obrigatório a do Juiz que figurará pela quinta vez consecutiva em lista triplice. Em caso de empate quanto ao tempo de serviço, tem precedência o mais antigo na carreira.

b) para compor lista tríplice, apura-se o merecimento da entrância, que é aferido com a prevalência de ordem objetiva, na forma prescrita pelo Tribunal de Justiça, tendo-se em conta a conduta do Juiz, sua operosidade no exercício do cargo, número de vez que tenha figurado na escolha, tanto para a circunscrição judiciária a prover como para as anteriores, bem como resultado de curso de aperfeiçoamento que tenha frequentado.

c) o Tribunal de Justiça recusa a promoção do Juiz mais antigo pelo voto da maioria absoluta de seus membros, no mínimo, repetindo-se o escrutínio até que se faça a escolha.

d) somente após dois anos de exercício na instância pelo Juiz ser promovido, salvo se não houver quem não aceite o lugar ou se o tribunal recusar candidatos que estejam habilitados quanto ao prazo que hora se fixa.

Art. 65. Para cada vaga destinada ao preenchimento por promoção abri-se-á inscrição distinta, sucessivamente, com a indicação da Comarca ou Vara a ser provida.

Parágrafo único. Ultimando o preenchimento das vagas, se mais de uma dava ser provida por merecimento, a lista conterà o número de Juizes igual ao das vagas mais dois.

Art. 66. Após a ocorrência da vaga no primeiro ou segundo grau do Poder Judiciário será publicado edital para o seu preenchimento no prazo de 15 (quinze) dias.

- *Caput e parágrafo único com redação dada pelo art. 1º da Lei Complementar n. 96, de 10/01/2008, publicada no DOE nº 08, de 11/01/2008, p. 1.*

Parágrafo único. Os editais serão numerados, publicados e julgados na ordem de vacância, respeitando-se a alternância dos critérios de merecimento e antiguidade em razão da ordem seqüencial, na respectiva entrância, e por modalidade de provimento.

Art. 67. O acesso dos Juizes de Direito ao Tribunal de Justiça, faz-se alternadamente, por antiguidade e por merecimento.

§ 1º Para a vaga de antiguidade, o Tribunal, em sessão e escrutínio secreto, decide preliminarmente, se o Juiz mais antigo da última instância deve ser indicado e se houver recurso, pelo voto da maioria absoluta dos Desembargadores, no mínimo, repete-se a votação até se fixar o escolhido.

§ 2º Para a de merecimento o Tribunal encaminha ao Poder Executivo a lista tríplice dos Juizes que obtenham o maior número de votos, obedecendo as prescrições legais.

§ 3º Na vaga de merecimento o Tribunal julga, precipuamente, os atributos morais, a cultura jurídica, a operosidade no exercício do cargo, apurada estas nos mapas mensais da Comarca ou da Vara e nos processos em que tenham o Juiz funcionado; e ainda a conduta na vida particular e pública, os trabalhos doutrinários publicados, os estudos em curso de aperfeiçoamento e as comissões exercida.

§ 4º Apura-se os requisitos morais e funcionais pelos assentamentos do candidato, pela informação do Conselho de Magistratura e da Corregedoria da Justiça e pela ciência própria de cada Desembargador.

§ 5º Nenhum Juiz integra lista de promoção caso não esteja em dia com os trabalhos da Comarca ou Vara.

Art. 68. Os candidatos a Desembargador requerem inscrição cumprindo as exigências legais, ao Presidente do Tribunal pelo prazo de quinze dias, contando o edital publicado no Diário da Justiça.

Art. 69. Preenche-se um quinto do Tribunal por advogado no efetivo exercício da profissão e por membros do Ministério Público, de notório saber jurídico e reputação ilibada, com dez anos, pelos menos, de prática forense, depois de formados, dos quais os cinco últimos na classe a que pertencer a vaga, observado o artigo 100 da Lei Orgânica da Magistratura Nacional.

§ 1º Somente membros do Ministério Público de carreira podem concorrer ao preenchimento de vaga.

§ 2º O preenchimento faz-se alternadamente, uma para advogado e outra para membro do Ministério Público, não podendo ser votado para a vaga daquele o integrante deste, ainda que exerça a advocacia.

§ 3º Para cada vaga, o Tribunal com presença de, pelos menos, dois terços de seus membros efetivos, em sessão e escrutínio secretos, vota a lista tríplice respectiva, encaminhado-a ao Governador do Estado, para a nomeação.

Art. 70. Os Candidatos a Juiz de Direito Adjunto devem provar:

I – ser brasileiro nato;

II – *(Revogado pelo art. 10 da Lei Complementar n. 88, de 05/09 2007, publicada no DOE nº 170, de 06/09/2007, pp. 1/2.)*

III – ser doutor ou bacharel em Direito por Universidade ou Faculdade reconhecida;

IV – pelo menos, dois anos de prática forense, seja como advogado, estagiário, membro do Ministério Público, ou Consultor Jurídico ou Assessor Jurídico em órgãos oficiais;

V - integridade moral;

VI – cumprimento das obrigações militares e eleitorais;

VII – gozo dos direitos políticos;

VIII – capacidade física e mental, em parecer de junta médica do Estado;

IX – ausência de antecedentes criminais, em folha corrida policial e judiciária.

§ 1º A idoneidade moral dos candidatos deve ser livremente julgada pelo Tribunal, em sessão secreta.

§ 2º *(Revogado pelo art. 10 da Lei Complementar n. 88, de 05/09/2007, publicada no DOE nº 170, de 06/09/2007, pp. 1/2.)*

## **Seção XVI**

### **Do Concurso Para Magistratura**

Art. 71. O ingresso na carreira da Magistratura Estadual, cujo cargo inicial será o de Juiz Substituto, se dará através de concurso público de provas e títulos, com a participação da Ordem dos Advogados do Brasil em todas as suas fases, e obedecerá, nas nomeações, à ordem de classificação.

Parágrafo único. O Tribunal de Justiça do Estado do Piauí regulamentará o concurso através de Resolução.

- *Artigo com redação dada pela Lei n. 5.211, de 04 de outubro de 2001, publicada no DOE nº 196, de 10/10/2001, p. 2.*

*Art. 72 (Revogado pelo art. 3º da Lei n. 5.211, de 04 de outubro de 2001, publicada no DOE nº 196, de 10/10/2001, p. 2.)*

*Art. 73 (Revogado pelo art. 3º da Lei n. 5.211, de 04 de outubro de 2001, publicada no DOE nº 196, de 10/10/2001, p. 2.)*

*Art. 74 (Revogado pelo art. 3º da Lei n. 5.211, de 04 de outubro de 2001, publicada no DOE nº 196, de 10/10/2001, p. 2.)*

*Art. 75 (Revogado pelo art. 3º da Lei n. 5.211, de 04 de outubro de 2001, publicada no DOE nº 196, de 10/10/2001, p. 2.)*

*Art. 76 (Revogado pelo art. 3º da Lei n. 5.211, de 04 de outubro de 2001, publicada no DOE nº 196, de 10/10/2001, p. 2.)*

*Art. 77 (Revogado pelo art. 3º da Lei n. 5.211, de 04 de outubro de 2001, publicada no DOE nº 196, de 10/10/2001, p. 2.)*

*Art.78 (Revogado pelo art. 3º da Lei n. 5.211, de 04 de outubro de 2001, publicada no DOE nº 196, de 10/10/2001, p. 2.)*

## **Seção XVII Das Remoções**

Art. 79. Faz-se a remoção mediante escolha pelo Poder Executivo, sempre que possível, em lista tríplice organizada pelo Tribunal de Justiça, com os nomes dos candidatos com mais de dois anos de efetivo exercício na entrância.

Art. 80. Para cada vaga destinada ao preenchimento por remoção, abre-se inscrição distinta sucessivamente, indicando-se a comarca ou vara a ser preenchida.

Art. 81. A vaga para preenchimento por remoção se noticia no Diário da Justiça.

Art. 82. O Tribunal de Justiça pode indeferir o pedido de remoção, mesmo que o candidato preencha os requisitos legais, caso considere tal fato contrário aos interesses da Magistratura.

Art. 83. Os Juízes que tenham sofrido qualquer punição, antes de decorrido um ano, ficam impossibilitados de concorrer à vaga preenchível por remoção.

Art. 84. Em caso de mudança da sede do juizado, elevação ou rebaixamento de entrância, supressão da comarca, facultada ao Juiz remove-se para a nova sede, para comarca de igual entrância ou obter disponibilidade com vencimentos integrais.

Art. 85. A remoção, no caso § 3º, do art. 113, da Constituição Federal, verifica-se quando a permanência do Juiz for prejudicial ao interesse público e houver pronunciamento, em escrutínio secreto, de dois terços, no mínimo, dos membros efetivos do Tribunal.

§ 1º O processo de remoção compulsória inicia-se por proposta do Presidente do Tribunal de Justiça; do Corregedor da Justiça; por representante de um terço pelo menos, dos Desembargadores; por representação do Chefe do Poder Executivo Estadual; do Procurador Geral da Justiça; ou do Conselho Seccional da Ordem dos Advogados do Brasil.

§ 2º O prazo de defesa prévia do magistrado é de quinze dias, contados da data em que receber a cópia do inteiro teor das acusações e das provas, a ele encaminhada por ofício do Presidente do Tribunal de Justiça.

§ 3º Com defesa prévia ou sem ela, decorrido o prazo, o Presidente, no dia útil e imediato, convoca o Tribunal para que, secretamente, decida sobre a instauração do inquérito; em caso afirmativo, distribui-se o feito ao Relator respectivo, podendo-se, na mesma sessão, afastar o magistrado de suas funções, sem prejuízo de vencimento ou vantagens, até decisão final.

§ 4º Defere-se a produção de provas em vinte dias, com a participação do representante do Ministério Público e do procurador do magistrado, se houver, com prazo de vista de dez dias a cada parte, para razões, seguindo-se o julgamento em sessão secreta, depois de relatório oral e da decisão publicando-se somente o que for escolhido.

Art. 86. Entre outros casos, reputa-se prejudicial ao interesse público a permanência na Comarca do Juiz que:

- a) se der ao vício de embriaguês ou de substância tóxicas;
- b) cometer atos atentatórios à moral e aos bons costumes, ainda que não haja representação;
- c) praticar atos de violência ou abuso de poder;
- d) na imprensa falada, escrita ou televisionada, se empenhar em polêmica utilizando-se de linguagem incompatível com a dignidade do cargo, ou por idênticos meios de comunicação criticar, de modo desrespeitoso, decisões do Tribunal de Justiça ou dos seus membros;
- e) estiver, em razão de exercício do cargo, ameaçado em sua segurança pessoal ou de sua família.

Parágrafo único. O Corregedor da Justiça, ao ter conhecimento dos fatos especificados neste artigo, deve apurá-los em sindicância, encaminhando o resultado ao Tribunal.

Art. 87. A remoção por permuta, admissível entre Juízes de Direito da mesma entrância é decidido pelo Tribunal de Justiça, por maioria simples de seus membros efetivos e encaminhada ao Poder Executivo Estadual.

Art. 88. Em qualquer caso de remoção, o Poder Executivo dispõe de quinze dias para lavratura do ato respectivo.

## **Seção XVIII**

### **Dos Serviços Auxiliares da Justiça**

## **CAPÍTULO I DA SECRETARIA DO TRIBUNAL**

Art. 89. Os serviços auxiliares da Justiça são efetuados pela Secretaria do Tribunal, pelos oficiais de Justiça de primeira e segunda entrâncias e pelos serventuários e funcionários da Justiça.

Art. 90. A Secretaria do Tribunal funciona sob a responsabilidade de um Diretor Geral, subordinado diretamente à Presidência, sendo seus serviços executados na forma prevista no Regimento Interno do Tribunal, com um quadro de servidores fixado em lei, mediante proposta deste.

## **CAPÍTULO II DOS TABELIÃES DE NOTAS**

Art. 91. Compete aos Tabeliães de Notas:

- I – lavrar, em livro de notas, testamentos públicos, contratos e procurações;
- II – fazer instrumento de aprovação de testamento cerrado;
- III – lançar o nome do testador no invólucro de testamento cerrado, declarando a data da respectiva aprovação, encerramento e entrega;
- IV – aprovar o testamento, entregá-lo ao testador, anotar no livro respectivo, mencionando o lugar, dia mês e ano em que foi aprovado e entregue;
- V – transcrever, nas escrituras, os documentos e procurações a que se referem aquelas salvo quando estas tiverem sido lavradas em suas notas, assim como o conhecimento ou o certificado do imposto de transmissão nos contratos a ele sujeito;
- VI – emacçar e guarda, para servirem de suplementos ao livro de notas, os conhecimentos, as procurações ou certidões que transcreverem nas escrituras em cumprimento ao disposto no número antecedente;
- VII – lavrar protestos de títulos, de letras de câmbio e notas promissórias, intimar os interessados e extrair os respectivos instrumentos;
- VIII – extrair, independentemente de despacho judicial, traslado de escrituras lavradas em suas notas;
- IX – passar públicas formas de documento avulso;
- X – reconhecer, letra, firma e sinal públicos;
- XI – dar, independentemente de seu despacho judicial, certidão textual ou narrativa do que constar em razão do ofício;
- XII – comunicar ao oficial do Registro de Imóveis a escritura que lavrar, ou a relação de bens particulares da mulher lançados em suas notas;
- XIII – remeter ao Escrivão de órfãos certificados das escrituras de doação lavrar em favor de algum menor, interdito nascituro, especificando:
  - a) o nome e o domicílio do doador;
  - b) o nome, filiação e domicílio do menor ou interdito, e identificação do nascituro;
  - c) o objeto da doação e a data da doação, certificando à margem desta a remessa.
- XIV – assinar, no Tribunal de Justiça, em livro próprio, antes de assumir o ofício, o espécime de sua letra e firma e o sinal público que haja de usar, lavrando-se disto o competente termo, feito e subscrito pelo escrevente e assinado pelo Presidente;
- XV – notificar o donatário para declarar se aceita ou não a doação, quando o doador fixar data para isso;
- XVI – entregar às partes, dentro de cinco dias primeiros trasladados das escrituras que fizer, se numa escritura pública houver dois ou mais outorgados, ou as partes forem

reciprocamente outorgantes e outorgados, cada um daqueles ou cada uma destas tem direito a um primeiro traslado;

XVII – conservar em ordem os livros e papéis do cartório, facultada, em qualquer tempo, a remessa, de tais livros e papéis ao Arquivo Público, desde que tenham antiguidade superior a quinze anos e inferior a trinta;

XVIII – manter seus cartórios abertos das 8 (oito) às 11 (onze) horas e das 13 (treze) às 18 (dezoito) horas dos dias úteis, salvo aos sábados, cujo expediente é facultativo;

XIX – exercer, no desempenho de suas funções, rigorosa fiscalização do pagamento do imposto, taxas e emolumentos devidos por força dos atos jurídicos que lhe sejam apresentados;

XX – residir na sede da comarca, não podendo ausentar-se sem licença do Corregedor, na Capital, do Diretor do Forum onde houver, ou do Juíz nas demais Comarcas;

XXI – manter irrepreensível compostura e dignidade no exercício do cargo, acatar as determinações de seus superiores hierárquico e exercer, com absoluta probidade, o seu ofício;

XXII – cumprir as prescrições legais concernentes às suas atribuições e à fiel observância do Regime de Custas;

XXIII – tratar com urbanidade as partes e atendê-las com solicitude;

XXIV – assegurar a necessária disciplina em seus ofícios, solicitando, da autoridade competente, as providências devidas contra qualquer irregularidade funcional;

XXV – possuir, escriturados, todos os livros exigidos por lei e manter o cartório em prédio seguro, instalado em asseio e ordem, em próprio do município ou do Estado;

XXVI – facilitar as correições;

XXVII – fazer constar, obrigatoriamente, do próprio instrumento, independentemente da expedição do recibo, quando solicitado, o valor das custas de escrituras, certidões, buscas, averbações, registros, emolumentos e despesas legais.

Art. 92. É defeso aos tabeliães:

I – lavrar:

a) sem as formalidades legais, qualquer ato de seu ofício;

b) escritura especial, pacto total ou exclusivo de comunhão, no todo ou em parte, sem que conste os bens respectivos, ou excluídos, e o valor em que são estimados;

c) qualquer instrumento ou contrato, sem a prova de pagamento do imposto ou emolumentos devidos.

II – cancelar, riscar, emendar, rasurar ou por nas entrelinhas qualquer palavra da escritura ou instrumento sem fazer no fim, antes de assinar, a ressalvada devida;

III – dar certidões além do que constar nos livros autos e papéis do cartório;

IV – usar de abreviatura ou escrever em algarismo o dia mês, ano, salvo quando o façam também por extenso.

Art. 93. O tabelião poderá fazer lavrar escritura ou qualquer outro instrumento por compromissado, contando que o subscreva e carregue com a responsabilidade do ato.

Parágrafo único. Excetuam-se os atos lavrados fora do cartório.

### **CAPÍTULO III DOS OFICIAIS DE REGISTRO CIVIL**

Art. 94. Compete aos Oficiais do Registro Civil:

I – servir em todos os atos preparatórios do casamento e da sua celebração, lavrando o respectivo assento em livro próprio;

II – opor à celebração do casamento as circunstâncias impeditivas do art. 183, n° 1 e 11 do Código Civil, de cuja existência tenha certeza;

III – dar aos nubentes ou a seus representantes legais nota do impedimento oposto, indicando os fundamentos, as provas e, se o impedimento não se opuser de ofício, o nome do oponente;

IV – lavrar os assentos e fazer a inscrição do nascimento e óbito;

V – entregar, com certidão do registro de nascimento, a folho do preceito de puericultura, na forma determinada pelo Decreto Federal n° 9.017, de 23.07.1946;

VI – transcrever:

a) no livro destinado ao registro de casamento, o termo avulso lavrado por oficial ad hoc, no caso do art. 198, & 1° do Código Civil, assim como, quando passarem em julgado, as decisões judiciais que homologarem o casamento celebrado em iminente risco de vida;

b) as opções de nacionalidade.

VII – inscrever:

a) no livro destinado ao registro de casamento, logo que transite em julgado, a sentença proferida em processo judicial de que resulte prova de celebração legal de casamento;

b) a emancipação por outorga do pai, da mãe, ou sentença do Juíz, a interdição de loucos, surdos e mudos ou dos pródigos e a sentença declaratória de ausência.

VIII – averbar no registro:

a) as sentenças que decidirem sobre nulidade ou anulação de casamento, separação judicial, divórcio e reconhecimento de filhos adulterinos e restabelecimento da sociedade conjugal;

b) as sentenças que julgarem ilegítimos os filhos concebidos na constância do casamento e as que declararem legítima a filiação;

c) as sentenças de reconhecimento dos filhos naturais depois de separação judicial e divórcio, de acordo com a lei;

d) os casamentos de que resultar legitimação de filhos havidos ou concebidos anteriormente;

e) os atos judiciais ou extrajudiciais de reconhecimento de filho legítimo;

f) as escrituras de adoção e os atos que as revogarem;

g) as alterações ou averbações de nomes.

IX – funcionar nas causas que correrem no juizado de casamento e nos processos de justificações relativos ao registro civil de nascimento, casamento e óbito;

X – anotar, no novo assento, os papéis mencionados nos números anteriores, quando o nascimento ou casamento houver sido registrado em outro cartório;

XI – remeter ao cartório em que tiver sido feito o registro, a fim de que seja averbada a certidão do novo assento lavrado na conformidade do número anterior.

Art. 95. Nos Termos Judiciários incumbem aos oficiais as atribuições constantes do artigo antecedentes sem que possam funcionar nas causas correm nos juzados de casamento.

Parágrafo único. Os processos ao registro civil dos Termos Judiciários, quando dependam de despacho do Juiz togado, a este são remetidos pelo Oficial do Registro da sede do Juizado.

Art. 96. Os Oficiais do Registro Civil das Pessoas Naturais exercem, por distribuição, onde houver mais de um cartório, as funções de escrivania nas causas de separação, divórcio, de nulidade e anulação de casamento.



Art. 97. Os Oficiais do Registro Civil obrigam-se a remeter à Fundação IBGE, dentro dos primeiros oito (8) dias dos meses de janeiro, abril, julho e outubro de cada ano, um mapa dos nascimentos, casamentos e óbitos que houverem registrado no trimestre anterior, observado o disposto na respectiva legislação.

Art. 98. Os Oficiais do Registro Civil são obrigados a satisfazer as exigências da legislação federal sobre alistamento e sorteio militar nas condições estabelecidas na lei.

Art. 99. Se os Oficiais do Registro Civil recusarem ou demorarem a fazer registro, averbação, anotação ou certidão, as partes prejudicadas podem queixar-se à autoridade judiciária competente que decidirá, ouvido o acusado.

Art. 100. No caso de ser injusta a recusa ou injustificável a demora, a autoridade que tomar conhecimento do fato deve impor ao Oficial a multa do valor de um a três vezes o maior valor de referência cobrada na forma da lei, ordenando que, no prazo improrrogável de vinte e quatro horas, seja cumprida a obrigação.

#### **CAPÍTULO IV DOS OFICIAIS DO REGISTRO DE IMÓVEIS**

Art. 101. Compete aos Oficiais do Registro de Imóveis:

a) a inscrição:

1. de instrumento público que instituir bem de família;
2. de instrumento público das convenções ante-nupciais;
3. das hipotecas legais;
4. dos empréstimos por obrigações ao portador;
5. do penhor e máquinas e aparelhos utilizados na indústria, instalados e em funcionamento, com os respectivos pertences;
6. das penhoras, arrestos e sequestros de imóveis;
7. das citações de ações reais, ou processos reipersecutórios relativos a imóveis;
8. do memorial de loteamentos de terrenos urbanos e rurais para venda de lotes e prazo em prestações;
9. do contrato de locação do prédio, no qual tenha sido consignada cláusula de vigência no caso de alienação de coisa locada;
10. dos títulos dos servidões não aparentes para a sua constituição;
11. do usufruto e do uso sobre imóveis e sobre a habitação, quando não resultarem de direito de família;
12. das rendas constituídas de imóveis ou a estes vinculadas, por disposição de última vontade;
13. de contrato de penhor agrícola;
14. da promessa de compra e venda do imóvel não loteado, para a sua validade para as partes contratantes e em relação à terceiros.

b) a transcrição:

1. da sentença da separação judicial, de divórcio e de nulidade ou anulação de casamento, quando nas respectivas partilhas existirem imóveis ou direitos reais, sujeitos a essa formalidade;
2. dos títulos, ou a inscrição de atos intervivos relativamente aos direitos reais sobre imóveis, que para se adquirir domínio, que para a validade contra terceiros;
3. dos títulos translativos da propriedade imóvel, intervivos para a sua aquisição ou extinção;

4. dos julgados, nas ações divisórias, pelas quais se puser termo à indivisão;
  5. das sentenças que, nos inventários e partilhas, adjudicarem de raiz em pagamento das dívidas da herança;
  6. dos atos de entrada de legado de imóveis dos formais de partilha e das sentenças de adjudicação em inventário quando não houver partilha;
  7. da arrecadação do que for adjudicado em hastas públicas;
  8. da sentença declaratória da posse do imóvel, por vinte anos, sem interrupção ou oposição, para servir de título ao adquirente por usucapião;
  9. da sentença declaratória de posse incontestada e contínua de uma servidão aparente por dez e vinte anos, nos termos do art. 551 do Código Civil;
  10. para a perda de propriedade imóvel, dos títulos transmissíveis ou dos atos renunciativos.
- c) a averbação:
1. das convenções antenupciais, especialmente em relação aos imóveis existentes ou posteriormente adquiridos que se atinjam pelas cláusulas exclusiva do regime legal;
  2. da inscrição da sentença que separar o dote;
  3. de sentença de restabelecimento da sociedade conjugal;
  4. da cláusula de inalienabilidade, imposta a imóveis pelos testadores ou doadores;
  5. por cancelamento, da extinção dos direitos reais;
  6. dos contratos de promessa de compra e venda de terreno lotado, em conformidade com as disposições do Decreto nº58, de 10/12/1937;
  7. da circunstância da mudança de número da construção, da reconstrução, da demolição e do desmembramento do imóvel;
  8. da alteração de nome por casamento, separação judicial ou divórcio.

Art. 102. Todos os atos enumerados no artigo anterior são obrigatoriamente efetuados nos cartórios da situação do imóvel.

Parágrafo único. Em se tratando de imóveis situados em comarcas limítrofes, o regime deve ser observado em todas elas.

Art. 103. O registro de imóvel rege-se pelas disposições da lei nº 6.015, de 31.12.73, e legislação federal subsequente, sendo os Oficiais obrigados a possuir os livros exigidos pelos diplomas legais e escriturá-los de acordo com suas prescrições.

## **CAPÍTULO V**

### **DOS OFICIAIS DO REGISTRO DE TÍTULOS E DOCUMENTOS**

Art. 104. Compete aos Oficiais do Registro de Títulos e Documentos, quanto ao Registro Civil das pessoas jurídicas:

- a) inscrever:
  1. os contratos, os atos constitutivos, estatutos ou compromissos das sociedades civis, religiosas, pias, morais, científicas ou literárias, e os das associações de utilidade pública e das fundações;
  2. as sociedades civis que revestirem as formalidades estabelecidas nas leis comerciais;
- b) fazer registro dos jornais e de publicações periódicas, das oficinas impressoras de qualquer natureza, pertencentes a pessoas naturais ou jurídicas, das empresas de radiodifusão que matenham serviços de notícias, reportagens, comentários, debates e entrevistas e das empresas que tenham por objeto o agenciamento de publicidade, conforme o disposto nos arts. 8 a seguintes de Lei nº 5.250, de 09.02.1967;

c) averbar todas as alterações supervenientes, que importem em modificações ou alterações das circunstâncias constantes de registro, atendidas as exigências das leis em vigor.

Art. 105. Compete aos Oficiais de Registro de Títulos e Documentos, na parte referente ao registro de títulos e documentos:

a) a transcrição:

1. dos instrumentos particulares, para a prova das obrigações convencionais de qualquer valor, bem como de cessão de crédito e outros direitos, por eles criados para valer contra terceiros e do pagamento com subrogação;

2. do penhor comum de causa móveis, feito por instrumento particular;

3. da caução de título de crédito pessoal e da dívida pública federal, estadual ou municipal, ou de bolsa ao portador;

4. de contrato, por instrumento particular, de penhor de animais, não compreendido nas disposições do art. 781, n° V, do Código Civil;

5. de contrato, por instrumento particular, de parceria agrícola;

6. do mandado judicial de renovação de contrato de arrendamento para sua vigência que entre as partes constantes, que em face de terceiros;

7. facultativo de quaisquer documentos para sua conservação.

b) averbação de fato de prorrogar contrato particular de penhor de animais.

Art. 106. Sujeitam-se a transcrição de registro de títulos e documentos, para valer contra terceiros:

I – os contratos de locação de prédio, feitos por instrumento particular, não compreendidos nas disposições do art. 1.197 do Código Civil;

II – os documentos decorrentes de depósitos, ou de cauções feitos em garantia de se cumprirem obrigações contratuais ainda que em separado dos respectivos instrumentos;

III – as cartas de fianças em geral, feitas por instrumento particular, seja qual for a natureza do compromisso por ela abonado;

IV – os contratos de locação de serviços não atribuídos a outras repartições;

V – os contratos de compra e venda em prestações, a prazo com reserva de domínio ou não, qualquer que sejam a forma de que se revistam;

VI – os documentos de procedência estrangeira, acompanhados das respectivas traduções, quando tenham de produzir efeitos em órgãos da União, dos Estados e dos Municípios, ou em qualquer instância, Juízo ou Tribunal;

VII – os contratos de compra e venda de automóveis, bem como o de penhor deles, qualquer que seja a forma que assumam.

Art. 107. O Registro de Títulos e documentos rege-se pelas disposições da Lei n° 6.015, modificada pela legislação subsequente, sendo os Oficiais obrigados a possuir os livros exigidos e escriturá-los de acordo com as prescrições do referido diploma legal.

## **CAPÍTULO VI DOS ESCRIVÃES EM GERAL**

### **Seção XIX Preliminares**

Art. 108. Compete aos Escrivães em Geral:

I – escrever em forma, os mandados, termos, atos e demais peças judiciárias consignando o dia, mês e ano em que se fizerem;

- II – passar procuração apud-acta;
- III – comparecer às audiências ou providencias para que a elas compareça um escrevente;
- IV – lavrar, no protocolo, termos do que ocorrer nas audiências relativamente aos feitos em que servirem;
- V – efetuar citações, notificações e intimações dos despachos ou sentenças e de atos de andamento do processo;
- VI – ter sob sua guarda e responsabilidade todos os autos e papaeis que lhe tocarem por distribuição ou que, por vidade de seu ofício, lhes forem entregue pelas partes;
- VII – encaminhas aos Juízes as petições que as partes lhe dirigirem, com anotações da hora em que foram apresentadas;
- VIII – praticar, de sua conta, os atos e diligências que por erro ou negligência sua, devam ser renovados sem embargo da penas em que por isso, tenham incorrido;
- IX – prestar às partes interessadas ou a seus procuradores, quando o solicitarem, informações escritas ou verbais sobre o estado e andamento dos feitos e facilitar-lhe em qualquer tempo, a consulta dos processos em cartório;
- X – certificar, antes do termo, de visita à parte contrária, se os documentos Junto aos autos estão cancelados, riscados, emendados, raspados, borrados ou de qualquer modo, viciados em seu contexto;
- XI – conferir e consertar traslados de autos;
- XII – receber custas consignadas no Regimento e entregá-las a quem competir;
- XIII – dar às partes, ainda que não exijam, recibos das custas que receberam;
- XIV – remeter, sob protocolo, aos Juízes, membros do Ministério Público, curadores, advogados, peritos ou partidores do Juízo, quando for o caso, os autos conclusos ou com vistas;
- XV – fazer os autos conclusos ao Juiz, logo que estiverem em termo de despacho ou sentença;
- XVI – cumprir as determinações legais dos Juízes com os quais sirvam;
- XVII – numerar todas as folhas do processo e rubricar as em que não houver a sua assinatura ou a do Juiz;
- XVIII – executar os atos judiciais, salvo disposição em contrário dentro de quarenta e oito horas, contando-se o prazo:
  - a) para os atos que se devam praticar, em virtude da lei, da data em que se houver concluído o ato processual anterior;
  - b) para os atos ordenados pelo Juiz, da data em que o serventuário tiver ciência da ordem.
- XIX – cumprir o disposto no art. 141 do Código de Processo Civil;
- XX – recusar, em cartório, quando for o caso, a petição inicial, a defesa, os quesitos, laudos ou quaisquer requerimentos não constantes de registro público, quando não vierem acompanhados por cópia, datados e assinados por quem os oferecer ou por procuradores legítimos;
- XXI – conferir cópia e formar, com elas e com as autenticadas dos depoimentos, termos de audiências, despachos, sentenças e acórdãos, os autos suplementares dos feitos em curso.

Art. 109. É defeso aos escrivães em Geral.

- I – permitir a retirada do Cartório dos autos originais sob pena de responsabilidade, salvo:
  - a) quando tenham de subir à conclusão do Juíz;
  - b) quando devam ser remetidas ao Contador ou Partidor do Juizado;

c) em caso de vista ao órgão do Ministério Público e aos advogados;  
 d) nos casos em que, por modificação da competência, tenham de ser remetidos a outro Juízo.

II – dar certidões além do que constar dos livros, autos ou papéis do cartório;

III – aceitar depósito nos feitos em que servirem;

IV – escrever em autos que lhes não seja distribuídos salvo nos casos de substituição;

V – tratar incivemente as partes;

VI – cancelar, riscar, emendar, por nas entrelinhas qualquer palavra escrita, sem fazer, no fim antes de assinatura, a devida ressalva;

VII – usar de abreviaturas e escrever em algarismo do dia, mês e ano, salvo quando façam também por extenso.

Art. 110. O escrivão tem fé pública nos atos pertencentes a seu ofício, mas esta pode ser ilidida por prova em contrário.

Art. 111. Os erros e omissões do escrivão não prejudicam as partes que tenham cumprido as disposições legais.

Art. 112. O escrivão deve ter um livro-tombo no cartório com a indicação dos nomes das partes, pela ordem alfabética da natureza dos feitos e ordem cronológica das datas de entrada e distribuição, e os livros destinados a protocolo de audiências, carga e descarga dos autos e os demais determinados pela Corregedoria da Justiça, na capital, pelos Juízes, no interior, ou Diretor do Forum, onde houver.

Art. 113. O escrivão que for chamado a servir junto às autoridades policiais do interior faz jus às custas regimentais pelos atos que praticarem, pagas pelo Estado.

## **Seção XX** **Dos Escrivães do Crime**

Art. 114. Compete aos Escrivães do Crime:

I – servir na formação da culpa, desde a queixa ou denúncia até a pronúncia nos processos cujo julgamento seja da competência do Tribunal do Juri;

II – lavar, autuações, citações, notificações e demais atos processuais;

III – funcionar até o final, nos processos por crime comum da competência do Juíz singular, e no recurso de pronúncia ou impronúncia, e intimar as partes dos despachos respectivos;

IV – remeter, com máxima brevidade, ao Escrivão do Tribunal do Júri, os processos da competência deste, se for decretada a pronúncia;

V – escrever nas justificações criminais, sem prejuízo de igual competência de Escrivão do Júri, e na restauração de autos criminais ainda não remetidos ao Escrivão do Júri.

## **Seção XXI** **Dos Escrivães do Juri e Execuções Criminais**

Art. 115. Compete aos Escrivães do Júri e Execuções Criminais:

I – secretariar as sessões do Tribunal do Júri, praticando os atos que lhe são atribuídos pelo Código de Processo Penal;

II – servir:

- a) na formação de culpa e na fase de julgamento dos crimes funcionais da competência do Juíz singular;
  - b) nos processos de habeas-corpus;
  - c) nos processos de fiança e quaisquer incidência posteriores à pronúncia e à sentença condenatória;
  - d) no sorteio e revisão dos jurados;
  - e) nas execuções das sentenças criminais.
- III – funcionar, após a pronúncia, em todos os termos do processo, tendo seguimento por seu cartório os recursos cabíveis;
- IV – intimar as partes dos despachos de pronúncia ou de sua revogação;
- V – lançar os nomes dos réus pronunciados ou absolvidos;
- VI – escrever nas justificações criminais e na restauração de autos perdidos, sem prejuízo de igual competência do Escrivão do Crime;
- VII – passar folha corrida e certidões.

### **Seção XXII Dos Escrivães do Cível**

Art. 116. Compete aos Escrivães do Cível:

- I – servir em todas as ações, execuções e negócios cíveis que não pertençam privativamente a outro escrivão;
- II – escrever nas justificações que tenham de ser apresentadas ao Juíz Cível, guadadas as atribuições dos demais ofícios;
- III – tomar protestos para ressalva de direito;
- IV – praticar os demais atos do ofício.

### **Seção XXIII Dos Escrivães da Provedoria**

Art. 117. Compete aos Escrivães da Provedoria:

- I – registrar o testamento e codicilos, remetê-los à inscrição e arquivá-los;
- II – escrever todos os feitos, que correm pelo Juízo da Provedoria;
- III – remeter ao Escrivão, quando registrar testamento instituindo legado ou herança, em favor de algum menor ou interdito, certificado, especificando:
  - a) o nome e domicílio do testador;
  - b) o objeto da herança ou legado;
  - c) o nome e domicílio do menor ou interdito.
- IV – certificar, à margem do registro, a remessa efetuada em cumprimento ao disposto no número anterior;
- V – comunicar às fundações ou associações pias ou beneficentes os legados intituídos em seu favor, nos testamentos e codicilio que registrar.

### **Seção XXIV Dos Escrivães de Órfãos e Ausentes**

Art. 118. Compete ao Escrivães de Órfãos e Ausentes:

- I – servir em todas as causas que correrem em Juízo orfanológico;
- II – remeter, de ofício, ao Oficial do Registro de Imóveis cópia do termo de tutela ou curatela, que se assinar no Juízo de Órfãos;

III – notificar o tutor ou curador de menor ou de interdito, logo que for assinado o termo de tutela ou curatela, para que se faça a inscrição de hipoteca legal;

IV – certificar à margem do compromisso ou do termo, a notificação efetuada em obediência ao disposto no número anterior.

Art. 119. Ao receber os certificados das escrituras de doação e de algum testamento instituindo legado em favor de menor ou interdito, o Escrivão procede da seguinte forma:

§ 1º Se o menor não estiver sob pátrio poder ou tutela, apresenta o certificado ao Juiz de Órfãos para que se nomeie tutor ou curador, finalmente cumprindo determinação do número III do artigo anterior;

§ 2º Se o menor estiver sob pátrio poder ou tutela e houver inventário, faz juntada aos autos de certificado, para que o Juiz providencie sobre a arrecadação dos bens da herança doados ou legados. Caso não haja inventário, autuado o certificado, apresenta-o ao Juiz, para que este ordene o que for de direito, e efetue a notificação do número III do artigo anterior.

Art. 120. Compete também aos Escrivães de Órfãos e Ausentes, levar ao conhecimento do Juiz:

- a) a existência de Órfãos que não tenham tutor;
- b) os que devam dar bens de Órfãos a inventários;
- c) a existência de bens de Órfãos, ausentes e interditos que devam ser arrecadados;
- d) a falta de prestação de contas de tutores ou curadores;
- e) a necessidade de que se notifiquem os responsáveis pela administração dos bens de órfãos, menores, interditos, para que procedam à especialização e inscrição das hipotecas legais.

### **Seção XXV**

#### **Dos Escrivães dos Feitos da Fazenda**

Art. 121. Compete aos Escrivães dos Feitos da Fazenda:

- I – escrever em todos os processos que correrem no Juízo da sua privatividade;
- II – organizar, na especialização das hipotecas legais da Fazenda, os extratos respectivos, para que se escrevam no registro de imóveis;
- III – tomar protestos referentes a objetos que envolvam interesses da Fazenda;
- IV – exercer qualquer atribuição de seu Ofício privativo.

### **Seção XXVI**

#### **Dos Escreventes**

Art. 122. Os Escreventes dividem-se em duas categorias: compromissados e auxiliares.

Art. 123. Aos Escreventes, em geral, incumbe:

- I – comparecer ao serviço nos dias úteis, nele permanecendo durando o expediente do cartório;
- II – executar os encargos que lhe forem determinados pelos serventuários a que estiverem subordinados.

Art. 124. O Escrevente compromissado pode praticar todos os atos de serventuário, salvo os que a lei expressamente determinar devem ser feito por este, e escrever todos os termos e atos, que tiverem de ser subscritos pelos serventuários, quando o exija a fé pública.

Art. 125. Nos Juízos ou ofícios em que haja mais de um escrevente, designa-se um dos compromissados para as funções de substituto, indicado pelo respectivo titular.

Art. 126. Ao Escrevente substituto incumbe substituir o serventuário, quando for o caso.

Art. 127. O Escrevente substituto deve arquivar a sua firma e sinal público no Tribunal de Justiça, por intermédio do Tabelião titular.

Art. 128. O Escrevente substituto dos Oficiais do Registro Civil, com mais de cinco anos de exercício efetivo, pode, autorizado pelo Corregedor, pelo Diretor do Fórum, onde houver, e pelos Juízes nas demais comarcas, e sob a responsabilidade do Escrivão, praticar todos os atos do registro civil.

Art. 129. Ao Escrevente Auxiliar incumbe:

I – nos cartório dos Juízes executar os serviços de expediente e de entrega dos processos, além dos que lhe forem determinados pelos Escrivães;

II – no ofício de notas e de registros, exercer as funções de protocolista, arquivista, rasista e verificador de firma.

## **CAPÍTULO VIII DOS DISTRIBUIDORES, CONTADORES E OUTROS SERVENTUÁRIOS**

### **Seção XXVII Dos Distribuidores**

Art. 130. O Distribuidor efetua, com rigorosa igualdade, entre os Juízes e Escrivães, quando for o caso, a distribuição alternada dos feitos assim classificados:

- a) ações cíveis;
- b) ações penais;
- c) processos precatórios, preventivos ou assecuratórios de direito ou ação;
- d) processos de falência;
- e) os feitos diversos não compreendidos nas cláusulas referidas.

Art. 131. Distribuem-se por dependência, os feitos de qualquer natureza, que se relacionarem com outros já distribuídos.

Art. 132. São isentos de distribuição os feitos de competência privativa ou exclusiva.

Art. 133. Nos casos de competência mediante prorrogação de continência ou conexão, distribui-se por despacho do Juíz, por meio de averbação, que se lança em coluna especial dos livros de distribuição, com a necessária referência ao feito principal, consignada no pedido do dependente.

Art. 134. A distribuição e o registro se efetuam em livros correspondentes a cada classe de processo, em ordem sucessiva, de acordo com a natureza da ação ou o título especial do feito.

Art. 135. Cada feito é registrado pelo Distribuidor, com as seguintes indicações:



- a) número de ordem;
- b) ano, mês, dia e hora em que se apresentar a petição inicial;
- c) o nome do autor e do réu do requerente interessado;
- d) o nome do advogado ou do procurador, quando a petição não for assinada pela própria parte;
- e) objeto do pedido;
- f) designação do Juiz e do cartório a que couber, na distribuição.

Art. 136. Na petição inicial do Distribuidor anota o Juíz e o cartório a que couber o feito, com a data e hora da apresentação e o número correspondente no livro de distribuição e no tombamento geral.

Art. 137. Não se distribui o feito que não for instruído com o respectivo instrumento de mandato, salvo:

- a) nas ações em causa própria;
- b) quando o autor gozar de benefício de gratuidade;
- c) quando o feito for promovido pelo Ministério Público;
- d) na hipótese do art. 37, do Código do Processo Civil.

Parágrafo único. Não se distribui o feito, sob pena de responsabilidade se a petição inicial não vier acompanhada da taxa judiciária devida na forma da lei, salvo se o autor gozar de benefício de gratuidade ou de isenção.

Art. 138. A petição assinada pela própria parte, nos casos em que a lei o permite, só deve ser distribuída depois que se reconhece a firma do signatário.

Parágrafo único. Se a petição for assinada a rogo, as firmas do signatário e das testemunhas abonadoras são devidamente reconhecidas.

Art. 139. A distribuição, uma vez feita, só pode ser declarada sem efeito por despacho do Juiz competente:

- a) quando o feito tiver que ser remetido a outro Juiz em vidade de continência ou conexão;
- b) quando o Juiz se julgar incompetente.

Art. 140. A falha do erro da distribuição se compensa de ofício ou a requerimento do prejudicado, não anulando o feito mas sujeito o Distribuidor às penas em que incorrer por dolo o negligência.

Art. 141. as partes ou seus procuradores podem fiscalisar a distribuição.

Art. 142. Não se sujeitam a distribuição as execuções de sentenças, nem a reformas de outros pedidos, escrevendo nelas os mesmos Escrivães que sirvam nas ações e nos autos originais.

### **Seção XXVIII Dos Contadores**

Art. 143. Aos contador compete:

I – contar as custas, emolumentos e percentagens em todas as instâncias e Juízos, inclusive os privativos; e o principal e juros das dívidas exequendas, bem como as multas nos processos criminais;

- II – glosar as custas excessivas ou indevidas;
- III – fazer o Cálculo para pagamento do imposto devido à Fazenda Pública;
- IV – escrever, datar, assinar e cartificar os autos do seu ofício.

### **Seção XXIX** **Dos Depositários Públicos**

Art. 144. Compete aos Depositários Públicos:

- I – guardar e conservar todos os bens que lhes forem entregue por ordem do Juiz;
- II – arrecadar os frutos e rendimentos dos bens depositados;
- III – representar ao Juíz sobre a conveniência ou necessidade de reparação e serviços indispensáveis, para evitar a ruína ou para que não fiquem improdutivos e sobre a necessidade ou conveniência da venda, em hasta pública, dos bens de fácil deterioração ou guarda dispendiosa;
- IV – promover, mesmo em Juízo, todos os atos e providências indispensáveis ao resguardo da posse dos bens depositados, ou à conservação dos direitos das partes especialmente para evitar prescrição;
- V – solicitar do Juíz providências sobre o destino do dinheiro que houver recebido e dos bens depositados;
- VI – mostrar os bens a qualquer interessado que os procure ver, ou exhibi-los quando e onde o Juiz determinar;
- VII – prestar contas em Juízo dos bens depositados e seus rendimentos;
- VIII – entregar, mediante mandado do Juíz os bens depositados dentro de vinte e quatro horas, sob as penas da Lei;
- IX – escriturar, em livro próprio, aberto e rubricado pelo Juiz, os depósitos recebidos com especificações de objetos, valores, nomes dos interessados, natureza da documentação de recolhimento, quando se tratar de dinheiro;
- X – exercer as demais atribuições conferidas em Lei.

Art. 145. O depósito em dinheiro, título ao portador ou pedras preciosas ou jóias, se efetua em Banco do Estado, ou onde houver a agência respectiva, noutro estabelecimento oficial, e ainda, na falta delas, na Exatoria Estadual.

Art. 146. É defeso aos depositários empregar em serviços próprios os objetos depositados, ou emprestá-lo sob pena de responsabilidade criminal.

### **Seção XXX** **Dos Avaliadores**

Art. 147. Aos avaliadores compete avaliar os móveis e imóveis, rendimentos, direitos e ações, descrevendo cada causa com a devida individuação e fixando-lhes, separadamente o respectivo valor.

Art. 148. Na determinação do valor dos bens em espécie os Avaliadores observam as seguintes regras:

- I – os móveis se avaliam com atenção à sua matéria, mão-de-obra e utilidade;
- II – os semoventes, observadas as dificuldades e riscos da criação, idade, raça e préstimos;
- III – os imóveis urbanos, examinadas em circunstâncias em que estiverem situados de construção, valor locativo, destino e encargos que os onerem;

IV – os imóveis rústicos relativamente aos rendimentos e benfeitorias, deduzidas as despesas de culturas;

V – os direitos e ações vistas a dificuldade de os tornar efetivos;

VI – as ações de bancos e companhias e quaisquer títulos nominativos ou ao portador particulares ou públicos, segundo a cotação corrente do dia da avaliação e, na sua falta, pelo valor presumível do mercado;

VII – o domínio direto e avaliado em vinte prestações de foros anuais;

VIII – os rendimentos avaliam-se segundo o contrato ou, na falta deste, pelo que possam produzir, deduzidos os encargos;

IX – as peças de ouro, prata, pedra preciosas, jóias e qualquer objeto de metal pelo o que resultar do quilate e do custo do feitiço;

X – o valor de qualquer prestação, consistente em cereais ou outros gêneros, se determina pela cotação comercial ou social;

XI – os feitos pendentes são avaliados separadamente.

Art. 149. Não se repete a avaliação, salvo:

a) Provando-se que a primeira foi irregularmente feita, excessiva ou lesiva, por ignorância ou dolo dos avaliadores;

b) Se entre a avaliação e arrematação se descobrirem falhas ou defeitos na causa avaliada, que lhe diminuam o valor.

Art. 150. Quando tiver de ser feita nova avaliação, nada percebem os avaliadores, podendo ainda ser compelidos a fazê-la sob pena de desobediência e perda dos emolumentos da avaliação reformada.

Art. 151. Toda vez que houver despacho do Juíz mandando proceder à avaliação, as autos respectivos devem encaminhar-se com vistas aos avaliadores, mediante carga, e estes lavrem, neles, no prazo estabelecido em lei, ou fixado pelo Juiz, o seu laudo, sendo-lhes encaminhados pelo Escrivão os quesitos apresentados, se for o caso.

Art. 152. Aos avaliadores se concebe transporte, quando a avaliação se fizer fora do perímetro da cidade.

### **Seção XXXI Dos Partidores**

Art. 153. Aos partidores compete fazer partilha dos inventários, de acordo com o despacho de deliberação de partilha.

### **Seção XXXII Dos Intérpretes e Tradutores**

Art. 154. Os Intérpretes e Tradutores são nomeados, para cada causa, pelo respectivo Juiz, observada a legislação federal em vigor.

Art. 155. Os Intérpretes e Tradutores têm as atribuições e vantagens que a lei lhes conferir.

### **Seção XXXIII Dos Porteiros dos Auditórios**

Art. 156. Aos Porteiros dos Auditórios compete:

- I – acompanhar o Juíz nas diligências;
- II – certificar a afixação de editais;
- III – apregoar a abertura e encerramento das audiências e outros atos em que o pregão for necessário;
- IV – fazer a chamada das partes e testemunhas.

Art. 157. Incumbe, ainda, aos Porteiros dos Auditórios, a guarda, conservação e asseio da causa das audiências e dos móveis nela existentes, que recebem, por inventário escriturados com as rubricas das entradas e saídas.

Parágrafo único. A critério do Juíz, pode ser acumulada a função do Porteiro dos Auditórios com a de Oficial de Justiça, com percepção cumulativa de emolumentos.

#### **Seção XXXIV Dos Oficiais de Justiça**

Art. 158. Aos Oficiais de Justiça compete:

- I – fazer citações, prisões, penhores arrestos e mais diligências ordenadas pelo Juiz;
- II – certificar as citações, intimações e notificações que fizerem no auto das diligências efetuadas;
- III – notificar, sob pena de desobediência pessoas que o auxiliam nas diligências, para a prisão ou para a testemunhar atos de seu ofício;
- IV – executar as ordens de *habeas corpus*;
- V – guardar segredo de justiça;
- VI – nomear depositário nos casos especiais;
- VII – exercer as demais atribuições que lhe forem cometidas em lei, regulamentos ou regimentos.

#### **Seção XXXV Dos Comissários e Vigilantes de Menores**

Art. 159. O Comissário de Menores funciona junto ao Juiz de Menores, em plantão permanente, com escala alternativa do seu pessoal.

Art. 160. Incumbe aos Comissários de Menores, sem prejuízo das atribuições dos órgãos de serviços:

- a) efetuar as investigações relativas aos menores, seus pais, tutores ou encarregados de sua guarda, com o fim de esclarecer a ação do Juiz;
- b) exercer vigilância sobre os menores em geral, fiscalizando a execução das leis de assistência e proteção que digam respeito;
- c) apreender menores abandonados ou que hajam praticado atos anti-sociais, apresentando-os ao Juiz de Menores e procedendo a seu respeito às investigações referidas na letra b;
- d) realizar o serviço de fiscalização e vigilância de menores sujeitos a medidas de segurança, ou entregues mediante termos de responsabilidade e guarda ou, ainda, dados à soldada;
- e) auxiliar, pelos órgãos de serviço, o preparo de processos relativos a menores, sugerindo ou promovendo as medidas preliminares de instrução e levando-as ao conhecimento do Juiz respectivo;

- f) vigiar e fiscalizar as ruas, praças, logradouros públicos em geral, cinemas, bares, cabarés, casas de jogo e diversões públicas na parte que interessa à proteção dos menores;
- g) proceder às investigações concernentes ao meio em que vivem os menores, aos lugares que frequentam e às pessoas que os cercam;
- h) visitar as pessoas das famílias dos menores, para investigações dos antecedentes destes, pessoais ou hereditários;
- i) executar, fielmente, as determinações e instruções do Juiz e Curador de menores.

Art. 161. Aos Vigilantes de Menores incumbe:

- a) além das atribuições cometidas aos Comissários de Menores, auxiliá-los em tudo o que estiver ao seu alcance, cumprindo e fazendo cumprir as suas ordens e instruções;
- b) executar fielmente as determinações e instruções de Juiz e do Curador de Menores.

### **Seção XXXVI** **Dos Outros Serventuários não especializados**

- No texto publicado, por equívoco, parte do Art. 161 foi repetido nesta Seção XXXVI.

Art. 162. Os demais Serventuários e funcionários da Justiça não referidos nas seções anteriores exercem as atribuições decorrentes de suas funções específicas, e na forma das leis, processuais e regulamentos em vigor acatando as ordens e instruções das autoridades superiores.

Art. 163. A Diretoria da Secretaria do Tribunal terá atribuições regulares do Regimento Interno.

### **Seção XXXVII** **Das Substituições**

Art. 164. O Presidente do Tribunal de Justiça é substituído pelo Vice-Presidente e este, observada a ordem de antiguidade dos membros desimpedidos do Colegiado.

Art. 165. O Presidente e o Vice-Presidente, ao deixarem os respectivos cargos, tomam assento nas Camaras de que fazia parte os seus sucessores.

Art. 166. Nos impedimentos e faltas ocasionais, os Desembargadores de uma Câmara são substituídos por Desembargadores de outra Câmara, mediante sorteio público.

§ 1º Nos demais casos, ou quando se esgotarem as substituições previstas neste artigo, os Desembargadores serão substituídos por Juizes de Direito, em exercício, mediante sorteio público.

§ 2º Cessada a causa que houver dado lugar à convocação de Desembargador este devolve ao substituto os autos que lhe tiverem sido distribuídos, cabendo-lhe, todavia, tomar parte no julgamento dos processos de que tenha feito revisão.

Art. 167. O Desembargador em férias ou afastado por solicitação superior Eleitoral, para serventia na Justiça respectiva, pode comparecer ao Tribunal de Justiça para as eleições de Presidente, Vice-Presidente, Corregedor da Justiça, escolha de Juiz para promoção, ou ainda para discutir e votar qualquer assunto de natureza administrativa e regimental.

Art. 168. O Corregedor Geral da Justiça é substituído nas licenças, férias, faltas e impedimentos pelo Vice-Corregedor Geral da Justiça.

- Artigo com redação dada pelo art. 2º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.

Art. 169. Na comarca de Parnaíba, os Juízes se substituem com observância de ordem crescente de numeração das varas, cabendo ao da primeira substituir o da quarta.

Art. 170. Os Juízes de Direito da comarca da Capital serão substituídos:

- a) da 1ª. Vara Criminal substitui o da 2ª., o da 2ª. o da 3ª, e nesta sequência até final, sendo que o último substitui o da 1ª.;
- b) o da 1ª. Vara Cível substituirá o da 2ª., o da 2ª. o da 3ª. e nesta sequência até final, sendo que o último substitui o da 1ª.;
- c) o Juiz da Vara da Fazenda substitui o da Vara de Menores, o da Família e este o da Fazenda;
- d) depois de esgotada esta série de substituições, os Juízes cíveis substituirão os criminais e vice-versa, guardada a ordem de numeração;
- e) os das varas especializadas pelos Juízes cíveis obedecida a mesma ordem.

Art. 171. Nas comarcas de Floriano, Picos e Campo Maior, os Juízes se substituem reciprocamente.

Parágrafo único. Afastados ou impedidos os dois assume o Juiz de Direito Adjunto que o Tribunal designar.

Art. 172. Nenhum Juiz de Direito ou Juiz de Direito Adjunto pode ter exercício, simultaneamente em mais de duas (2) varas ou comarcas.

Art. 173. Os demais Juízes de Direito são substituídos pelo Juiz de Direito Adjunto da Zona e, no impedimento ou impossibilidade de serventia deste por Juiz que o Tribunal designar.

Art. 174. Os Juízes de Paz são substituídos pelos seus respectivos suplentes na ordem numérica.

Art. 175. Os Tabeliães, Escrivães, Oficiais de Registro Civil podem ser substituídos por um dos seus Escreventes Juramentados que ao Juiz compete, no interior, designar. Na Comarca da Capital essa designação cabe ao Corregedor da Justiça.

Parágrafo único. Onde houver dois Escrivães e não existirem Escreventes Juramentados no Cartório, aqueles se substituem reciprocamente, se não for nomeado substituto provisório.

Art. 176. Os Oficiais de Justiça se substituem reciprocamente, podendo, se necessário se nomeado pelo Juiz de Direito, Oficial de Justiça, ad hoc para determinados feitos.

Parágrafo único. Onde houver mais de dois, compete ao Juiz de Direito designar o substituto.

Art. 177. Os Avaliadores e Depositários Públicos, os Contadores, Partidores e Distribuidores são substituídos por pessoas idôneas, devidamente comprometido pelo Juiz, e o Porteiro dos Auditores pelos Oficiais de Justiça, na ordem de antiguidade.

Parágrafo único. Os demais serventuários e funcionários da Justiça são substituídos por quem o Juíz designar, ou por quem a lei ou os regulamentos concedem autoridade para a sua designação.

### **Seção XXXVIII** **Dos vencimentos e Vantagens**

Art. 178. Os vencimentos dos Desembargadores do Tribunal de Justiça não podem ser inferiores ao que percebem os Secretários de Estado, não podendo ultrapassar, porém, os fixados para os Ministros do Supremo Tribunal Federal.

Parágrafo único. Os vencimentos dos Magistrados serão pagos na mesma data fixado para o pagamento dos vencimentos dos Secretários de Estado ou dos subsídios dos membros do Poder Legislativo, considerando-se que desatende às garantias do Poder Judiciário atraso que ultrapasse o décimo dia útil do mês seguinte ao vencido.

Art. 179. O vencimento básico dos Juízes de Direito é fixado com a diferença de 5% (cinco por cento) de uma entrância para outra, feita a computação da mais alta para menos elevada.

- [Artigo com redação dada pelo art. 1º da Lei n. 5.360, de 18/12/2003, publicada no DOE nº 244, de 22/12/2003, p. 2.](#)

Art. 180. Fica atribuída aos Desembargadores em atividade uma representação mensal de trinta por cento sobre o vencimento básico.

Art. 181. Aos Magistrados se atribuem gratificações adicionais por tempo de serviço, não excedente a trinta e cinco (35%) dos vencimentos, computando-se a partir dos cinco (5) anos de serviço público, cinco por cento (5%) por quinquênio.

Art. 182. Os Magistrados podem ainda gozar as seguintes vantagens:

- a) ajuda de custo, para despesa de transporte e mudança;
- b) ajuda de custo, para moradia, nas comarcas em que não houver residência oficial para Juíz, exceto na Capital;
- c) salário família,
- d) diárias;
- e) representação;
- f) gratificação por exercício do magistério em cursos de aperfeiçoamento de magistrados;
- g) gratificação especial por exercício em comarca considerada de difícil provimento e acesso, na forma desta lei.

§ 1º A verba de representação, salvo em exercício de cargo em função temporária, integra os vencimentos para todos os efeitos legais.

§ 2º É proibida qualquer outra vantagem não prevista nesta lei.

Art. 183. As custas contadas por ato das autoridades judiciárias são pagas pelas partes e revestem em favor do Estado, conforme lei específica.

Art. 184. Os Juízes de Direito que substituam outro Juiz, por falta, licença ou férias, recebem uma gratificação correspondente ao período da substituição na base de dez por cento dos próprios vencimentos.

- Artigo com redação dada pelo art. 1º da Lei n. 4.481, de 1º/06/1992, publicada no DOE nº 102, de 1º/06/1992, p. 10.

Art. 185. Os Juízes de Direito Adjunto, quando em exercício do cargo de Juiz de Direito, fora da sede de sua Zona, auferem além dos próprios vencimentos e durante a substituição, uma quantia remuneratória proporcional à metade dos vencimentos do substituído.

Parágrafo único. A despesa prevista neste artigo é paga pelo Órgão Fazendário da sede da zona, mediante requerimento do interessado.

Art. 186. Os Juízes promovidos ou removidos continuam a receber os vencimentos do Juizado anterior até que assumam o novo, sem qualquer vantagem, salvo as do cargo anterior.

Art. 187. A título de representação, ficam atribuídas as vantagens, sobre os respectivos vencimentos, de quinze por cento, ao Desembargador que estiver no exercício do mandato de Presidente; dez por cento aos que estiverem exercendo os mandatos de Vice-Presidente e Corregedor Geral da Justiça; e de cinco por cento ao que estiver exercendo o mandato de Vice-Corregedor da Justiça.

- Artigo com redação dada pelo art. 3º da Lei n. 5.243, de 12/06/2002, publicada no DOE nº 123, de 28/06/2002, p. 3.

Art. 188. Além dos casos previstos na legislação comum, as autoridades judiciárias não podem sofrer qualquer desconto em seus vencimentos, quando chamadas pelo Presidente do Tribunal, pelo Corregedor da Justiça e pelo Conselho de Magistratura, para o desempenho de comissão especial.

Art. 189. O Estado construirá mais sedes das Comarcas prédios com que for nomeado Desembargador, uma ajuda de custo de um mês de vencimento, a título de primeiro estabelecimento.

- No texto publicado, por equívoco, este artigo e o seguinte aparecem como art. 190.

§ 1º Ao bacharel que for nomeado Juiz de Direito Adjunto e deferida uma ajuda de custo correspondente a um mês de vencimento.

§ 2º Quando promovido à entrância imediata, ou o cargo de Desembargador, o Juiz de Direito faz jus a uma ajuda de custo correspondente a um mês de vencimento do novo cargo.

Art. 190. O Estado construirá nas sedes das Comarcas prédios condignos para o foro e residência do Juiz e do Promotor.

Parágrafo único. O Município interessado na instalação funcionamento de comarca recém-criada pode firmar convênio com o Estado para o cumprimento do presente artigo.

Art. 191. *(Revogado pelo art. 7º da Lei Complementar n. 54, de 26/10/2005, publicada no DOE nº 203, de 27/10/2005, p 2.)*

Art. 192. O Magistrado que houver exercido comissão permanente, ou percebido gratificação de representação, a qualquer título, durante quatro anos, consecutivos ou não,



tem, ao aposentar-se, direito a perceber, como vantagem pessoal, o valor correspondente a esta última, fixado na época da aposentadoria, esteja, ou não gozando vantagem.

Parágrafo único. O benefício deste artigo estende-se ao magistrado que tenha exercido, por qualquer tempo, a Presidência do Tribunal de Justiça.

Art. 193. As filhas de magistrados, viúvas, sem rendimentos, passam a perceber a pensão correspondente à que perceberiam como se inuptas fossem.

- *Artigo com redação dada pela Lei n. 3.786, de 02 de abril de 1981, publicada no DOE nº 68, de 10/04/1981, p. 3.*

Art. 194. Ao magistrado que se deslocar, temporariamente, da sede de seu Juízo ou Comarca, em objeto de serviço público ou em estudos especializados, concedem-se ajuda de custo e diárias, a título de indenização para despesas de alimentação e pousada.

### **Seção XXXIX Das Licenças e Férias**

Art. 195. São competentes para conceder licenças:

- a) o Tribunal de Justiça a seus membros, aos Juízes e serventuários que lhe forem imediatamente subordinados (C.F.art.115, item IV);
- b) o Corregedor ao pessoal da Secretaria da Corregedoria da Justiça e ao do Foro da Capital;
- c) os Juízes de Direito aos serventuários e funcionários da Justiça de sua Comarca e, onde houver, o Diretor do Forum, no interior do Estado.

Art. 196. *(Revogado pelo art. 10 da Lei Complementar n. 88, de 05/09/2007, publicada no DOE nº 170, de 06/09/2007, pp. 1/2.)*

Art. 197. *(Revogado pelo art. 10 da Lei Complementar n. 88, de 05/09/2007, publicada no DOE nº 170, de 06/09/2007, pp. 1/2.)*

Art. 198. Os Magistrados têm direito a sessenta (60) dias de férias anualmente, coletivas ou individuais.

§ 1º Os Desembargadores gozam férias coletivas, nos períodos de 2 a 31 de janeiro e de 2 a 31 de julho.

§ 2º Aos Juízes de Direito se atribuem sessenta (60) dias de férias individuais, anualmente de acordo com a escala organizada pelo Tribunal de Justiça, no mês de Dezembro.

Art. 199. Se a necessidade do serviço não lhes permitir gozo de férias coletivas, gozam-se individualmente, em período de trinta (30) dias, o Presidente, Vice-Presidente do Tribunal e o Corregedor, bem assim os magistrados que servem no Tribunal Regional Eleitoral, na forma que for estabelecida pelo Tribunal de Justiça.

§ 1º As férias individuais não podem fragmentar-se em períodos inferiores a trinta (30) dias, e somente se acumulam por imperiosa necessidade do serviço e pelo prazo máximo de dois (2) meses.

§ 2º É vedado o afastamento do Tribunal ou de qualquer de seus órgãos judicantes, em gozo de férias individuais no mesmo período, de membro em número que possa comprometer o quorum para julgamento.

Art. 200. São feriados forenses os domingos e os dias de sexta-feira e Sábado da semana santa, os de festas nacionais, estaduais, e municipais e os como tal especialmente decretados.

Art. 201. As autoridades judiciárias e os serventuários da Justiça, no período de férias coletivas, não podem ausentar-se de suas circunscrições senão para lugares donde lhes seja possível voltar ao trabalho dentro de vinte e quatro horas.

Art. 202. Para efeito de protesto de títulos os Cartórios de Notas do Estado não funcionam aos sábados.

Art. 203. No período de férias coletivas e nos dias de feriados não se praticam atos judiciais.

§ 1º Excetua-se.

I – as medidas cautelares;

II – as citações, protestos e interpelações, os quais, no entanto, para fluência dos prazos deles decorrentes e para os efeitos de comparecimento do citado, em Juízo, se têm como feitos no primeiro dia útil;

III – os arrestos, penhores, sequestros e arrecadações, buscas e apreensões, depósitos, detenções pessoais, abertura de testamento, embargos de obra nova e atos análogos;

IV – *habeas corpus*, mandados de segurança, processos e recursos crimes, prisões, fianças e soltura de presos.

§ 2º Além dos atos enumerados no parágrafo anterior, podem ser processados e julgados no período de férias coletivas e não se suspendem pela superveniência destas:

I – as causas de alimentos provisionais, desapropriações, impedimentos matrimoniais, separação judicial, divórcio, nulidade e anulação de casamento, acidentes do trabalho, soldadas, ações possessórias de rito especial, inventários e partilhas, falências e concordatas preventivas;

II – nomeação e remoção de tutores e curadores;

III – as ações prescritíveis em tempo não superior a um mês;

IV – os atos de jurisdição voluntária a todos aqueles necessários à conservação do direitos, e possam prejudicar-se com o adiamento;

V – as sessões do Júri e seus atos preparatórios e os de polícia judiciária ou administrativa;

VI – o processo de relevação e a execução das multas impostas aos jurados.

## **Seção XLI**

### **Das Incompatibilidades e Suspeições**

- No texto publicado, por equívoco, passa-se da Seção XXXIX diretamente à Seção XLI.

Art. 204. Não podem ter assento, simultaneamente, no Tribunal de Justiça, Desembargadores parentes ou afins em linha reta, ou na colateral, até o terceiro grau, inclusive.

Art. 205. A incompatibilidade se resolve:

I – antes da posse, contra o último nomeado ou menos idoso, sendo a nomeação da mesma data;

II – depois da posse, contra o que deu causa à incompatibilidade, se for o imputável a ambas, contra o de investidura mais recente.

Art. 206. Na mesma Comarca não podem servir ao mesmo tempo como Juiz e substituto os que sejam parentes ou afins em grau indicado no art. 204 bem assim marido e mulher.

Parágrafo único. A mesma incompatibilidade existe quando o parentesco for entre o Juiz ou o seu substituto e os serventuários da Justiça.

Art. 207. Não podem requerer nem funcionar como advogados os que forem cônjuges, parentes e afins de Juiz nos graus indicados.

§ 1º Fica o Juiz impedido, se o trabalho do advogado se der em virtude de distribuição obrigatória ou de ter sido antes da propositura da ação, constituída procurador do réu, salvo se a indicação for procurada maliciosamente.

§ 2º A incompatibilidade se resolve contra o advogado, se este estiver de intervir no curso da causa em primeiro ou segundo grau de jurisdição, ou em primeira ou segunda instância.

Art. 208. São nulos os atos praticados pelo Juiz, depois de se tornar incompatível.

Art. 209. O Juíz deve dar-se de suspeito e, se não o fizer, pode como tal ser recusado, por qualquer das partes, nos casos legais.

Art. 210. Também será impedido o Juíz de funcionar:

I – se tiver oficiando na causa como órgão do Ministério Público, advogado, árbitro ou perito, ou nesta qualidades tiver servido parente seu em grau que o torne incompatível;

II – se tiver funcionado na causa como Juiz de outra instância ou grau, pronunciando-se sobre a mesma questão, de fato ou de direito, submetido a julgamento.

Art. 211. Pode o Juiz dar-se por suspeito, se afirmar a existência por motivo de ordem íntima, sem necessidade de expor o motivo, quando se tratar de questão civil.

Art. 212. A suspeição, sob pena de nulidade, será restrita aos casos enumerados e sempre motivada, salvo o disposto no artigo anterior.

Art. 213. O Juiz deve declarar nos autos os motivos quando for incompatível, ou tiver impedimento legal para funcionar.

Art. 214. Os promotores não podem advogar em causas em que seja obrigatório, em primeira instância, a intervenção do Ministério Público, por qualquer de seus órgãos.

§ 1º Não podem também servir em júízo de cujo titular sejam cônjuges, ascendentes, descendentes ou colaterais, até o terceiro grau inclusive, por consanguidade ou afinidade, resolvendo-se a incompatibilidade por permuta ou transferência, conforme o caso.

§ 2º São nulos os atos praticados pelo Promotores depois que se tornam incompatíveis.

Art. 215. Os membros do Ministério Público são impedidos de funcionar como advogado em causas cíveis contra interesse de menores, ausentes ou interditos, declarados por atos judiciais, ainda que tenham de intervir nelas em razão do ofício, nem contra os interesses de vítima de acidente de trabalho ou seus sucessores ou beneficiários ou de pobre em qualquer Juízo ou instância.

Art. 216. É vedada também ao membro do Ministério Público a advocacia em juízo criminal, que a ação seja pública, ou privada, ainda no caso de estar ele de licença ou férias.

Art. 217. Aos órgãos do Ministério Público aplicam-se as prescrições relativas às suspeições dos Juízes.

Art. 218. O membro do Ministério Público deve declarar nos autos os motivos, quando for incompatível ou tiver impedimento legal para funcionar.

Art. 219. Aos serventurários e funcionários da Justiça são extensivos os dispositivos sobre suspeições dos Juízes no que for aplicável.

Art. 220. São nulos os atos aplicados pelos serventurários e funcionários incompatíveis.

## **Seção XLII** **Da Aposentadoria e Disponibilidade**

Art. 221. *(Revogado pelo art. 10 da Lei Complementar n. 88, de 05/09/2007, publicada no DOE nº 170, de 06/09/2007, pp. 1/2.)*

Art. 222. *(Revogado pelo art. 10 da Lei Complementar n. 88, de 05/09/2007, publicada no DOE nº 170, de 06/09/2007, pp. 1/2.)*

Art. 223. Aplicam-se aos serventurários e servidores da Justiça as normas do Estatuto dos Funcionários Públicos Cíveis do Estado e estes sobre a contagem de tempo, quando não colidirem com as disposições desta Lei.

Art. 224. As autoridades judiciárias são aposentadas compulsoriamente aos setenta anos de idade.

§ 1º Aposentam-se, ainda, antes da referida idade, quando estiverem inválidas para o serviço;

§ 2º A aposentadoria por invalidez decreta-se compulsoriamente, quando comprovada a incapacidade por inspeção de saúde ou a requerimento do Procurador Geral da Justiça, deferida pelo Tribunal de Justiça, ou ordenada por este de ofício.

§ 3º Na recusa do magistrado em submeter-se a inspeção de saúde, presume-se a invalidez para a aposentadoria.

§ 4º Nos Casos de doença grave, contagiosa ou incurável, indicada no Estatuto dos Funcionários Públicos Cíveis do Estado, ou por acidente ocorrido no serviço ou por moléstia profissional, licencia-se o magistrado compulsoriamente com vencimentos integrais por prazo não inferior a seis meses nem superior a um ano, ao fim do qual se submete a segundo exame, se for reconhecida a invalidez ou a incapacidade para o exercício da função, converte-se a licença em aposentadoria, com vencimentos integrais.

§ 5º Os proventos da aposentadoria serão reajustados na mesma proporção dos aumentos dos vencimentos concedidos, a qualquer título, aos magistrados em atividade.

Art. 225. Independentemente de prova de invalidez, concede-se a aposentadoria a requerimento do magistrado que tiver mais de trinta anos de serviço público, inclusive com vantagens desta Lei.

Art. 226. Em qualquer dos casos enumerados nos artigos precedentes as autoridades judiciárias deixam o exercício das funções no dia em que apresentam o pedido de aposentadoria, ou delas se afastam por ordem superior, ou quando completam setenta anos de idade.

Parágrafo único. Continuam, porém, a perceber os seus vencimentos, na dotação por meio da qual eram pagos, até que o Tribunal de Contas do Estado julgue a aposentadoria em definitivo.

Art. 227. O tempo de serviço de advocacia computa-se até dez anos para efeito de aposentadoria, disponibilidade e adicionais de magistrados, serventuários da Justiça, vedada a contagem cumulativa.

Art. 228. A aposentadoria do magistrado por limite de idade se decreta por provocação do interessado, a requerimento do Ministério Público ou de ofício.

§ 1º Quando requerida, o processo obedece às normas adotadas para aposentadoria por invalidez, dispensado o exame de saúde, juntando o interessado ao seu requerimento a liquidação do tempo de serviço.

§ 2º Se não for a aposentadoria requerida até o dia seguinte ao que completada a idade limite, o Tribunal de Justiça, de ofício, ou a requerimento do Procurador Geral da Justiça, deve decretá-la, hipótese em que a liquidação de tempo de serviço para o cálculo das vantagens da aposentadoria se ultima em vista da aprovação da autoridade que tiver tomado a iniciativa do processo, trinta dias depois de publicado, o respectivo Decreto no Diário da Justiça.

§ 3º Assegura-se ao interessado o direito de provar, documentadamente, os defeitos dos assentamentos individuais.

Art. 229. O pedido de aposentadoria do magistrado apresenta-se ao Presidente do Tribunal de Justiça, instruído com a liquidação do tempo de serviço e, devidamente informado, vai remetido ao Chefe do Poder Executivo, para as providências legais.

Art. 230. Após o julgamento da legalidade da aposentadoria pelo Tribunal de Contas, e a lavratura e publicação do Decreto respectivo, devolve-se o processo ao Tribunal de Justiça, para arquivamento.

Art. 231. Os proventos da inatividade serão revistos sempre que, por motivo de alteração, do poder aquisitivo da moeda se modificarem os vencimentos dos funcionários ativos.

Art. 232. Os proventos dos magistrados são iguais aos vencimentos dos em atividade, compreendidas todas as vantagens da categoria correspondente,

Parágrafo único. Aplica-se o disposto neste artigo às aposentadorias com vencimentos não integrais, observada a proporção nela estabelecida.

Art. 233. Deve constar do Orçamento anual do Estado dotação consignada ao Tribunal de Justiça, para atender o pagamento dos magistrados inativos.

Art. 234. No caso de mudança de sede de Juízo ou de supressão da Comarca é facultada ao Juiz remover-se para a nova sede ou para a Comarca de igual entrância ou pedir disponibilidade com vencimentos integrais.

Parágrafo único. A disponibilidade, nesses casos, é requerida ao Tribunal de Justiça, que depois de processar o pedido o encaminha ao Chefe do Poder Executivo, para os fins convenientes.

Art. 235. A aposentadoria dos serventuários da Justiça obedece às disposições da legislação especial já expedida e em vigor, observados os textos constitucionais.

Art. 236. *(Revogado pelo art. 10 da Lei Complementar n. 88, de 05/09/2007, publicada no DOE nº 170, de 06/09/2007, pp. 1/2.)*

Art. 237. *(Revogado pelo art. 10 da Lei Complementar n. 88, de 05/09/2007, publicada no DOE nº 170, de 06/09/2007, pp. 1/2.)*

### **TÍTULO III DAS GARANTIAS E DIREITOS**

#### **CAPÍTULO I DAS GARANTIAS**

Art. 238. Salvo as restrições expressas nesta Lei, os Juízes gozam das seguintes garantias:

- I – vitaliciedade, não podendo perder o cargo senão por sentença judiciária;
- II – inamovibilidade, exceto por interesse público;
- III – irredutibilidade de vencimentos, sujeitos, entretanto, aos impostos gerais, inclusive o de renda e os impostos extraordinários.

#### **CAPÍTULO II DOS DIREITOS**

Art. 239. Os pagamentos devidos pela Fazenda Federal, Estadual ou Municipal, em virtude de sentença judiciária, se fazem na ordem de apresentação dos precatórios e por conta das dotações orçamentárias próprias, proibida a designação de casos ou pessoas nas dotações orçamentárias e nos créditos extra-orçamentários abertos para esse fim.

§ 1º É obrigatória a inclusão, no orçamento das entidades de direito público, de verbas necessárias ao pagamento dos seus débitos, constantes dos precatórios judiciais, apresentados até primeiro de julho.

§ 2º As dotações orçamentárias e os créditos abertos são consignados ao Poder Judiciário, recolhendo-se as importâncias respectivas à repartição competente. Cabe ao Presidente do Tribunal que proferir a decisão exequenda determinar o pagamento, segundo as possibilidades do depósito, e autorizar, a requerimento do credor preterido no seu direito de precedência, ouvido o Chefe do Ministério Público, o sequestro da quantia necessária à satisfação do débito.

Art. 240. No caso de prisão em flagrante de qualquer autoridade judiciária, os autos devem ser encaminhados, dentro de quarenta e oito horas ao Presidente do Tribunal de Justiça, que procede na forma do art. 310 do Código do Processo Penal, ouvido o Procurador Geral da Justiça, em vinte e quatro horas.

§ 1º A autoridade judiciária que for detida em flagrante de crime inafiançável, fica, desde o momento da detenção, sob custódia do Presidente do Tribunal de Justiça.

§ 2º Se forem necessárias investigações ou diligências complementares, o Corregedor da Justiça as executará.

Art. 241. Os Juízes podem recusar as promoções, conservando-se nos seus cargos, caso em que se promove o imediato, se a promoção for pelo critério de antiguidade; completa a lista, se a vaga for por merecimento serão observadas as disposições legais.

Art. 242. O Desembargador pode, concordando o Tribunal, ser removido, a seu requerimento, de uma Câmara para outra, em caso de vaga, mediante permuta.

Art. 243. Os magistrados só perdem os seus cargos quando exonerados a pedido, ou por sentença judicial condenatória passada em julgado, em caso de crime doloso, ou de responsabilidade; os serventuários e funcionários em iguais circunstâncias e ainda mediante inquérito administrativo, em que lhes assegura ampla defesa.

### **CAPÍTULO III DOS DEVERES E SANÇÕES**

Art. 244. Os magistrados devem manter irrepreensível procedimento na vida pública e particular, zelando pela dignidade de suas funções.

Art. 245. Os magistrados usam vestes talares durante os julgamentos do Tribunal de Justiça, no Tribunal do Júri, nas audiências do Juízo e quando presidem a realização de casamento.

Art. 246. Os Juízes devem permanecer na sede dos seus Juizados durante o horário do expediente e quando necessário.

Parágrafo único. Os Juízes só podem sair da sede da Comarca ou Zona, a objeto de serviço ou a chamado do Presidente do Tribunal, ou do Corregedor da Justiça.

Art. 247. Pelas faltas cometidas no cumprimento dos deveres, além da responsabilidade civil e penal em que incorrem, ficam os magistrados sujeitos às sanções da lei.

Art. 248. Incorre também em culpa grave o magistrado que não punir as faltas dos seus subordinados ou não providenciar como de direito, para que se lhes imponha a sanção disciplinar ou penal, pelos órgãos judiciários competentes.

Art. 249. A autoridade judiciária que exceder os prazos legais, para sentenciar ou despachar, incorre ainda em sanções estabelecidas na legislação processual civil e penal.

Art. 250. Devem os serventuários da Justiça exercer com dignidade e compostura seus ofícios, obedecendo às ordens de seus superiores, cumprindo as disposições legais e observando, fielmente, o Regimento de Custas.

Art. 251. Pelas faltas cometidas no cumprimento dos deveres, os serventuários ficam sujeitos às seguintes penas disciplinares, aplicadas de ofício ou em virtude de reclamação ou representação das partes interessadas:

I – advertência particular ou pública;

II – representação;

- III – censura pública;
- IV – multa até seis meses com perda total ou parcial das vantagens do cargo;
- V – multa até um valor de referência regional;
- VI – perda do cargo.

Parágrafo único. *(Revogado pelo art. 10 da Lei Complementar n. 88, de 05/09/2007, publicada no DOE nº 170, de 06/09/2007, pp. 1/2.)*

Art. 252. No caso de falta grave, de notória incontinência de conduta ou de terceira pena de suspensão, e nos de que possa resultar a pena de perdas do cargo, os serventuários da Justiça são precessados perante o Juiz local, dando-se publicidade da ocorrência e garantia ampla de defesa ao acusado.

Art. 253. Os deveres, sanções e penas disciplinares referentes aos serventuários da Justiça são regulados pelo que dispõe esta Lei, o Regimento Interno do Tribunal de Justiça e subsidiariamente o Estatuto dos Funcionários Públicos Civis do Estado.

§ 1º Nas comarcas do interior, onde houver mais de um Juiz de Direito, mediante representação ou de ofício, cabe ao Diretor do Forum determinar a abertura do inquérito administrativo.

§ 2º Tratando-se de funcionário da Diretoria Geral da Secretaria do Tribunal de Justiça, a competência é do Presidente deste.

Art. 254. No processo de que trata o artigo precedente, observa-se o que, a respeito de inquérito administrativo, esta disposto no Estatuto dos Funcionários Públicos Civil do Estado.

Art. 255. Os deveres, sanções e penas disciplinares referentes aos funcionários da Justiça de primeira instância e aos da Diretoria Geral da Secretaria do Tribunal de Justiça, são regulados pelo disposto no Estatuto dos Funcionários Públicos Civil do Estado e no Regimento Interno do Tribunal de Justiça.

Art. 256. Em todos os casos em que, além da falta disciplinar, houve indício de prova de crime a punir, depois de aplicada a pena disciplinar, os documentos, autos ou papéis devem ser enviados à autoridade competente para promover a ação penal contra o responsável.

Art. 257. As penas disciplinares são impostas:

I – pelo Tribunal de Justiça a seu Presidente, Desembargadores, Corregedor da Justiça, Juizes e a qualquer serventuário ou funcionário da Justiça de primeira instância ou da Diretoria Geral da Secretaria do Tribunal de Justiça;

II – pelas Câmaras Reunidas e Câmaras Especializadas aos Juizes e a qualquer serventuário ou funcionário da Justiça de primeira instância ou funcionário da Diretoria Geral da Secretaria do Tribunal de Justiça, quando cometerem falta em autos submetidos ao seu julgamento;

III – pelo Presidente do Tribunal de Justiça aos Juizes, serventuários e funcionários da Justiça de primeira instância e aos funcionários da Diretoria Geral da Secretaria do Tribunal de Justiça;

IV – pelo Conselho da Magistratura aos Juizes, serventuários e funcionários da Justiça de primeira instância;

V – pelo Diretor do Forum aos serventuários e funcionários da Justiça de primeira instância;



VI – pelo Diretor do Forum aos serventuários e funcionários da Justiça de primeira instância da Comarca respectiva;

VII – pelos Juízes de Direito aos Juízes de Paz e aos serventuários e funcionários da Justiça de suas respectivas Comarcas;

VIII – pelos Juízes de Paz aos serventuários e auxiliares de seu termo judiciário.

Art. 258. Da imposição de pena disciplinar cabe recurso voluntário, com efeito devolutivo somente, no prazo de cinco dias de ciência de ato para:

I – o Tribunal de Justiça, quando a pena for imposta pelo Presidente, Câmaras Reunidas ou Câmaras Especializadas e pelo Conselho da Magistratura;

II – o Conselho da Magistratura, quando a pena for imposta pelo Corregedor;

III – o Corregedor, quando a pena for imposta pelos Juízes de Direito ou pelos Juízes de Direito Adjunto;

IV – o Juiz em exercício, quando a pena for imposta pelos Juízes de Paz;

§ 1º Em matéria de recurso disciplinar só são admitidas duas (2) instâncias imponente da pena e aquela para a qual se recorre; nesta, o recurso se exaure completamente.

§ 2º Quando se tratar de pena disciplinar imposta em única instância pelo Tribunal de Justiça, admite-se pedido de reconsideração dentro de cinco dias, a partir da ciência pelo punido.

Art. 259. O recurso de pena disciplinar, apresentado à autoridade que a impõe, será, se for tempestivo, encaminhado a quem tenha competência para julgá-lo, com ou sem razões de sustentação do ato de quem aplicou a penalidade.

Art. 260. Somente depois de passado em julgado, lança-se a pena disciplinar no assentamento individual do punido.

#### **TÍTULO IV DISPOSIÇÕES GERAIS, FINAIS E TRANSITÓRIAS**

Art. 261. Na Comarca da Capital há os seguintes serventuários e funcionários da Justiça:

a) seis Tabeliães de Notas, denominado Primeiro, Segundo, Terceiro, Quarto, Quinto e Sexto Ofícios, com os encargos cumulativos exercidos por distribuição, sendo os terceiro, quinto e sexto privativo do Oficial do Registro de Protestos de Letras e outros títulos. O primeiro, segundo e quarto Tabeliães de Notas exercem as funções de Oficial do Registro de Imóveis, pela forma seguinte:

1. 1º Tabelião – 2ª Circunscrição;

2. 2º Tabelião – 3ª Circunscrição;

3. 4º Tabelião – 1ª Circunscrição.

Art. 262. A Comarca de Teresina é dividida em quatro circunscrições para efeito de Registro de Nascimento, Casamento e óbitos, a saber:

a) a primeira circunscrição compreende a área situada ao Norte da Rua Senador Teodoro Pacheco e seu prolongamento pela Avenidas Antonino Freire e Frei Serafim, até o Rio Poti, daí seguindo à jusante, pela margem esquerda, até a sua desembocadura do Rio Parnaíba;

b) a segunda circunscrição compreende a área situada ao lado sul da Rua Senador Teodoro Pacheco e seu prolongamento pela Avenidas Antonino Freire e Frei Serafim, até o

Rio Poti, por este seguindo, à margem esquerda até encontrar, na sua montante, a Avenida Industrial Gil Martins, por esta seguindo até o Rio Parnaíba;

c) a terceira circunscrição compreende toda a área urbana e rural não incluída nas primeira, segunda e quarta circunscrições;

d) a quarta circunscrição compreende a área sul do município que partindo da localidade Salobro, segue, a oeste, pelo Rio Parnaíba, até os limites de Palmeirais; por este segue até os limites de Teresina com Monsenhor Gil e Demerval Lobão, até o ponto em que se encontram com a BR-316. No Norte o limite é uma linha deste ponto ao lugar salobro, em diagonal.

Parágrafo único. O povoado Nazária é a sede da quarta circunscrição enquanto as demais têm sua sede na zona urbana de Teresina.

Art. 263. Para efeito de Registro de Imóvel há na Comarca de Teresina três circunscrições.

§ 1º A primeira e segunda circunscrição coincidem com as do artigo precedente.

§ 2º A terceira circunscrição compreende toda a área urbana e rural não incluída nas primeira e segunda circunscrições.

§ 3º Enquanto não efetivar a vigência do disposto neste artigo, o Registro de Imóvel, na Capital continua sendo feito pelos primeiro e quarto tabeliães de Notas, na forma da legislação anterior.

Art. 264. Na Comarca de Parnaíba passam a ser observadas as seguintes disposições em relação aos quatro Cartórios da Comarca:

I – o Protesto de Título será privativo dos Cartórios do 1º e 3º Ofícios;

II – o Registro Imobiliário continuará privativo dos Cartórios do 1º e 4º Ofícios;

III – o Registro de Títulos e Documentos e demais registros de que trata a Lei dos Registros Públicos, com exceção dos Registros Imobiliários e Registro Civil, serão privativo do Cartório do 2º Ofício:

IV – os feitos cíveis distribuídos às 1ª e 2ª Varas, serão privativos do Cartório do 2º Ofício e os distribuídos às demais Varas, privativos do Cartório do 3º Ofício.

Parágrafo único. A 1ª Circunscrição do Registro Imobiliário é privativa do Cartório do 1º Ofício. Os seus limites abrangem o restante dos terrenos não situados na delimitação da 2ª Circunscrição do Registro Imobiliário, privativa do Cartório do 4º Ofício que é a seguinte: Toda a Ilha Grande de Santa Isabel e mais parte da zona sul da cidade, com a seguinte delimitação: a partir do entroncamento da Av. São Sebastião com os trilhos da rede ferroviária, seguindo em linha reta, em direção leste, pela Av. São Sebastião (lado da Igreja do mesmo nome) até os limites do município de Luiz Correia. Em direção sul, margeando sempre o leito da estrada de ferro (lado esquerdo no sentido Parnaíba/Teresina) até encontrar o cruzamento da BR 343 com a estrada municipal que liga Parnaíba ao bairro Rosápolis daí seguindo sempre pelo mesmo lado esquerdo da mencionada BR 343, na direção Parnaíba/Teresina, até encontrar com terrenos dos municípios de Luiz Correia e Burito dos Lopes.

Art. 265. As varas, comarcas e zonas criadas por esta Lei devem instalar-se à medida em que permita a situação econômica-financeira do Estado, em data acordada entre os Poderes Judiciário e Executivo.

Art. 266. Enquanto não forem instaladas as zonas, comarcas e varas criadas nesta Lei, permanecem a atual jurisdição e competência de cada qual.

Art. 267. As audiências dos Juízes verificam-se nos dias úteis, entre as nove e às dezoito horas, na sede do Juízo ou, excepcionalmente, na local que o Juíz designar, e são públicas, presentes o Escrivão, o Porteiro dos Auditórios e os Oficiais de Justiça.

Art. 268. Se da publicidade da audiência ou da sessão, pela natureza do processo, resultar escândalo, inconveniente grave, ou perigo para a ordem pública, o Juíz pode, de ofício, ou a requerimento do interessado do Ministério Público, ordenar que uma ou outra se efetue a portas fechadas ou limitar o número de pessoas que a ela possam assistir, sendo, em todo caso, permitida a presença das partes e seus procuradores.

§ 1º A determinação do Juíz, em qualquer hipótese, é inserta no termo da audiência e nos autos do processo.

§ 2º Nos processos contra menores de dezoito anos as audiências se fazem em segredo de justiça.

Art. 269. À hora marcada, o Juiz determina que o Porteiro dos Auditórios, ou o Oficial de Justiça declare aberta a audiência apregoando as partes, cujo comparecimento for obrigatório, e, sendo o caso, o órgão do Ministério Público, os peritos, seguindo-se o estabelecimento nos Códigos de Processo Civil e Penal.

Art. 270. No recinto reservado às audiências ou sessões somente podem tomar assento, além do Juíz e representante do Ministério Público, Escrivão, Advogados, Peritos e pessoas que forem judicialmente convocadas.

Parágrafo único. Durante as audiências, sessões ou ato a que presidir o Juíz, o Porteiro e os Oficiais de Justiça devem permanecer no local para cumprimento de ordens.

Art. 271. Os presentes às audiências têm de manter-se respeitosamente e em silêncio, sendo-lhe vedada qualquer manifestação de aquiescência ou reprovação.

Art. 272. A polícia da audiência ou sessão compete ao Juiz dela Presidente, que deve exigir o que convier à ordem e ao respeito, podendo, se preciso for, requisitar a força policial, que ficará a sua disposição.

Art. 273. Em caso de desacato ou desobediência, o Juiz pode expulsar do recinto os culpados, devendo, além disso, prendê-los se for o caso, e lavrar o respectivo auto de prisão em flagrante delito, para que sejam processados.

Art. 274. Os serventuários da Justiça devem entregar, por inventário, ao seu substituto em definitivo ou provisório, os livros e papéis do Cartório.

§ 1º No caso de recusa, o Juiz promove as diligências para entrega do Cartório e a responsabilidade do recusante, podendo determinar que outros serventuários procedam ao inventário.

§ 2º Na hipótese de abandono do cargo ou morte, o Juiz adota providência sobre a entrega, expedindo a necessária portaria.

§ 3º Tratando-se de Diretor Geral ou Diretoria da Secretaria do Tribunal de Justiça ou de Serventuário da Justiça sujeito a jurisdição privativa, as atribuições constantes dos parágrafos primeiro e segundo são exercidas pelo Presidente do Tribunal e pelo Juiz respectivamente.

Art. 275. Em cada Cartório há um livro próprio para registro das sentenças que puserem termo ao feito, ainda que delas as interponha recurso.

§ 1º O prazo para o escrivão registrar a sentença é de três dias a contar da data em que o Juiz houver entregue os autos.

§ 2º Quando a sentença passar em julgado na primeira instância, o Escrivão assim o certifica no registro dentro de vinte e quatro (24) horas.

Art. 276 Rubrica-se todas as folhas do processo em que não houver assinatura do escrivão, exceto-as em que estiver lançada a decisão do Juiz.

Parágrafo único. O Juiz rubrica as folhas dos autos em que intervier, salvo aquelas onde haja sua assinatura; o advogado da parte pode rubricar qualquer folha de autos.

Art. 277. Os autos em andamento no Tribunal ou em qualquer Juízo só podem sair do Cartório conclusos a Desembargador ou a Juiz ou com vista aos representantes do Ministério Público, Curador à lide, defensores ou advogados constituídos pelas partes, mediante protocolo de entrega, do qual fiquem constando a data e o recibo daquele a quem foram remetidos.

§ 1º A entrega de autos findos a Desembargador, Juiz ou representante do Ministério Público, também depende de protocolo com data e recibo de quem os receber.

§ 2º Os advogados podem retirar autos findos para exame, deixando o competente recibo no protocolo. Findo o prazo marcado, devem restituí-los, sujeitando-se os remissos, às sanções administrativas, civis e penais para tornar efetiva a volta dos autos a cartório.

§ 3º Em qualquer hipótese, o funcionário da Secretaria ou Cartório que receber os autos de volta há de rever a numeração das folhas, certificando quaisquer irregularidades encontradas.

§ 4º O Diretor Geral e os Diretores da Secretaria do Tribunal e os titulares de Cartório se sujeitam as sanções disciplinares, cíveis e criminais, caso entreguem autos findos, ou em andamento, sem protocolo. A desobediência ao disposto neste artigo e seus parágrafos importa, sem prejuízo das demais cominações legais, em falta grave, punível com suspensão.

Art. 278. Em nenhum caso ficam prejudicados os recurso interpostos pelas partes, quando por erro ou omissão do Oficial de Justiça, ou de outros servidores, não tiverem seguimento ou não forem apresentados em tempo ao Juiz *ad quem*.

Art. 279. Pela duplicata de autos do escrivão, para que se formem os suplementares, as custas são devidas na proporção de um terço.

Art. 280. Nas comarcas onde houver mais de uma vara fica estabelecido plantão judiciário para os *habeas corpus* e outras medidas de caráter urgente inadiável.

Parágrafo único. O plantão judiciário da Capital é organizado semanalmente pelo Corregedor da Justiça com os Juízes das Varas Criminais; no interior, a escala compete ao Diretor do Forum.

Art. 281. Aos sábados, o expediente forense se encerra às doze horas, salvo para casamento e atos do registro civil que podem ser realizados depois desse horário e nos domingos e feriados.

Art. 282. É assegurado à família dos magistrados em atividade, aposentados, ou em disponibilidade, o direito, por falecimento, deles, a dois meses de vencimentos do morto, a título de despesas funerárias, pagas pelos cofres públicos do Estado, sem prejuízo de qualquer pecúlio, ou benefício devido em virtude de Lei.

Art. 283. É removida ou designada para servir na sede onde residir o marido, a funcionária pública estadual casada com magistrado, sem prejuízo de quaisquer direitos ou vantagens do cargo.

Parágrafo único. Não havendo vaga nos quadros da repartição a que pertence, a funcionária fica adida a qualquer serviço público estadual existente na sede da comarca.

Art. 284. O Diário da Justiça, órgão oficial do Poder Judiciário destina-se à publicação dos atos judiciais para os efeitos previstos em Lei.

Art. 285. As certidões fornecidas pela Diretoria Geral da Secretaria Geral do Tribunal de Justiça fazem prova bastante na contagem de tempo de serviço dos magistrados, para todos os efeitos legais, inclusive concessão de adicionais e deferimento de aposentadoria.

Art. 286. Destina-se, no orçamento do Poder Judiciário, verba especial para as despesas com as sessões do Tribunal do Júri, distribuída pelo Presidente do Tribunal de Justiça também é às comarcas do interior do Estado.

Art. 287. Cabe ao Tribunal de Justiça promover a reforma do seu Regimento Interno e dos demais órgãos do Poder Judiciário, e elaborar o regulamento de sua Secretaria, para adaptá-lo a presente Lei.

Art. 288. Em decorrência da presente Lei ficam criados os cargos a seguir discriminados:

I – nove cargos de Juiz de Direito de 4ª entrância;

II – quatro cargos de Juiz de Direito de 3ª entrância;

III – vinte e três cargos de Juiz de Direito de 1ª entrância;

IV – seis cargos de Juiz de Direito Adjunto;

V – mais um cargo de Tabelião de Notas e Registro Civil nas comarcas de Altos, Amarante, Barras Corrente, Campo Maior, Jose de Freitas, Piracuruca, Piripiri, São João do Piauí, São Raimundo Nonato;

VI – em cada uma das vinte e três comarcas de 1ª entrância a serem instaladas:

a) um cargo de Tabelião de Notas;

b) dois cargos de Oficial de Justiça;

c) um cargo de Distribuidor, Contador e Partidor;

d) um cargo de Avaliador Oficial.

VII – na Comarca de Campo Maior:

a) um cargo de Escrivão do Cível;

b) um cargo de Escrivão do Crime;

c) dois cargos de Oficial de Justiça;

d) um cargo de Comissário de Menores;

e) um cargo de Vigilante de Menores.

VIII – na Comarca de Picos, mais um Cartório denominado 2º Cartório do Registro Civil de Nascimento, Casamentos e Óbitos anexado ao 3º Tabelionato de Notas, cabendo-lhe a privatividade da 2ª circunscrição. Ao 1º Cartório ficam privativos os registros da 1ª circunscrição.

IX – nos termos judiciários de Cajueiro, Nazária, Patos, Várzea Grande, Dirceu Arcoverde, Curral Novo, Capivara, Coivaras e Brasileira, (Art. 5º inciso VI, desta Lei), os cargos de Juiz de Paz e dois suplentes e de Escrivão do Registro Civil de Nascimento, Casamento e Óbitos.

§ 1º Os cargos a que se referem os incisos nºs. I, III, IV e VI somente serão providos quando se instalarem as respectivas varas, comarcas e zonas, observando o disposto no Art. 267, desta Lei.

§ 2º Os atos de escrivães criados na Comarca de Campo Maior tem competência exclusiva para serventia, respectivamente, no cível e no crime, e se substituem reciprocamente.

§ 3º Os atuais titulares de Cartório de Campo Maior são exclusivo das funções notariais e de Registro Públicos, obedecida a presente Lei.

§ 4º A competência e jurisdição dos cartórios constantes do inciso V deste artigo serão definidas oportunamente, mediante proposta do Tribunal de Justiça.

Art. 289. Ficam extintos dois cargos de Juiz de Direito de 2ª entrância.

Art. 290. Os casos omissos serão resolvidos pelo Tribunal de Justiça no seu Regimento Interno.

Art. 291. Esta Lei entrará em vigor na data de sua publicação, revogadas as disposições em contrária.

Palácio do Governo do Estado do Piauí, em Teresina, 12 de dezembro de 1979.

Waldemar de Castro Macêdo (Valdemar de Castro Macêdo)  
Governador do Estado,

Antônio de Almendra Freitas Neto  
Secretário de Governo,

Manoel Leocádio de Melo  
Secretário de Administração.

***Este texto não substitui o publicado no Diário Oficial do Estado nº 237, de 12/12/1979, pp. 1/18.***

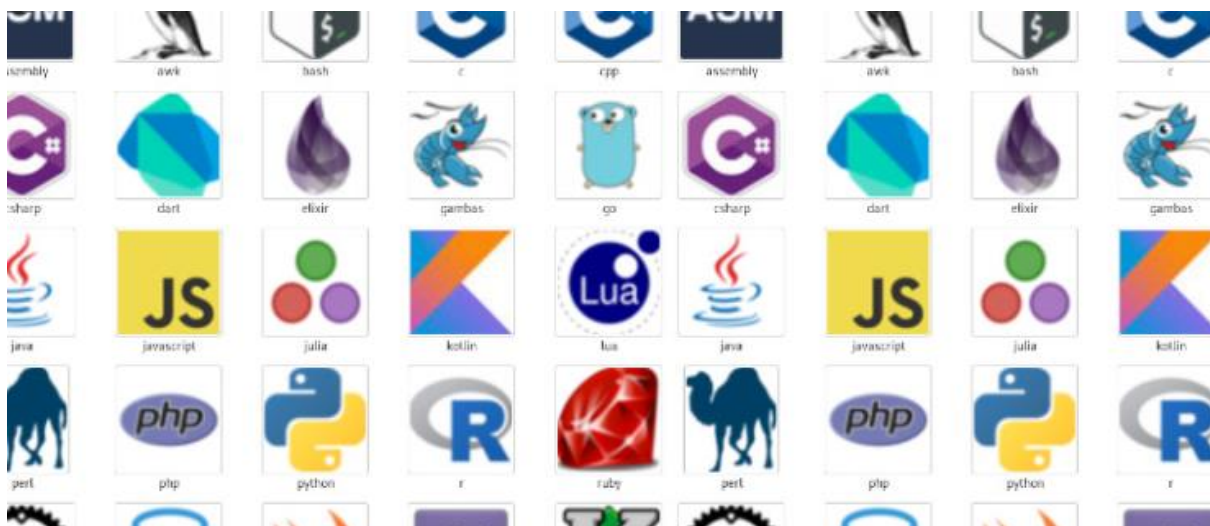
## Conhecimentos específicos

*“Camuflar um erro seu é  
anular a busca pelo  
conhecimento. Aprenda  
com eles e faça novamente  
de forma correta.”*

Nara Nubia Alencar

## Linguagens de Programação

Podemos imaginar o computador como uma super calculadora, capaz de fazer cálculos muito mais rápido que nós, mas para isso devemos dizer para o computador o que deve ser calculado e como deve ser calculado. A função das linguagens de programação é exatamente essa, ou seja, servir de um meio de comunicação entre computadores e humanos.



Existem dois tipos de linguagens de programação: as de **baixo nível** e as de **alto nível**. Os computadores interpretam tudo como números em base binária, ou seja, só entendem zero e um. As linguagens de baixo nível são interpretadas diretamente pelo computador, tendo um resultado rápido, porém é muito difícil e incômodo se trabalhar com elas. Exemplos de linguagens de baixo nível são a linguagem binária e a linguagem **Assembly**.

### Exemplo de Código em Assembly:

```
MOV r0, #0C ;load base address of string into r0
```

```
LOAD: MOV r1,(r0) ;load contents into r1
```

```
CALL PRINT ; call a print routine to print the character in r1
```

```
INC r0 ;point to next character
```

```
JMP LOAD ;load next character
```

Como pode-se notar, é uma linguagem bastante complicada.

Já as linguagens de alto nível são mais fáceis de se trabalhar e de entender, as ações são representadas por palavras de ordem (exemplo faça, imprima, etc) geralmente em inglês, foram feitos assim para facilitar a memorização e a lógica. Elas não são interpretadas diretamente pelo computador, sendo necessário traduzí-las para linguagem binária utilizando-se de um programa chamado **compilador**.

Quando programamos em uma linguagem de programação de alto nível primeiramente criamos um arquivo de texto comum contendo a lógica do programa, ou seja, é onde falamos ao computador como deve ser feito o que queremos. Este arquivo de texto é chamado de **código-fonte**, cada palavra de ordem dentro do código-fonte é chamada de instrução. Após criarmos o código-fonte devemos traduzir este arquivo para linguagem binária usando o compilador correspondente com a linguagem na qual estamos programando. O compilador irá gerar um segundo arquivo que chamamos de executável ou programa, este arquivo gerado é interpretado diretamente pelo computador.

Existem algumas linguagens de programação que não necessitam de compiladores, como o PHP, uma linguagem dedicada à produção de websites dinâmicos, como o InfoEscola. As instruções em PHP são compiladas e executadas ao mesmo tempo.



Cada linguagem de programação é diferente da outra, contendo palavras-chave próprias. Exemplos de linguagens de alto nível são C++, Java, C#, Delphi (Pascal), PHP, Visual Basic, etc.

Computadores são estúpidos. Isso pode vir como uma surpresa, considerando que eles são capazes de fazer cálculos complexos em poucos segundos, de forma que nenhum humano conseguiria. Mas é verdade! Computadores são extremamente estúpidos e precisam de instruções exatas para tudo o que fazem. O que é óbvio para você, humano, certamente não é óbvio para uma máquina. E se você quer que a máquina faça algo pra você, bem, você precisa, de certa forma, "falar com ela".

Você já deve ter ouvido falar que computadores entendem apenas binário, ou seja, a linguagem de 0 e 1. Todas as instruções que são dadas para eles são traduzidas em sinais elétricos que significam, basicamente, ligado e desligado. Pense em uma pequena lâmpada. Quando ela está ligada, indica 0. Quando desligada, indica 1. Isso é binário. E há uma infinita quantidade de combinações que você pode fazer com isso.

É claro, nenhum - talvez algum - humano na face da terra iria programar um Skyrim ou mesmo um simples site da Web utilizando apenas 0 e 1. Isso seria extremamente cansativo e demorado. Pra isso que servem as linguagens de programação.

Pense em você e sua máquina. Você fala português, mas a sua máquina fala apenas binário. Entre você e a máquina está um intérprete, o compilador, mas ele só sabe binário e uma outra língua - Java, por exemplo. Aprender a "falar" Java é muito mais fácil que aprender a "falar" binário. Isso porque Java possui muitas palavras em comum com o inglês, e você pode escrever instruções que você entende, como "imprima" ou "leia", ao invés de 0010100010101010. Assim, quando você quer se comunicar com a sua máquina e criar uma série de instruções para ela seguir, você escreve em Java para o compilador e ele traduz as instruções para a máquina, em binário. Isso é uma linguagem de programação.

Linguagens de programação podem ser catalogadas em dois tipos: de **baixo nível** e **alto nível**. Isso não quer dizer que uma é mais rica que a outra ou algo do tipo. Significa que uma está muito mais próxima da máquina do que a outra.

Chamamos de **linguagem de programação** (ou informática) uma linguagem destinada a descrever o conjunto das ações consecutivas que um computador deve executar. Desta forma, uma linguagem informática é uma maneira prática para nós, humanos, darmos instruções a um computador. Por outro lado, o termo 'linguagem natural' representa as possibilidades de expressão compartilhadas por um grupo de indivíduos (por exemplo, o inglês ou o português).

As linguagens que servem para que os computadores se comuniquem entre si não têm nada a ver com linguagens informáticas. Neste caso, falamos de protocolos de comunicação, que são duas noções totalmente diferentes. Em uma linguagem de programação rigorosa, cada instrução gera uma ação do processador.

A linguagem utilizada pelo processador chama-se **linguagem das máquinas**. Trata-se de dados tal como chegam ao processador, constituídos por uma sequência de 0 e 1 (dados binários). Assim, ela não é compreensível por um ser humano e é por isso que foram criadas linguagens intermediárias, compreensíveis por homens. O código escrito neste tipo de linguagem é transformado em linguagem das máquinas para poder ser usada pelo processador.

A **linguagem de montagem** foi a primeira linguagem de programação a ser usada. Ela é parecida com a linguagem das máquinas, mas é mais compreensível pelos programadores. Ela é tão parecida com a linguagem das máquinas que depende do tipo de processador utilizado, já que cada um pode ter a sua própria linguagem das máquinas. Assim, um programa desenvolvido para uma máquina não poderá ser aplicado a outro tipo de máquina. O termo **portabilidade** designa a aptidão que um programa informático tem em ser utilizado em máquinas de tipos diferentes. Às vezes, para poder utilizar um programa informático escrito em linguagem de montagem em outro tipo de máquina, o programa deverá ser reescrito completamente.

Uma linguagem informática tem, então, várias vantagens, ou seja, ela é mais fácil de entender se comparada à linguagem das máquinas e, sobretudo, ela permite uma maior portabilidade, isto é, uma maior facilidade de adaptação nas máquinas de tipos diferentes.

## Quais são os Tipos de Linguagem de Programação

As linguagens de programação estão divididas em duas grandes famílias, dependendo da maneira como as instruções são processadas: as **linguagens imperativas** e as **linguagens funcionais**.

### Como Funciona a Linguagem Imperativa

A **linguagem imperativa** organiza o programa através de uma série de instruções, reunidas por blocos e compreendendo saltos condicionais que permitem retornar a um bloco de instruções se a condição for realizada. Historicamente, são as primeiras linguagens, mesmo se várias linguagens modernas ainda utilizem este princípio de funcionamento. Porém, as linguagens imperativas estruturadas sofrem da falta de flexibilidade, dado o caráter sequencial das instruções.

### Como é Construída a Linguagem Funcional

A **linguagem funcional** tem um nível elevado de abstração, já que elimina muitos detalhes da programação, diminuindo assim a eventualidade de erros. Por quê? Como não depende das operações de atribuição, este tipo de linguagem evita estados ou dados mutáveis.

Ela se baseia na aplicação de funções e, desta forma, qualquer função pode ter ou não regras e um simples valor de retorno. As regras são os valores de entrada da função e o valor de retorno é o resultado desta função. A determinação de uma função vai dizer como ela será avaliada em relação às outras, não exigindo definições adicionais. A linguagem funcional é, pelas suas características, mais simples para tratar das provas e análises matemáticas do que a linguagem imperativa.

### Como é Feita a Interpretação e a Compilação

As linguagens de programação podem ser classificadas em duas categorias: as **linguagens interpretadas** e as **linguagens compiladas**.

#### O Processo da Linguagem Interpretada

Uma linguagem de programação é, por definição, diferente da linguagem das máquinas. Por isso, é necessário traduzi-la para torná-la compreensível para o processador. Um programa escrito numa **linguagem interpretada** precisa de um programa auxiliar (o intérprete) para traduzir progressivamente as instruções recebidas.

#### A Construção da Linguagem Compilada

Um programa escrito numa **linguagem compilada** vai ser traduzido completamente por um programa anexo, chamado compilador, a fim de gerar um novo arquivo autônomo, que não precisará mais de outro programa para ser executado; dizemos, então, que este arquivo é executável.

Um programa escrito em linguagem compilada não precisa, uma vez compilado, de um programa anexo para ser executado. Além disso, como a tradução é feita completamente, a execução é mais rápida. Contudo, ele é menos flexível do que um programa escrito com uma linguagem interpretada, porque a cada modificação do arquivo-fonte (aquele que vai ser compilado) será preciso recopiá-lo para que as alterações tenham efeito.

Por outro lado, um programa compilado tem a vantagem de garantir a segurança do código-fonte. Na verdade, uma linguagem interpretada e compreensível, permite a qualquer programador conhecer os segredos de construção de um programa e, desta forma, copiar ou alterar o código. Por causa disso, existe o risco de desrespeito aos direitos autorais. Certos aplicativos protegidos precisam garantir a confidencialidade do código para evitar o hacking.

#### A Estrutura das Linguagens Intermediárias

Algumas linguagens podem pertencer, de certa forma, às duas categorias (LISP, Java, Python, etc.), já que o programa escrito com estas linguagens pode, em certas condições, sofrer uma fase de compilação intermediária para ser compatível com um arquivo escrito numa linguagem que não é inteligível (logo, diferente do arquivo-fonte) e, por isso, não executável, necessitando de um intérprete. Por

exemplo, os applets Java, pequenos softwares executados na janela de um aplicativo para ampliar as funcionalidades dos navegadores (som, animação, etc.), encontrados nas páginas Web. Eles são arquivos compilados que só podem ser executados a partir de um navegador (arquivos com extensão Class).

### As Linguagens na Prática

Veja, abaixo, a lista das linguagens de programação mais conhecidas:

Linguagem	Domínio de aplicação principal	Compilada/interpretada
ADA	O tempo real	Linguagem compilada
BASIC	Programação básica com objetivos educativos	Linguagem interpretada
C	Programação do sistema	Linguagem compilada
C++	Programação do sistema objeto	Linguagem compilada
Cobol	Gestão	Linguagem compilada
Fortran	Cálculo	Linguagem compilada
Java	Programação orientada para a Internet	Linguagem intermediária
MATLAB	Cálculo matemático	Linguagem interpretada
Matemática	Cálculo matemático	Linguagem interpretada
LISP	Inteligência artificial	Linguagem intermediária
Pascal	Ensino	Linguagem compilada
PHP	Desenvolvimento de sites web dinâmicos	Linguagem interpretada
Prolog	Inteligência artificial	Linguagem interpretada
Perl	Processamento de cadeias de caracteres	Linguagem interpretada

## Montadores, Compiladores, Ligadores e Interpretadores

No nosso dia a dia, utilizamos programas de computadores para os mais diversos fins, seja para nos auxiliar em nosso trabalho, seja para usos pessoais.

Um **programa de computador** é um **conjunto de instruções** que:

- Possui um **determinado fim**;
- Por exemplo, o Photoshop para edição de imagens; e
- É **executado** por um **processador**.

Mas para haja essa execução pelo processador, o programa precisa estar em uma linguagem que o processador possa entender: a linguagem de máquina. Ela é a linguagem que um processador é capaz de compreender e é composta de apenas de números 0 e 1.

Para executar uma tarefa qualquer, um computador precisa receber instruções precisas sobre o que fazer. Uma seqüência adequada de instruções de computador, para a realização de uma determinada tarefa, se constitui num PROGRAMA de computador. Uma linguagem de programação é um conjunto de ferramentas, regras de sintaxe e símbolos ou códigos que nos permitem escrever programas de computador, destinados a instruir o computador para a realização de suas tarefas.

A primeira e mais primitiva linguagem de computador é a própria linguagem de máquina, aquela que o computador entende diretamente e pode ser diretamente executada pelos circuitos do processador (pelo hardware). No início da era da computação, os programas eram escritos em linguagem de máquina, isto é, as instruções eram escritas diretamente na linguagem do computador (formada apenas com 1's e 0's). Um programa em linguagem de máquina é uma longa série de 0's e 1's, ordenados de forma que alguns representam códigos de instruções e outros representam os dados que serão processados (ou indicam onde esses dados estão armazenados). Em um programa escrito em linguagem de máquina, cada instrução escrita pelo programador será individualmente executada, isto é, a cada instrução do programa corresponderá uma ação do computador

. A relação é portanto 1 para 1 – uma instrução do programa corresponde a uma operação do computador.

Imagine então um programa extenso escrito apenas usando 1's e 0's; imagine que para cada diferente marca ou modelo de computador, as regras para entender esses códigos serão totalmente diferentes; e finalmente imagine que voce teria que escrever uma a uma as instruções e os dados adequadamente codificados e ordenados, perfurar todos o programa em cartões e submeter toda a massa de cartões ao computador, para finalmente receber algumas horas depois o seu programa de volta com uma mensagem de erro tipo "erro no cartão X" ... e mais nada! Um programa escrito nessa linguagem era difícil de ser escrito sem que se cometessem muitos erros, processo esse longo, difícil, entediante e principalmente caro.

Um programa em linguagem de máquina era também extremamente difícil de ser entendido por outros programadores que futuramente viessem a trabalhar na manutenção do programa. Essa complexidade levou à necessidade de se desenvolverem técnicas e ferramentas para tornar a escrita e manutenção de programas mais fácil, mais rápida e principalmente mais barata.

Cada família de computadores possui sua própria linguagem de máquina. Um programa em linguagem de máquina é dependente do computador ou seja, tendo sido escrito para um determinado computador, somente poderá ser executado em computadores da mesma família, que lhe sejam 100% compatíveis.

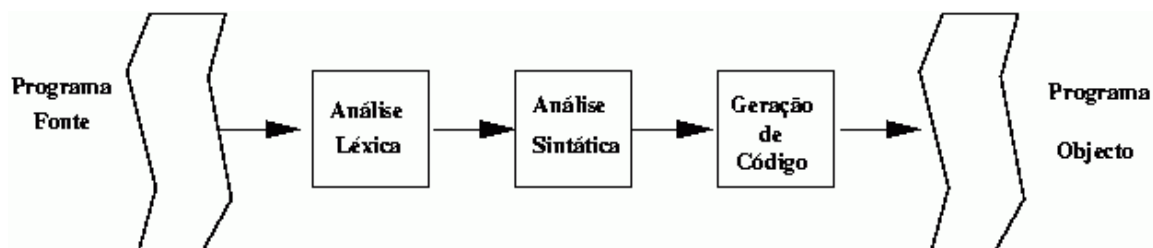
## Tradução

Um programa escrito por um programador (chamado código fonte) em uma linguagem de alto nível é um conjunto de instruções que é clara para programadores, mas não para computadores. Ou seja, os computadores entendem única e exclusivamente suas linguagens nativas, as linguagens de máquina.

Programas em linguagem de alto nível, a exemplo dos programas escritos em linguagem de Montagem, também precisam ser traduzidos para linguagem de máquina para poderem ser submetidos ao computador e processados.

O processo de tradução do programa escrito em uma linguagem simbólica pelo programador, chamado código fonte (source code) para a linguagem de máquina do computador chamada código objeto (object code), é chamado compilação e é realizado por um programa chamado Compilador (Compiler).

Tradução sucessiva de uma linguagem para outra intermediária mais simples, da linguagem fonte até à linguagem objeto.



Cada tradução pode dividir-se nas seguintes fases:

#### **Análise Léxica (scanner)**

Ele separa a sequência de caracteres que representa o programa fonte em entidades ou tokens, símbolos básicos da linguagem. Durante a análise léxica, os tokens são classificados como palavras reservadas, identificadores, símbolos especiais, constantes de tipos básicos (inteiro real, literal, etc.), entre outras categorias. Basicamente é reconhecer as sequências de símbolos que representam uma unidade. Ex.: o nome de uma variável, uma constante, uma palavra chave de uma instrução (while).

Um token consiste de um par ordenado (valor, classe). A classe indica a natureza da informação contida em valor.

Outras funções atribuídas ao analisador léxico são: ignorar espaços em branco e comentários, e detectar erros léxicos.

#### **Análise Sintática (parser)**

Ele agrupa os tokens fornecidos pelo analisador léxico em estruturas sintáticas, construindo a árvore sintática correspondente. Para isso, utiliza uma série de regras de sintaxe, que constituem a gramática da linguagem fonte. É a gramática da linguagem que define a estrutura sintática do programa fonte.

O analisador sintático tem também por tarefa o reconhecimento de erros sintáticos, que são construções do programa fonte que não estão de acordo com as regras de formação de estruturas sintáticas como especificado pela gramática.

Identifica a estrutura gramatical do programa e reconhece o papel de cada componente. É normalmente construída uma árvore sintática do programa e uma tabela de símbolos, que identifica variáveis.

#### **Analisador Semântico**

O compilador executa ainda a análise semântica. O analisador semântico utiliza a árvore sintática determinada pelo analisador sintático para: identificar operadores e operandos das expressões, reconhecer erros semânticos, fazer verificações de compatibilidade de tipo, analisar o escopo das variáveis, fazer verificações de correspondência entre parâmetros atuais e formais.

Fundamentalmente, a análise semântica trata os aspectos sensíveis ao contexto da sintaxe das linguagens de programação. Por exemplo, não é possível representar em uma gramática livre de contexto uma regra como “Todo identificador deve ser declarado antes de ser usado.”, e a verificação de que essa regra foi aplicada cabe à análise semântica.

### Otimização de Código

O processo de otimização de código consiste em melhorar o código intermediário de tal forma que o programa objeto resultante seja mais rápido em tempo de execução. Por exemplo, um algoritmo para geração do código intermediário gera uma instrução para cada operador na árvore sintática, mesmo que exista uma maneira mais otimizada de realizar o mesmo comando.

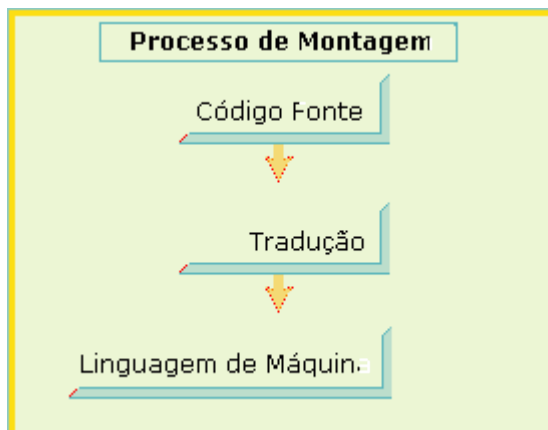
### Geração de Código

A fase final do compilador é a geração do código para o programa objeto, consistindo normalmente de código em linguagem assembly ou de código em linguagem de máquina. Aqui é o processo de construir instruções da linguagem máquina (em assembly, normalmente) que simulam as instruções reconhecidas pelo analisador sintático. A geração de código pode englobar: análise semântica, geração de código intermédio, otimizadores e geração de código final.

### Montagem

Citamos anteriormente uma forma de tradução rápida e simples: a executada pelo programa Montador. O processo de montagem traduz um programa escrito em linguagem Assembly em um programa equivalente em linguagem de máquina, possível de ser executado pelo computador.

A seguir, é apresentado o fluxo que representa o processo de montagem.



No processo de montagem, o código fonte (programa em linguagem simbólica escrito pelo programador) é examinado, instrução por instrução e é feita a tradução, gerando o código que será executado (código objeto). Os passos executados pelo programa Montador são:

1. Verificar a correção do código de instrução (se o mnemônico corresponde a uma instrução válida para o computador, se os campos definidos na estrutura da linguagem e a sintaxe estão corretos) e substituir os mnemônicos pelos códigos numéricos binários equivalentes. Qualquer erro no código acarreta a interrupção do processo e a emissão de mensagem de erro.
2. Resolver as referências de memória: os nomes simbólicos adotados pelo programador são convertidos para endereços reais de memória (valores numéricos binários de endereços).
3. Reservar espaço em memória para o armazenamento das instruções e dados.
4. Converter valores de constantes em binário.

## Compilação

Compilação é o processo de tradução de um programa escrito em linguagem de alto nível para código em linguagem de máquina. Compilação é um processo análogo ao da montagem (verificação / análise do código fonte, resolução das referências de memória, reserva de espaço em memória e conversão para código de máquina binário). O que diferencia a compilação do processo de montagem é sua maior complexidade. No processo de montagem, há uma relação de 1:1, ou seja, cada instrução do código fonte resulta em uma instrução de máquina, enquanto na compilação a relação é múltipla, cada instrução do código fonte gerando várias instruções de máquina.



Durante a compilação, o código fonte é analisado (análise léxica, sintática e semântica), é gerado um código intermediário e são construídas tabelas de símbolos, alocam-se as áreas de memória para variáveis e atribui-se os registradores a serem utilizados, e é finalmente gerado o código objeto em linguagem binária de máquina. Em alguns compiladores, é gerado um código intermediário em Assembly (que pode ser visualizado pelo programador) e que em seguida passa pelo montador para gerar finalmente o código objeto em linguagem de máquina.

O código objeto pode ser absoluto (os endereços constantes são endereços reais de memória) ou relocável (os endereços são relativos, tendo como referência o início do programa, e os endereços reais de memória são definidos apenas em tempo de execução).

### Tipos de Compiladores:

- **Single-Pass:** compilação numa única leitura do programa fonte
- **Multi-Pass:** compilação através de várias leituras do programa fonte
- **Load-And-Go:** compilação e a execução do programa fonte
- **Debugging:** compilação permitindo a depuração do programa fonte
- **Optimizing:** compilação e a otimização do programa alvo

### Vantagens:

- O código compilado é mais rápido de ser acessado;
- Impossibilita ou pelo menos dificulta ser quebrado e visualizado o código-fonte original;
- Permite otimização do código por parte do compilador;
- Compila o código somente se estiver sem algum erro.

### Desvantagens:

- Para ser utilizado o código precisa passar por muitos níveis de compilação;
- Assim como vantagem a possibilidade de não poder visualizar o código-fonte, pode ser uma desvantagem;
- Processo de correção ou alteração do código requer que ele seja novamente recompilado.

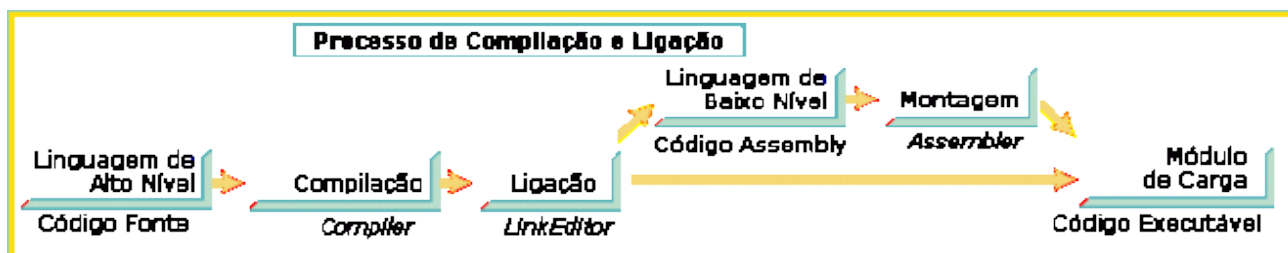
### Bibliotecas

O desenvolvimento de um programa certamente utilizará diversas operações que são comuns a muitos outros programas. Por exemplo, a execução de uma instrução de entrada e saída, a classificação dos dados de um arquivo, o cálculo de funções matemáticas, etc. Uma linguagem de alto nível geralmente incorpora diversas rotinas prontas (que fazem parte da linguagem) e que compõem bibliotecas (libraries) de funções pré-programadas que poderão ser utilizadas pelo programador, poupando tempo, aumentando a eficiência e evitando erros. Dessa forma, um programa em alto nível possivelmente conterá diversas chamadas de biblioteca (library calls). Essas funções não devem ser confundidas com as instruções da linguagem – na realidade, são pequenos programas externos que são chamados através de instruções especiais de chamada de biblioteca. Para serem executadas, essas rotinas precisam ser incorporadas ao código do programador, isto é, a chamada de biblioteca precisa ser substituída pelo código do programador, incluindo os parâmetros necessários.

### Ligação

Assim, o código objeto preparado pelo compilador em geral não é imediatamente executável, pois ainda existe código (as rotinas de biblioteca) a ser incorporado ao programa. A cada chamada de biblioteca encontrada no código fonte, o compilador precisará incluir uma chamada para a rotina e o endereço dos dados que devam ser passados para a rotina.

A tarefa de examinar o código objeto, procurar as referências a rotinas de biblioteca (que constituem referências externas não resolvidas), buscar a rotina da biblioteca, substituir a chamada pelo código (“resolver as referências externas”) e obter os parâmetros para incluí-los no código objeto é executada por um programa chamado Ligador (LinkEditor). O resultado da execução do Ligador é o código final pronto para ser executado pelo computador, chamado módulo de carga ou código executável.

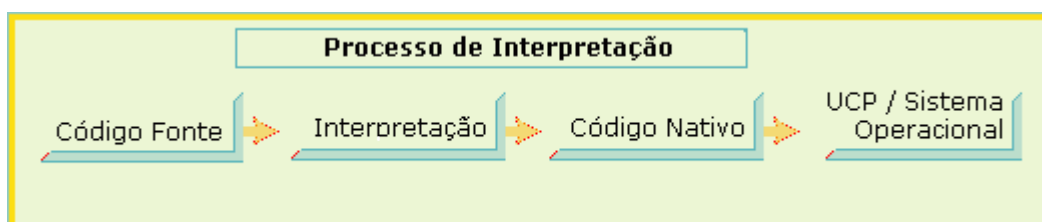


O módulo de carga, após testado e depurado (isto é, depois de resolvidos todos os erros, também chamados “bugs”) é armazenado em memória de massa para ser executado quando necessário. O processo de compilação e ligação é executado apenas pelo programador na fase de desenvolvimento e não mais precisará ser executado pelo usuário, quando da execução do programa.

### Interpretação

Com o processo de execução de um programa em fases distintas (compilação / ligação / execução) apresentado, um programa para ser executado precisa primeiro ter sido convertido para código objeto pelo compilador e depois ter passado pelo ligador. Esse processo é o mais largamente utilizado, porém não é o único.

O método alternativo chama-se de interpretação e, a partir do programa fonte, realiza as três fases (compilação, ligação e execução), comando por comando, em tempo de execução. Não existem fases distintas nem se produzem códigos intermediários. Todo o processo de conversão é efetuado em tempo de execução e imediatamente executado. Ou seja, cada comando é lido, verificado, convertido em código executável e imediatamente executado, antes que o comando seguinte seja sequer lido.





Linguagens como C, Pascal, COBOL, etc, são linguagens tipicamente compiladas, enquanto o BASIC foi desenvolvido como linguagem interpretada (hoje também existem linguagens BASIC compiladas e o programador pode optar). As linguagens de programação tipicamente de usuário, tais como das planilhas Excel, o Word Basic (linguagem de construção de Macros do Word), o Access, etc, são todas linguagens interpretadas.

Uma vantagem é que o ciclo de escrita, execução, modificação é mais rápido em relação à compilação. Por outro lado, a execução é mais lenta.

No programa em linguagem de alto nível, os interpretadores executam os passos definidos para cada instrução e produzem o mesmo resultado que o do programa compilado. Entretanto, a execução de um programa em linguagem de alto nível com o uso de interpretadores é mais lenta que a execução de um programa compilado, uma vez que precisa examinar cada instrução no programa-fonte, à medida que ela ocorre, e desviar para a rotina que executa a instrução.

**Vantagens:**

- Correções e alterações são mais rápidas de serem realizadas;
- Código não precisa ser compilado para ser executado;
- Consomem menos memória.

**Desvantagens:**

- Execução é mais lenta do programa;
- Necessita sempre ser lido o código original para ser executado;

Veja o comparativo abaixo:

	<b>Vantagens</b>	<b>Desvantagens</b>
Compiladores	Execução mais rápida	Várias etapas de tradução
	Permite estruturas de programação mais completas para a sua execução	Programação final é maior necessitando mais memória
	Permite a otimização do código fonte	Processo de correção de erros e depuração é mais demorado
Interpretadores	Depuração do programa é mais simples	Execução do programa é mais lenta
	Consome menos memória	Estruturas de dados demasiado simples
	Resultado imediato do programa ou rotina desenvolvida	Necessário fornecer o programa fonte ao utilizador

**Bytecode**

Traduzindo ao pé da letra, código em bytes — não confundir com código-máquina, é o resultado de um processo semelhante ao dos compiladores de código-fonte que não é imediatamente executável. Em oposição, o bytecode irá ser interpretado numa máquina virtual, que fará a execução. Assim, o bytecode é um estágio intermédio entre o código-fonte (escrito numa linguagem de programação específica) e a aplicação final, sendo a sua vantagem principal a dualidade entre a portabilidade — o bytecode irá produzir o mesmo resultado em qualquer arquitetura — e a ausência da necessidade do pré-processamento típico dos compiladores — o bytecode é encarado como um produto final, cuja validação da sintaxe e tipos de dados (entre outras funções dos compiladores) não será necessária.

**Compilador:**

Compilador é um programa ou um grupo de programas que escrito por uma linguagem (esta sendo necessária de compilação para sua execução) ao ser compilado gera outro código que é interpretado pelo computador. Este código compilado é chamado de código objeto, podendo ser um arquivo executável que é reproduzido em um sistema operacional, por exemplo. Este tipo de tradutor é um dos mais utilizados.

Os compiladores analisam o código em três partes, de forma sintática ou hierárquica, análise léxica ou linear e análise semântica.

Alguns compiladores contam com um pré-processamento. Este pré-processamento é responsável por modificar o código-fonte conforme necessidades que o compilador identifique que sejam necessárias, por exemplo, otimização de código.

**Vantagens:**

O código compilado é mais rápido de ser acessado;

- Impossibilita ou pelo menos dificulta ser quebrado e visualizado o código-fonte original;
- Permite otimização do código por parte do compilador;
- Compila o código somente se estiver sem algum erro.

**Desvantagens:**

Para ser utilizado o código precisa passar por muitos níveis de compilação;

- Assim como vantagem a possibilidade de não poder visualizar o código-fonte, pode ser uma desvantagem;
- Processo de correção ou alteração do código requer que ele seja novamente recompilado.

**Interpretador:**

O interpretador ao contrário do compilador roda o código-fonte escrito como sendo o código objeto, ele traduz o programa linha a linha, o programa vai sendo utilizado na medida em que vai sendo traduzido. Cada execução do programa precisa ser novamente traduzido e interpretado.

O interpretador analisa sintaticamente e semanticamente o código, se estas duas etapas forem realizadas e executadas de forma correta o código está pronto para funcionar.

**Vantagens:**

- Correções e alterações são mais rápidas de serem realizadas;
- Código não precisa ser compilado para ser executado;
- Consomem menos memória.

**Desvantagens:**

- Execução é mais lenta do programa;
- Necessita sempre ser lido o código original para ser executado;

s processadores CISC (Complex Instruction Set Computers) já possuem instruções complexas que são interpretadas por micro-programas. O número de registradores é pequeno e qualquer instrução pode referenciar a memória principal. Neste tipo de arquitetura, a implementação do pipeline é mais difícil. São exemplos de processadores CISC o VAX (DEC), 80x86 e o Pentium (Intel), e o 68xx (Motorola).

**Curso de Técnico em Informática – ETE Taquarituba****Software**

O Hardware por si só não tem a menor utilidade. Para torná-lo útil existe um conjunto de programas, utilizado como interface entre as necessidades do usuário e as capacidades do hardware. A utilização de softwares adequados às diversas tarefas e aplicações (conceitos de camadas) torna o trabalho do usuários muito mais simples e eficiente.

**Tradutor**

Nos sistemas operacionais antigos, o ato de programar era bastante complicado, já que o programador deveria possuir conhecimento do hardware e programar em painéis através de fios. Esses programas eram desenvolvidos em linguagem de máquina e carregados diretamente na memória principal para execução.

Com o surgimento das primeiras linguagens de montagem (assembly languages) e das linguagens de alto nível, o programador deixou de se preocupar com muitos aspectos pertinentes ao hardware, como em qual região da memória o programa deveria ser carregado ou quais endereços de memória seriam reservados para as variáveis. A utilização dessas linguagens facilitou a construção de programas em muitos aspectos. Desse modo, um programa poderia ser escrito de uma forma bem documentada e com facilidades para realizar alterações.

O tradutor, pelo tipo de linguagem de programação utilizada, pode ser chamado de montador ou compilador (Figura 6).

Programa-Fonte Programa-Fonte Programa-Objeto

Linguagem de Montagem Módulo-Objeto

Módulo-Objeto

Montador

Compilador

Linguagem de Alto Nível

Figura 6 - Tradutor

**Compilador**

É o utilitário responsável por gerar, a partir de um programa escrito em uma linguagem de alto nível, um programa em linguagem de máquina não executável.

As linguagens de alto nível, como pascal, fortran, cobol não tem nenhuma relação direta com a máquina, ficando essa preocupação exclusivamente com o compilador.

**Interpretador**

O interpretador é considerado um tradutor que não gera código-objeto. A partir de um programa fonte, escrito em linguagem de alto nível, o interpretador, no momento da execução do programa, traduz cada instrução e a executa em seguida.

**Linker**

O linker (ligador), também chamado de linkagem, é o utilitário responsável por gerar, a partir de um ou mais módulos-objetos, um único programa executável.

Curso de Técnico em Informática – ETE Taquarituba

Módulo

Objeto Compilador

Módulo Fonte

Módulo Objeto

Módulo Fonte LinkerCompilador

Módulo

Objeto Compilador

Módulo Fonte

Programa Executável

Figura 7 - Linker.

### **Loader**

Também chamado carregador é o utilitário responsável por colocar fisicamente na memória um programa para execução. O procedimento de carga varia com o código gerado pelo linker e, em função deste, o loader é classificado como sendo do tipo absoluto ou relocável.

Tipo absoluto - o loader só necessita conhecer o endereço de memória inicial e o tamanho do módulo para realizar o carregamento. Então, ele transfere o programa da memória secundária para a memória principal e inicia sua execução.

No caso de código relocável, o programa pode ser carregado em qualquer posição de memória, e o loader é responsável pela relocação no momento do carregamento.

### **Depurador**

O desenvolvimento de programas está sujeito a erros de lógica, independentemente de metodologias utilizadas pelo programador. A depuração é um dos estágios desse desenvolvimento, e a utilização de ferramentas adequadas é essencial para acelerar o processo de correção de programas.

O depurador (debugger) é o utilitário que permite ao usuário controlar a execução de um programa a fim de detectar erros na sua estrutura. Este utilitário oferece ao usuário recursos como:

- Acompanhar a execução de um programa instrução por instrução;

- Possibilitar a alteração e visualização do conteúdo de variáveis;
- Implementar pontos de parada dentro do programa (break-point), de forma que, durante a execução, o programa pare nesses pontos;
- Especificar que, toda vez que o conteúdo de uma variável for modificado, o programa envie uma mensagem (watchpoint).

### **Linguagem de Controle**

Também denominada a linguagem de comando, e a forma mais direta de um usuário se comunicar com o sistema operacional. Esta linguagem é oferecida por cada sistema operacional para que, através de comandos simples, o usuário possa ter acesso a rotinas específicas do sistema.

Curso de Técnico em Informática – ETE Taquarituba

### **Interpretador de Comandos (Shell)**

O sistema operacional é o código executor de chamadas de sistema. Os editores, compiladores, montadores, ligadores e interpretadores de comando não fazem parte do sistema operacional, apesar de serem softwares muito importantes e muito úteis. Esses comandos quando digitados pelos usuários, são interpretados pelo Shell, verifica sua sintaxe, envia mensagens de erro e faz chamadas a

rotinas do sistema. Dessa forma o usuário dispõe de uma interface interativa com o sistema operacional, para realizar tarefas como acessar um arquivo em disco ou consultar um diretório.

### **Linguagem de Máquina**

A linguagem de máquina de um computador é a linguagem de programação que o processador realmente consegue entender. Cada processador possui um conjunto único de instruções de máquina, definido pelo próprio fabricante. As instruções especificam detalhes, como registradores, modos de endereçamento e tipos de dados, que caracterizam um processador e suas potencialidades.

### **Microprogramação**

Um programa em linguagem de máquina é executado diretamente pelo hardware em processadores de arquitetura RISC, porém em máquinas CISC isto não acontece. Neste caso, como podemos observar na Figura 3, entre os níveis de linguagem de máquina e do hardware, existem ainda o da microprogramação.

Os microprogramas definem a linguagem de máquina de cada computador. Apesar de cada computador possui níveis de microprogramação diferentes, existem muitas semelhanças nessa camada se compararmos os diversos equipamentos. Uma máquina possui, aproximadamente 25 microinstruções básicas, que são interpretadas pelos circuitos eletrônicos.

### **Processos**

Um conceito chave da teoria dos sistemas operacionais é o conceito de processo. Um processo é basicamente um programa em execução, sendo constituído do código executável, dos dados referentes ao código.

### **Chamadas de Sistema**

Os programas de usuário solicitam serviços do sistema operacional através da execução de chamadas de sistema. A cada chamada corresponde um procedimento de uma biblioteca de procedimentos que o programa do usuário pode chamar.

### **Arquivos**

Arquivos são mecanismos de abstração que fornece uma forma de armazenar recuperar informações em disco. Isto deve ser feito de uma forma que mantenha o usuário isolado dos detalhes a respeito de como as informações são armazenadas, e de como os discos efetivamente trabalha.

Curso de Técnico em Informática – ETE Taquarituba

### **Tipos de Sistemas Operacionais**

Os tipos de sistemas operacionais e sua evolução estão intimamente relacionados com a evolução do hardware e das aplicações por ele suportadas.

Muitos termos inicialmente introduzidos para definir conceitos e técnicas acabam sendo substituídos por outros, na tentativa de refletir uma nova maneira de interação ou processamento. Isto fica muito claro quando tratamos da unidade de execução do processador. Inicialmente, os termos programa ou job eram os mais utilizados, depois surgiu o conceito de processo e sub-processo e, mais recentemente, os conceitos de tarefa e de thread.

A evolução dos sistemas operacionais para computadores pessoais e estações de trabalho popularizou vários conceitos e técnicas, antes só conhecidos em ambientes de grande porte. A nomenclatura, no entanto, não se manteve a mesma. Surgiram novos termos para conceitos já conhecidos, que foram apenas adaptados para uma nova realidade.

Tipos de Sistemas Operacionais

Sistemas

Monoprogramáveis/ Monotarefa

Sistemas

Multiprogramáveis/ Multitarefa

Sistemas com

Múltiplos Processadores

Figura 8 - Tipos de sistemas operacionais

Tipos de Sistemas Operacionais Sistemas Monoprogramáveis/Mono-tarefa Sistemas Multiprogramáveis/Multitarefa Sistemas Batch

Sistemas de Tempo Compartilhado Sistemas de Tempo Real

Sistemas com Múltiplos Processadores

Sistemas Fortemente Acoplados Sistemas Simétricos Sistemas Assimétricos Sistemas Fracamente Acoplados Sistemas Operacionais de Rede Sistemas Operacionais Distribuídos

### **Sistemas Monoprogramáveis/Monotarefa**

Os primeiros sistemas operacionais eram tipicamente voltados para a execução de um único programa (job). Qualquer outro programa, para ser executado, deveria aguardar o término do programa corrente. Os sistemas monoprogramáveis, como vieram a ser conhecidos, se caracterizam por permitir que o processador, a memória e os periféricos permaneçam exclusivamente dedicados à execução de um único programa.

#### **Curso de Técnico em Informática – ETE Taquarituba**

Neste tipo de sistema, enquanto um programa aguarda por um evento, como a digitação de um dado, o processador permanece ocioso, sem realizar qualquer tipo de processamento. A memória é subutilizada caso o programa não a preencha totalmente, e os periféricos, como discos e impressoras, estão dedicados a um único usuário.

Comparados a outros sistemas, os sistemas monoprogramáveis/monotarefa são de simples implementação, não existindo muita preocupação com problemas de proteção.

UCP Memória

Dispositivos de E/S

Programa/ Tarefa

Figura 9 - Sistemas monoprogramáveis/monotarefa

### **Sistemas Multiprogramáveis/Multitarefa**

Os Sistemas Multiprogramáveis, que vieram a substituir os monoprogramáveis, são mais complexos e eficientes. Enquanto em sistemas monoprogramáveis existe apenas um programa utilizando seus diversos recursos, nos multiprogramáveis vários programas dividem esses mesmos recursos.

As vantagens do uso de sistemas multiprogramáveis são o aumento da produtividade dos seus usuários e a redução de custos, a partir do compartilhamento dos diversos recursos do sistema.

A partir do número de usuários que interagem com o sistema, podemos classificar os sistemas multiprogramáveis como monousuário e multiusuário.

O conceito de sistemas multiprogramável está tipicamente associado aos mainframes e minicomputadores, onde existe a idéia do sistema sendo utilizado por vários usuários (multiusuário). No mundo dos computadores pessoais e estações de trabalho, apesar de existir apenas um único usuário interagindo como sistema (monousuário), é possível que ele execute diversas tarefas concorrentemente ou mesmo simultaneamente. Os sistemas multitarefa, como também são chamados, se caracterizam por

permitir que o usuário edite um texto, imprima um arquivo, copie um arquivo pela rede e calcule uma planilha. Abaixo estão relacionados os tipos de sistemas em função do número de usuários

Um usuário Dois ou mais usuários

Monoprogramação/Monotarefa Monousuário N/A Multiprogramação/Multitarefa Monousuário Multiusuário

Tabela 2 - Sistemas X Usuários

Os sistemas multiprogramáveis/multitarefa podem ser classificados pela forma com que suas aplicações são gerenciadas, podendo ser divididos em sistemas batch, de tempo compartilhado ou de tempo real. Um sistema operacional pode suportar um ou mais desses tipos de processamento.

Curso de Técnico em Informática – ETE Taquarituba

Sistemas Multiprogramáveis/Multitarefa

Sistemas

Batch

Sistemas de Tempo compartilhado

Sistemas de Tempo Real

Figura 10 - Tipos de sistemas multiprogramáveis/multitarefa

### **Sistemas Batch**

Os sistemas batch (lote) foram os primeiros sistemas multiprogramáveis a serem implementados e caracterizam-se por terem seus programas, quando submetidos, armazenados em disco ou fita, onde esperam para ser executados seqüencialmente.

Normalmente, os programas, também chamados de jobs, não exigem interação com os usuários, lendo e gravando dados em discos e fitas. Alguns exemplos de aplicações originalmente processadas em batch são compilações, linkedições, sorts, backups e todas aquelas onde não é necessária a interação com o usuário.

### **Sistemas de Tempo Compartilhado**

Os sistemas de tempo compartilhado (time-sharing) permitem a interação dos usuários com o sistema, basicamente através de terminais que incluem vídeo, teclado e mouse. Dessa forma, o usuário pode interagir diretamente com o sistema em cada fase do desenvolvimento de suas aplicações e, se preciso, modificá-las imediatamente. Devido a esse tipo de interação, os sistemas de tempo compartilhado também ficaram conhecidos como sistemas on-line.

Para cada usuário, o sistema operacional aloca uma fatia de tempo (time-slice) do processador. Caso o programa do usuário não esteja concluído nesse intervalo de tempo, ele é substituído por um de outro usuário, e fica esperando por uma nova fatia de tempo. Não só o processador é compartilhado nesse sistema, mas também a memória e os periféricos, como discos e impressoras. O sistema cria para o usuário um ambiente de trabalho próprio, dando a impressão de que todo o sistema está dedicado, exclusivamente, a ele.

Sistemas de tempo compartilhado são de implementação complexa, porém, se levado em consideração o tempo de desenvolvimento e depuração de uma aplicação, aumentam consideravelmente a produtividade dos seus usuários, reduzindo os custos de utilização do sistema.

### **Sistemas de Tempo Real**

Os sistemas de tempo real (real time) são bem semelhantes em implementação aos sistemas de tempo compartilhado. A maior diferença é o tempo de resposta exigido no processamento das aplicações.

Enquanto em sistemas de tempo compartilhado o tempo de resposta pode variar sem comprometer as aplicações em execução, nos sistemas de tempo real os tempos de resposta devem estar dentro de limites rígidos, que devem ser obedecidos, caso contrário poderão ocorrer problemas irreparáveis.

Não existe idéia de fatia de tempo, um programa detém o processador o tempo que for necessário, ou até que apareça outro prioritário em função de sua importância no sistema. Esta importância ou prioridade de execução é controlada pela própria aplicação e não pelo sistema operacional, como nos sistemas de tempo compartilhado.

Esses sistemas, normalmente, estão presentes em controle de processos, como no monitoramento de refinarias de petróleo, controle de tráfego aéreo, de usinas termelétricas e nucleares, ou em qualquer aplicação onde o tempo de resposta é fator fundamental.

Curso de Técnico em Informática – ETE Taquarituba

### **Sistemas com Múltiplos Processadores**

Os sistemas com múltiplos processadores caracterizam-se por possuir duas ou mais

UCPS interligadas, trabalhando em conjunto. Um fator-chave no desenvolvimento de sistemas operacionais com múltiplos processadores é a forma de comunicação entre as UCPs e o grau de compartilhamento da memória e dos dispositivos de entrada e saída. Em função desses fatores, podemos classificar os sistemas em fortemente acoplados ou fracamente acoplados.

Sistemas com Múltiplos Processadores

Sistemas Fracamente

Acoplados

Sistemas Fortemente Acoplados

Sistemas

Operacionais Distribuídos

Sistemas

Operacionais de Rede

Sistemas Simétricos

Sistemas Assimétricos

Figura 1 - Sistemas com múltiplos processadores.

### **Sistemas Fortemente Acoplados**

Nos sistemas fortemente acoplados (tightly coupled) existem vários processadores compartilhando uma única memória e gerenciados por apenas um sistema operacional. Múltiplos processadores permitem que vários programas sejam executados ao mesmo tempo, ou que um programa seja dividido em subprogramas, para execução simultânea em mais de um processador. Dessa forma, é possível ampliar a capacidade de computação de um sistema, adicionando-se apenas novos processadores, com um custo muito inferior à aquisição de outros computadores.

Com o multiprocessamento, novos problemas de concorrência foram introduzidos, pois vários processadores podem estar acessando as mesmas áreas de memória. Além disso, existe o problema de organizar de forma eficiente os processadores, a memória e os periféricos.

Uma consequência do multiprocessamento foi o surgimento dos computadores voltados, principalmente, para processamento científico, aplicado, por exemplo, ao desenvolvimento aeroespacial, prospecção de petróleo, simulações, processamento de imagens e CAD. A princípio qualquer aplicação que faça uso intensivo da UCP será beneficiada pelo acréscimo de processadores ao sistema.



Curso de Técnico em Informática – ETE Taquarituba

UCP Memória

Dispositivos de E/S

Dispositivos de E/S

Figura 12 - Sistemas fortemente acoplados

Memória Dispositivos de E/S

Memória Dispositivos de E/S

Link de Comunicação

Figura 13 - Sistemas fracamente acoplados

### **Sistemas Assimétricos**

Na organização assimétrica ou mestre/escravo(master/slave), somente um processador (mestre) pode executar serviços do sistema operacional, como, por exemplo, realizar operações de entrada/saída. Sempre que um processador do tipo escravo precisar realizar uma operação de entrada/saída, terá de requisitar o serviço ao processador mestre. Dependendo do volume de operações de entrada/saída destinadas aos processadores escravos, o sistema pode se tornar ineficiente, devido ao elevado número de interrupções que deverão ser tratadas pelo mestre.

UCP SlaveUCP Master

Dispositivos de E/S S.O Usuários Usuários

Figura 14 - Sistemas assimétricos.

Curso de Técnico em Informática – ETE Taquarituba

Se o processador falhar, todo o sistema ficará incapaz de continuar o processamento.

Neste caso, o sistema deve ser reconfigurado, fazendo um dos processadores escravos assumir o papel do mestre.

Mesmo sendo uma organização simples de implementar e quase um extensão dos sistemas multiprogramáveis, esse tipo de sistema não utiliza eficientemente o hardware, devido à assimetria dos processadores, que não realizam as mesmas funções.

### **Sistemas Simétricos**

O multiprocessamento simétrico (Symmetric Multiprocessing- SMP), ao contrário da organização mestre/escravo, implementa a simetria dos processadores, ou seja, todos os processadores realizam as mesmas funções. Apenas algumas poucas funções ficam a cargo de um único processador, como, por exemplo, a inicialização (boot) do sistema.

Dispositivos de E/S.OUsuários

Figura 15 - Sistemas simétricos.

Como vários processadores estão utilizando, independentemente, a mesma memória e o mesmo sistema operacional, é natural a ocorrência de acessos simultâneos às mesmas áreas de memória. A solução desses conflitos fica a cargo do hardware e do sistema operacional.

No processamento simétrico, um programa pode ser executado por qualquer processador, inclusive por vários processadores ao mesmo tempo (paralelismo). Além disso, quando um processador falha, o sistema continua em funcionamento sem nenhuma interferência manual, porém com menor capacidade de computação.

Os sistemas simétricos são mais poderosos que os assimétricos, permitindo um melhor balanceamento do processamento e das operações de entrada/saída, apesar de sua implementação ser bastante complexa.

### **Multiprocessamento**

Desde sua criação, os computadores têm sido vistos como máquinas seqüências, onde a UCP executa a instruções de um programa, uma de cada vez. Na realidade, essa visão não é totalmente verdadeira, pois, em nível de hardware, múltiplos sinais estão ativos simultaneamente, o que pode ser entendido como uma forma de paralelismo.

Com a implementação de sistemas com múltiplos processadores, o conceito de simultaneidade ou paralelismo pode ser expandido a um nível mais amplo, denominado multiprocessamento, onde uma tarefa pode ser dividida e executada, ao mesmo tempo, por mais de um processador.

### **Organização Funcional**

O esquema de comunicação interna das UCPs, memória e dispositivos de E/S (unidades funcionais) é fundamental no projeto de sistemas com múltiplos processadores, pois termina quantas UCPs o sistema poderá ter e como será o acesso à memória.

Para permitir múltiplos acessos simultâneos à memória (interliving), é comum que esta dividida em módulos, podendo assim ser compartilhada por várias unidades funcionais. As

Curso de Técnico em Informática – ETE Taquarituba organizações funcionais de multiprocessadores podem ser divididas basicamente em três tipos: barramento comum, barramento cruzado e memória multiport.

### **Sistemas Fracamente Acoplados**

Os sistemas fracamente acoplados caracterizam-se por possuir dois ou mais sistemas de computação interligados, sendo que cada sistema possui o seu próprio sistema operacional, gerenciando os seus recursos, como processador, memória e dispositivos de entrada/saída.

Até meados da década de 80, os sistemas operacionais e as aplicações suportadas por eles eram tipicamente concentradas em sistemas de grande porte, com um ou mais processadores. Nos sistemas centralizados, os usuários utilizam terminais não inteligentes conectados a linhas seriais dedicadas ou linhas telefônicas públicas para a comunicação interativa com esses sistemas.

No modelo centralizado, os terminais não têm capacidade de processamento. Sempre um usuário deseja alguma tarefa, o pedido é encaminhado ao sistema, que realiza o processamento e retorna uma resposta, utilizando as linhas de comunicação.

Com a evolução dos computadores pessoais e das estações de trabalho, juntamente com o avanço das telecomunicações e da tecnologia de redes, surgiu um novo modelo de computação, chamado de modelo de rede de computadores.

Rede

Nó Nó Figura 16- Sistemas fracamente acoplados

### **Sistemas Operacionais de Rede**

Em sistemas operacionais de rede (SOR), cada nó possui seu próprio sistema operacional, além de um hardware e software que possibilitam ao sistema ter acesso a outros componentes da rede, compartilhando seus recursos. O SOR permite entre outras funções:

- Cópia remota de arquivos

- Emulação de terminal
- Impressão remota
- Gerência remota

- Correio eletrônico. Cada nó é totalmente independente do outro, podendo inclusive possuir sistemas operacionais diferentes. Caso a conexão entre os nós sofra qualquer problema, os sistemas podem continuar operando normalmente, apesar de alguns recursos se tornarem indisponíveis.

O melhor exemplo da utilização dos sistemas operacionais de rede são as redes locais.

Nesse ambiente, cada estação pode compartilhar seus recursos com o restante da rede. Caso uma estação sofra qualquer, os demais componentes da rede podem continuar o processamento, apenas não dispondo dos recursos oferecidos por ela.

Curso de Técnico em Informática – ETE Taquarituba

Figura 17 - Sistemas operacionais de rede.

### **Sistemas Operacionais Distribuídos**

Em sistemas distribuídos, cada componente da rede também possui seu próprio sistema operacional, memória, processador e dispositivos. O que define um sistema distribuído é a existência de um relacionamento mais forte entre os seus componentes, onde geralmente os sistemas operacionais são os mesmos. Para o usuário e suas aplicações, é como se não existisse uma rede de computadores, mas sim um único sistema centralizado.

Rede Usuário

Figura 18 - Sistemas Operacionais Distribuídos.

A grande vantagem desses sistemas é a possibilidade do balanceamento de carga, ou seja, quando um programa é admitido para execução, a carga de processamento de cada sistema é avaliada e o processador mais livre é escolhido. Depois de aceito para processamento, o programa é executado no mesmo processador até o seu término. Também é possível o compartilhamento de impressoras, discos e fitas, independentemente do sistema em que a aplicação esteja sendo processada. Este tipo de sistema distribuído é muitas vezes chamado de cluster.

Curso de Técnico em Informática – ETE Taquarituba COMP 2COMP 1

Figura 19 - Cluster.

Suponha, por exemplo, uma configuração de dois computadores (COMP 1 e COMP 2), formando um cluster. Qualquer usuário conectado ao cluster poderá ter acesso aos dispositivos compartilhados, que permitem a ele imprimir uma listagem ou copiar um arquivo. Nesse tipo de configuração, se um dos sistemas falhar, o acesso aos dispositivos não será interrompido.

Os sistemas distribuídos podem ser considerados como uma evolução dos sistemas fortemente acoplados, onde uma aplicação pode ser executada por qualquer processador. Os sistemas distribuídos permitem que uma aplicação seja dividida em diferentes partes (aplicações distribuídas), que se comunicam através de linhas de comunicação, podendo cada parte ser processada em um sistema independente.

### **Organização Funcional**

A organização funcional dos sistemas fracamente acoplados ou topologia define como são interligados fisicamente os diversos sistemas da rede.

#### **Barramento**

Na organização de barramento, os sistemas são conectados a uma única linha de comunicação e todos compartilham o mesmo meio, tanto para receber como para enviar mensagens. Esse tipo de organização é utilizada geralmente em redes locais (Figura 20).

Neste tipo de topologia, caso haja algum problema com o meio de transmissão, todos os nós da rede ficarão incomunicáveis.

Curso de Técnico em Informática – ETE Taquarituba



## Sistemas de Numeração

Os números Naturais são utilizados para resolver problemas de contagem e foram os primeiros a serem utilizados pela humanidade. Já no início da civilização, havia uma certa “correspondência binária” neste processo, pois as tribos primitivas costumavam contar as coisas Como “um, dois e muitos”, ou seja, tudo conjunto ou coleção que tinha mais do que dois elementos era considerado como muitos ou infinito.

Essa herança primitiva permanece até hoje: em francês, por exemplo, a palavra “muitos” se traduz como “très” e este radical está presente em vários idiomas modernos para representar tal número: no próprio francês, três é traduzido como trois, em inglês, three, em Italiano, tre e assim por diante. Com a evolução histórica, social e intelectual da humanidade, evoluiu também o conceito de infinito mas, embora seja matematicamente incorreto, é comum pessoas dizerem que uma quantidade enorme ou incontável é infinita.

Embora a matemática tenha um alto nível de exigência no tocante ao rigor das demonstrações lógicas, por muito tempo o conjunto  $\mathbb{N}$  foi tido apenas como intuitivo e não demonstrável, tanto que o famoso matemático Leopold Kronecker (1823 – 1891) chegou a afirmar que Deus criou os números Naturais e o homem fez todo o resto. Foi apenas em 1889 que o matemático italiano Giuseppe Peano (1858 – 1932) conseguiu caracterizar os números Naturais de forma axiomática, através dos famosos Axiomas de Peano, Posteriormente, seguiram-se outras demonstrações mais complexas e rigorosas deste conjunto.

Para entendermos os sistemas de numeração, precisamos de dois conceitos importantes no conjunto  $\mathbb{N}$ : divisibilidade e divisão euclidiana.

### Divisibilidade em $\mathbb{N}$

A divisibilidade é a versão multiplicativa da relação de ordem e é definida como: dados dois números naturais  $a$  e  $b$ , com  $b$  diferente de 0, dizemos que  $b$  divide  $a$  (ou, de forma recíproca, que  $a$  é um múltiplo de  $b$ ) e anotamos  $b|a$  se e somente se existe um outro número natural  $c$  tal que  $a = b \cdot c$ .

Por exemplo:  $2|6$  pois  $6 = 3 \cdot 2$ ;  $7|63$  pois  $63 = 7 \cdot 9$ , mas 3 não divide 8, pois não existe nenhum número natural que multiplicado por três resulte em 8.

Percebamos, aqui, que o conceito de divisibilidade nada mais é do que achar uma solução para a equação  $bx = a$ . Por este motivo, tratamos deste assunto apenas no conjunto dos números Naturais e Inteiros, pois ela poderá ou não ter uma solução; em outros conjuntos, como  $\mathbb{Q}$ ,  $\mathbb{I}$  ou  $\mathbb{R}$ , sempre existirá algum  $x$  que satisfaça a equação. Se trouxéssemos o último exemplo para  $\mathbb{Q}$ , teríamos que  $x=8/3$ . Notemos com isso,  $b$  é sempre menor ou igual a  $a$ .

### O Algoritmo de Euclides Para a Divisão

Euclides foi um famoso matemático grego que viveu em aproximadamente 300 a.C. Foi ele quem escreveu a obra Os Elementos, que definem os alicerces para a Geometria que vemos nos ensinamentos Fundamental e Médio e que também é utilizada em várias áreas, como engenharia, design e construção civil.

Fora da Geometria, Euclides descobriu um teorema o qual nos permite dividir qualquer número Natural por outro – e você certamente o conhece desde o início de sua vida escolar. Ele se baseia na ideia de divisibilidade a qual acabamos de ver e tem por objetivos responder a uma simples pergunta: Se  $a$  e  $b$  são números Naturais e  $b$  não divide  $a$ , qual será o múltiplo de  $b$  que mais se aproxima de  $a$ ? A resposta a esta pergunta está em um teorema o qual afirma que, se  $a$  e  $b$  são números Naturais, então existem e são únicos os números Naturais  $q$  e  $r$  tais que  $a = q \cdot b + r$ , sendo que  $r$  é menor do que  $b$ .

Chamamos a  $q$  de quociente e a  $r$  de resto. A condição de que o resto seja menor do que o divisor  $b$  nos garante que o quociente será único. Caso não fizéssemos essa restrição, a divisão de  $a$  por  $b$  poderia ter infinitos resultados.

Como vimos no exemplo anterior, 3 não divide 8. Assim, podemos utilizar o algoritmo da divisão euclidiana e concluir que  $8 = 2 \cdot 3 + 2$ , ou seja: 6,  $3 \cdot 2$ , é o múltiplo de 3 que mais se aproxima de 8.

Nesta expressão, perceba que  $q = 2$  e  $r = 2$  e, ainda, que o valor de  $r$  é menor do que o divisor 3. Caso não considerássemos a condição  $r < b$ , também poderíamos escrever que  $8 = 1 \cdot 3 + 5$  ou que  $8 = 0 \cdot 3 + 8$  e, se considerássemos o conjunto dos números inteiros,  $Z$ , teríamos mais infinitas possibilidades. Desta forma, garantimos a unicidade do resultado ao considerar que o resto sempre deverá ser menor do que o divisor. Perceba, ainda, que se  $a$  for múltiplo de  $b$ , o resto será igual a 0 e a divisão será dita exata.

### Sistemas de Numeração

Agora que temos toda a base matemática de que precisamos, podemos tratar dos sistemas de numeração em si. Um sistema de numeração nada mais é do que um sistema onde um conjunto de números é representado por numerais de forma consistente.

Atualmente, nós utilizamos um sistema de numeração decimal, isto é, de base 10, que é dito posicional, ou seja, cada algarismo, além de seu valor, possui um peso dado através da posição que ocupa. Desta forma, embora os números 518 e 851 sejam compostos pelos mesmos algarismos, sabemos que eles representam quantidades diferentes.

A princípio, a base do sistema de numeração pode ser um número qualquer. Os babilônios antigos usavam um sistema de base 60 cujos vestígios encontramos ainda hoje na medição de ângulos e nos relógios, por exemplo. Quando a base do sistema é menor ou igual a 10, utilizamos os algarismos indo arábicos para representar os numerais; quando é maior, devemos utilizar outros símbolos, geralmente letras do alfabeto latino.

Graças a um teorema matemático, sabemos que cada número Natural possui uma representação única em uma base qualquer. Este teorema pode ser enunciado como: "Seja  $b$  um número Natural com  $b \geq 2$ . Então, para todo  $a$  Natural diferente de 0, existem e são únicos números Naturais  $c_0, c_1, c_2, \dots, c_n$  tais que  $a = c_n b^n + c_{n-1} b^{n-1} + \dots + c_1 b + c_0$  para os índices de  $c$  menores do que  $b$  e o último diferente de 0".

Esta expressão, acima, chama-se de "expansão de  $a$  na base  $b$ " e também pode ser descrita como  $a = (c_n c_{n-1} \dots c_1 c_0)_b$ .

Com isso, a expansão do número 524 na base decimal é  $5 \cdot 10^2 + 2 \cdot 10 + 4$ . Perceba que o número que multiplica cada algarismo é a base numérica elevada a um expoente que corresponde à posição do numeral menos 1 e o último algarismo está, na verdade, sendo multiplicado pela base elevada a 0.

### Convertendo de Uma Base Qualquer Para a Base 10

Com o que vimos anteriormente, para convertermos um número que está em uma base qualquer para a base decimal simplesmente devemos escrever e calcular a expansão deste número na base dada.

Por exemplo, vamos calcular quanto vale  $(52024)_7$  na base 10:

$$(52024)_7 = 5 \cdot 7^4 + 2 \cdot 7^3 + 0 \cdot 7^2 + 2 \cdot 7 + 4 = (12709)_{10}$$

Portanto, a representação de 52024 na base 7 é 12709 na base 10.

Convertendo da base 10 para uma base qualquer

Para convertermos um número que esteja na base decimal para outra base  $b$ , devemos encontrar uma expressão da forma  $a = c_n b^n + c_{n-1} b^{n-1} + \dots + c_1 b + c_0$  para o mesmo.

Como exemplo, vamos escrever o número 139 na base 3. Para isso, precisamos encontrar para ele uma expressão do tipo  $139 = a_n 3^n + a_{n-1} 3^{n-1} + \dots + a_1 3 + a_0$ .

Ao utilizarmos a propriedade distributiva da multiplicação, temos que

$$139 = 3 \cdot (a_n 3^{n-1} + \dots + a_1) + a_0$$

Desta forma, concluímos que  $a_0$  é, na verdade, o resto da divisão de 139 por 3. Se realizarmos tal

divisão, teremos que  $139 = 46 \cdot 3 + 1$ , ou seja,  $a_0$  é igual a 1. Com isso,  $46 \cdot 3 = 139 - 1$ . Mas nós sabemos que  $139 = 3 \cdot (a_n 3^{n-1} + \dots + a_1) + a_0 = 3 \cdot (a_n 3^{n-1} + \dots + a_1) + 1$ . Temos, então, que  $139 = 3 \cdot (a_n 3^{n-1} + \dots + a_1)$ , ou seja, eliminamos o termo independente. Vamos repetir essa operação até eliminarmos todos os  $a$ 's e, então, teremos que:

$$139 = 46 \cdot 3 + 1$$

$$46 = 15 \cdot 3 + 1$$

$$15 = 5 \cdot 3 + 0$$

$$5 = 1 \cdot 3 + 2$$

Agora, vamos fazer uma série de substituições:

$$139 = 46 \cdot 3 + 1 = (15 \cdot 3 + 1) \cdot 3 + 1 = 15 \cdot 3^2 + 1 \cdot 3 + 1 = 5 \cdot 3^3 + 1 \cdot 3 + 1 = (1 \cdot 3 + 2) \cdot 3^3 + 1 \cdot 3 + 1 = 1 \cdot 3^4 + 2 \cdot 3^3 + 0 \cdot 3^2 + 1 \cdot 3 + 1$$

Assim, concluímos que a expansão de 139 na base 3 é  $(12011)_3$ . Note que apesar de esta ser a explicação matematicamente correta, muitos cursos técnicos preferem simplificar o processo simplesmente dizendo que, para converter um número na base decimal para outra base devemos realizar sucessivas divisões euclidianas do número pelo algarismo que representa a base até que o divisor seja igual a 0 e pegarmos os restos de baixo para cima. Este procedimento, muito mais simples, está correto e é justificado pelo exemplo anterior.

### Sistemas de Numeração Utilizados na Informática

Os sistemas de numeração mais utilizados na informática são:

**Base 2:** também conhecido como sistema binário. É um sistema posicional composto pelos numerais 0 e 1 e, além da Informática, é utilizado na Eletrônica Digital na implementação de circuitos de portas lógicas. Uma de suas primeiras aplicações na informática surgiu quando da utilização de cartões perfurados para representar informações e programas.

**Base 8:** o sistema octal também é um sistema posicional e foi utilizado na Informática como alternativa ao sistema binário. É composto pelos numerais 0, 1, 2, 3, 4, 5, 6 e 7.

**Base 16:** o sistema hexadecimal é, talvez, um dos mais conhecidos da atualidade. É composto de 16 algarismos, representados por 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F. Trabalha-se com ele como qualquer outro sistema, mas deve-se prestar atenção ao valor dos caracteres alfabético na hora de fazer operações e conversões.

É atualmente a maior alternativa ao sistema binário por ser extremamente compacto e é utilizado para representar portas, interrupções e endereços de memória, além de cores no desenvolvimento web, em substituição ao sistema RGB. Para representar as cores, é utilizada uma notação de seis dígitos, onde cada dupla, da esquerda para a direita, representa o valor da intensidade do vermelho, do verde e do azul, respectivamente, variando de 00 até FF, que representa o valor decimal 255.

Assim, as cores variam de 000000, que representa o preto, até FFFFFFFF, que corresponde ao branco. Os tons de cinza são representados por valores iguais nas três posições, como por exemplo 666666, DEDEDE ou CCCCCC. Quanto mais próximo de FFFFFFFF, mais clara é a tonalidade de cinza.

Ao todo, essa notação hexadecimal permite a representação de mais de 16 milhões e meio de cores. Alguns programas de desenho vetorial e de tratamento de imagem incluem, ainda, uma quarta dupla de valores na notação para representar o nível de transparência da cor selecionada.

Graças à base hexadecimal, as rotinas de tratamento de imagem foram em muito facilitadas. Por exemplo: para fazer o efeito de negativo em uma foto, isto é, inverter suas cores, basta subtrair de FF cada valor da tripla que representa a cor de cada pixel.

**Base 62:** Talvez você nunca tenha ouvido falar deste sistema, mas acredite: você já o utilizou. O

sistema de base 62 está se tornando cada vez mais popular porque seus 62 algarismos são representados pelos numerais de 0 a 9 e pelas letras de A a Z e de a a z. Uma de suas aplicações mais recorrentes está nos famosos encurtadores de URL: o código gerado pelo encurtador nada mais é do que a conversão de um número decimal – geralmente o número de identificação único para cada URL no banco de dados – nesta base. Assim, quanto mais caracteres um encurtador de URLs utilizar para gerar sua URL curta, mais URLs ele terá cadastradas.

Base 64: É um sistema numérico utilizado para codificação de dados binários que precisam ser armazenados e transferidos em meios que foram desenhados originalmente para lidar com dados textuais. É composto pelos algarismos de 0 a 9, pelas letras de A a Z e de a a z e pelos símbolos / e +. O caractere = é utilizado como sufixo especial.

### O Sistema de Numeração Binário

Apesar das tentativas de utilizar-se outros sistemas de numeração mais compactos para substituir o sistema binário, é este que forma a “linguagem” dos computadores. Tudo, desde números, cores, palavras, textos e imagens é tratado pelo sistema operacional nesta base de numeração. No exemplo mais simples, se você fizer uma soma na calculadora de seu sistema operacional, ele irá converter os valores para sua representação base 2, realizará a operação e converterá, novamente, o resultado para a base decimal.

Para convertermos um número em base 2 para a base 10, basta fazermos o que já foi explicado, com a vantagem de que, como os algarismos são apenas 0 e 1, poderemos ter a oportunidade de cancelar várias potências.

Por exemplo: vamos converter o número binário 1010 para a base decimal. Escrevendo a expansão do número na base 2, temos:

$$1010 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2 + 0 = 2^3 + 2 = 8 + 2 = 10$$

Já o procedimento contrário pode ser feito pelo algoritmo explicado anteriormente ou, de maneira mais simples, pelo “truque” de realizar divisões sucessivas por 2 e pegar os restos de baixo para cima:

$$10 = 5 \cdot 2 + 0$$

$$5 = 2 \cdot 2 + 1$$

$$2 = 1 \cdot 2 + 0$$

$$1 = 0 \cdot 2 + 1$$

Desta forma, 10 em base 2 se escreve como 1010.

Agora, o que devemos fazer se desejarmos converter um número que esteja na base 2 para a forma octal ou hexadecimal? Evidentemente, uma maneira seria converter o número em base 2 para a base 10 e fazer a outra conversão, mas isso demoraria muito tempo. Assim, utilizamos um dispositivo mais prático: como a base octal possui 3 dígitos, basta agruparmos os algarismos do número em base 2 em grupos de três, completando com zeros, caso necessário, e convertendo estes grupos para sua forma decimal.

O mesmo procedimento pode ser utilizado para a base hexadecimal, com a diferença de que devemos agrupá-los em grupos de 4 e cuidar o valor correspondente às letras. Por exemplo, para convertermos 11101111 para a base octal, temos que separá-lo em grupos de três algarismos: 011 101 111. Agora, basta realizar a conversão e termos 3 5 7. 357 é a representação do número em base 8.

A adição de dois valores na base dois é realizada por transporte, da mesma forma do que a adição na base 10. Escrevemos os números um abaixo do outro e somamos por colunas da direita para a esquerda. Notemos que há quatro possibilidades:

$$0 + 0 = 0$$



$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

Perceba que na última possibilidade, o resultado é maior do que a quantidade máxima de algarismos que podem ser armazenados em cada posição e, neste caso, há o transporte, ou popular “vai um”.

A subtração ocorre de forma análoga, apenas com a diferença de que:

$$0 - 0 = 0$$

$$0 - 1 = 1$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

No segundo caso, o resultado será 1, mas ocorrerá um transporte para a próxima coluna, que será acumulado no subtraendo.

Logo no começo de sua utilização na informática, surgiram algumas questões conceituais sobre o sistema binário. Uma das principais era a de como fazer um computador representar números negativos. Para tal, reservamos o bit mais significativo de um byte para o sinal, ou seja, o primeiro. Ele será 0 caso o número positivo e 1 caso negativo e usamos o complemento para 1 ou o complemento para 2.

O primeiro consiste em simplesmente inverter zeros e uns: o que é zero vira um e vice-versa. O sinal é dado pelo bit mais significativo. Por exemplo: 3 em base 2 é 11. Representamos-lhe por 011. Para representarmos -3, simplesmente invertemos os bits e temos 100. Já no complemento para 2, invertemos os bits 0 e 1 e somamos 1 ao resultado. A vantagem de utilizar-se o complemento para dois é que ele possui um zero único e preserva as regras de adição e de subtração.

Outro ponto polêmico diz respeito à representação de números com vírgula. A questão apenas foi resolvida com o lançamento da Norma IEEE 754, publicada originalmente em 1985 e revisada em 2008. A norma original dita que os números reais conhecidos por números de ponto flutuante no meio informático – devem ser armazenados em sequências de 32 bits (a revisão de 2008 adicionou a representação de 64-bit para precisão dupla), onde o bit mais significativo é reservado ao sinal – 0 se positivo e 1 se negativo -; os oito bits seguintes são reservados ao expoente e os 23 bits restantes são reservados à parte fracionária. O expoente em questão corresponde a 127 mais o expoente de base 2 do número em questão.

### **O Sistema Binário e a Eletrônica Digital**

Enquanto a eletrônica digital trabalha apenas com dois valores possíveis de amplitude, a analógica admite que a amplitude de um sinal pode assumir qualquer valor entre um mínimo e um máximo dados através do tempo. O valor 1 representa o valor mais alto de amplitude, logicamente assumido como verdadeiro e o 0, o mais baixo, correspondente ao lógico falso. É um grande equívoco, portanto, acreditar que o 0 representa a ausência completa de energia. O complemento 1 a presença de energia.

Os circuitos integrados digitais mais utilizados atualmente podem ser de dois tipos: TTL e CMOS. Os circuitos TTL (Transistor-Transistor Logic) são comercializadas em duas séries de modelos: aqueles que começam com 54 são destinados ao uso militar e os que começam com 74 à utilização comercial. Nesta família de circuitos lógicos, o sinal de entrada pode variar em uma amplitude que vai de 0V a 5V.

Quando o sinal está entre 0V e 0,8V, o circuito o interpreta como sendo o nível mais baixo de amplitude, isto é, o 0 binário. Já quando o sinal recebido está entre 2V e 5V, este é interpretado como sendo o nível alto 1. As tensões entre 0,8V e 2V não são reconhecidas pelo circuito e deve-se,

portanto, evitar seu uso em circuitos digitais. Já na família CMOS (Complementary Metal Oxid Semiconductor), tensões que estejam entre 0V e 1,5V são interpretadas como o baixo nível de energia, ou o Falso lógico. Já as tensões de 3,5V a 5V são reconhecidas como o nível alto ou 1. As tensões intermediárias não são reconhecidas.

Assim, cai por terra o mito de que o 0 binário representa a ausência completa de energia: o zero, na verdade, representa o nível lógico correspondente ao falso. Isso não significa, necessariamente, que haja ausência de energia.

Portas lógicas são circuitos digitais que recebem uma ou mais tensões de entrada e devolvem apenas uma tensão de saída. Através da associação dessas portas, podemos construir circuitos capazes de realizar operações aritméticas simples, utilizando como base a álgebra booleana. Essas portas lógicas são constituídas por transistores que atuam como chaves ao receber um sinal elétrico, permitindo ou não sua passagem.

Hoje, porém, existem circuitos integrados que já desempenham este papel. É desnecessário dizer que elas também estão presentes em processadores, em escala muito reduzida. As principais portas lógicas disponíveis hoje, com exemplos entre parêntesis, são:

NOT (TTL 7404) – também conhecida como inversora. Caso receba um nível lógico 1, devolverá 0 e caso receba um nível lógico 0, devolverá 1;

AND (TTL 7421 / CMOS 4081) – Possui duas entradas e uma saída. Retorna 1 apenas se receber o nível lógico 1 em ambas as entradas;

OR (TTL 7432 / CMOS 4075) – Também possui duas entradas e uma saída, mas devolve 1 se pelo menos uma das entradas informar o nível lógico 1.

NAND (TTL 7400 / CMOS 4011) – Equivale a uma porta AND seguida de uma porta NOT. Retorna 0 se receber o nível lógico 1 em ambas as entradas e 1 nos demais casos;

NOR (TTL 7402 / CMOS 4001) – Equivale à porta OR seguida de NOT. Retorna 1 apenas se as duas entradas informarem o nível lógico 0;

XOR (TTL 7486 / CMOS 4070) – A porta Ou Exclusivo retornará o nível lógico 1 se e somente se apenas uma das entradas for igual a 1 e a outra igual a 0.

Embora seja possível combinar livremente as portas lógicas, as empresas buscam economizar recursos quando de sua utilização em projetos eletrônicos calculando circuitos equivalentes, isto é, circuitos que produzirão o resultado esperado com um menor número de portas. Para isso, os engenheiros utilizam o Teorema de Morgan ou o Mapa de Karnaugh.

Os sistemas de numeração transcenderam o âmbito matemático e são utilizados de várias formas na Informática e na Eletrônica.

### **Aritmética Computacional**

As palavras de um computador são compostas por bits e podem representar números armazenados na memória. Estes números podem ter diferentes significados, como inteiros ou reais, serem positivos ou negativos. A manipulação dos números inclui operações de soma, subtração, multiplicação e divisão.

O objetivo deste texto é mostrar como o hardware implementa a representação dos números, os algoritmos adequados para operações aritméticas e sua implicação no conjunto de instruções da máquina.

Números com Sinal e Números sem Sinal

Os números podem ser representados em qualquer base. Porém, a base 2 é a mais adequada para

os computadores porque tratam com somente dois valores: 0 e 1. Estes valores são implementados facilmente através de circuitos elétricos.

Da aritmética temos que, em qualquer base, o valor do  $i$ -ésimo dígito  $d$  de um número é dado por:  $d \times \text{base}^i$ , onde  $i$  começa em 0 e cresce da direita para a esquerda, de acordo com a posição ocupada pelo dígito. Por exemplo, o número 1011 na base dois é igual a:

$$(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = (1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 0) = 8 + 0 + 2 + 1 = 11$$

Portanto, os bits são numerados como 0,1,2,3,... da direita para a esquerda em uma palavra. Utilizamos a expressão bit menos significativo para designar o bit 0, e a expressão bit mais significativo para designar o bit de mais alta ordem, como por exemplo o bit 31 numa palavra de 32 bits.

Como o tamanho de uma palavra manipulada por um computador tem tamanho limitado, os números que podem ser representados também tem tamanho limitado. Se o tamanho de uma palavra é igual a  $n$  bits, o maior número possível de ser representado é igual a  $2^n$ . Se ao realizarmos operações sobre os números, elas gerarem resultados que não podem ser representados na quantidade de bits reservados ocorre o que denominados de overflow (números muito grandes) ou underflow (números muito pequenos). Tanto o overflow quanto o underflow geram exceções e são tratados pelo sistema operacional.

Os computadores manipulam tanto números positivos quanto números negativos, que são representados em complemento a 2. Nesta convenção os números que possuem zeros à esquerda são considerados positivos e os números com uns à esquerda são considerados negativos. O complemento a 2 é obtido invertendo-se o número binário e depois somando 1 a este valor. Porém, uma regra simples para transformar um número binário em sua representação em complemento a 2 é a seguinte:

- 1) copie da direita para a esquerda todos os bits até encontrar o primeiro bit 1 inclusive e
- 2) inverta todos os demais bits.

A figura abaixo ilustra um exemplo da obtenção de representação em complemento a 2 de um número binário com 4 dígitos.

0110 = 6 na base 10	usando a regra:
1001 (número binário invertido)	0110 → 0110
+ 0001 (soma com 1)	
1010 (complemento a 2)	
Representação em complemento a 2	

A representação em complemento a 2 tem a vantagem de representar números negativos sempre com o bit 1 em sua posição mais significativa. Assim, o hardware só precisa testar este bit para verificar se o número é positivo ou negativo. Este bit é conhecido como bit de sinal. Na figura abaixo está representada uma sequência de números binários (8 dígitos) representados em complemento 2.

0000 0000 = 0
0000 0001 = 1
0000 0010 = 2
0000 0011 = 3
.....
0111 1101 = 125
0111 1110 = 126
0111 1111 = 127
.....
1000 0001 = -127 (primeiro número negativo)
1000 0010 = -126
1000 0011 = -125
.....
1111 1101 = -3
1111 1110 = -2
1111 1111 = -1

Seqüência de números binários representados em complemento a 2

### Conversão entre Diferentes Bases

As bases octal e hexadecimal também são muito úteis em computação. A base octal é representada com 8 dígitos que variam entre 0 e 7. A base hexadecimal é composta por dígitos e letras da seguinte forma: 0 a 9 e as letras a, b, c, d, e e f. Na figura abaixo ilustramos como realizar a mudança da base binária para as demais bases de forma simples. Lembre-se de que precisamos de 3 ou 4 dígitos binários para a mudança de base octal e hexadecimal, respectivamente.

Binário	Octal	Hexadecimal
9 8 7 6 5 4 3 2 1 0	<u>1 0 0 0</u> <u>1 1 1 1 0 1</u>	<u>1 0</u> <u>0 0 1 1</u> <u>1 1 0 1</u>
1 0 0 0 1 1 1 1 0 1 = 573	1 0    7    5 = 1075	2    3    d = 23d
Mudança de base		

Binário	Octal	Hexadecimal
9 8 7 6 5 4 3 2 1 0	<u>1 0 0 0</u> <u>1 1 1 1 0 1</u>	<u>1 0</u> <u>0 0 1 1</u> <u>1 1 0 1</u>
1 0 0 0 1 1 1 1 0 1 = 573	1 0    7    5 = 1075	2    3    d = 23d
Mudança de base		

### Adição e Subtração

Numa soma os bits são somados um a um da direita para a esquerda, com os carries sendo passados para o próximo bit à esquerda. A operação de subtração usa a adição. O subtraendo é simplesmente negado antes de ser somado ao minuendo. Lembre-se que a máquina trata com números representados em complemento a 2. A figura abaixo mostra as operações de soma (6+7) e subtração (7-6) bit a bit entre dois números representados com 4 dígitos binários.

Representação binária	Soma	Subtração
7 = 0 1 1 1	1 1 (vai um)	1 1 1 (vai um)
6 = 0 1 1 0	0 1 1 1	0 1 1 1
13 = 1 1 0 1 (soma)	+ <u>0 1 1 0</u>	+ <u>1 0 1 0</u> (complemento a 2)
1 = 0 0 0 1 (subtração)	1 1 0 1	0 0 0 1
Operações de soma e subtração (complemento a 2) com representação binária		

Como citado anteriormente, tanto a soma como a subtração podem gerar overflow ou underflow, se o resultado obtido não puder ser representado pela quantidade de bits que formam uma palavra. Se somarmos ou subtraírmos dois números com sinais contrários, nunca ocorrerá overflow ou underflow. Isto porque operandos com sinais contrários nunca podem ser maior do que qualquer dos operandos.

O overflow ocorre quando somamos dois operandos positivos e obtemos um resultado negativo, ou vice-versa. Isto significa que utilizamos o bit de sinal, gerando um carry, para armazenar um valor pertencente ao resultado da operação. Raciocínio semelhante é realizado para detectar a ocorrência do underflow numa subtração. Neste caso, o bit de sinal também é usado para armazenar um valor pertencente ao resultado da operação.

Os projetistas de um sistema devem decidir onde tratar a ocorrência de overflow ou de underflow em operações aritméticas. Elas podem ser tratadas tanto por hardware quanto por software. Pode existir a detecção por hardware que gera uma exceção, e que depois é tratada por software.

### Operações Lógicas

Os computadores manipulam palavras, mas é muito útil, também, manipular campos de bits dentro de uma palavra ou mesmo bits individuais. O exame de caracteres individuais (8 bits) dentro de uma palavra é um bom exemplo dessa necessidade. Assim, as arquiteturas de conjuntos de instruções incluem instruções para manipulação de bits.

Um dos tipos de instrução utilizados são as de deslocamento de bits. As instruções podem deslocar bits tanto à direita quanto à esquerda. Todos os bits são movidos para o lado determinado e os bits

que ficam vazios são preenchidos com zeros. Outras instruções lógicas muito úteis que são implementadas na unidade lógica e aritmética de um processador são as operações NOT, AND, OR e XOR. A figura abaixo mostra as operações lógicas, bit a bit, de deslocamento à direita, à esquerda, NOT, AND, OR e XOR.

Desl. à direita	Desl. à esquerda	NOT	AND	OR	XOR
1101	1101	1101	1101	1101	1101
0110	1010	0010	<u>0101</u>	<u>0101</u>	<u>0101</u>
			0101	1101	1000

Operações lógicas

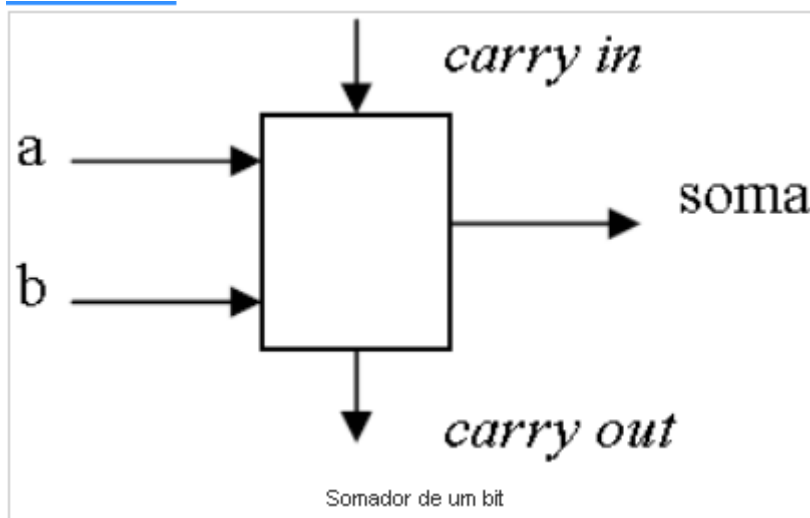
### Construção de uma Unidade Lógica Aritmética

A unidade lógica aritmética (ALU – Arithmetic Logic Unit) é o dispositivo que realiza as operações lógicas e aritméticas, definidas pelo conjunto de instruções, dentro do processador.

A ALU é construída basicamente por quatro blocos básicos de hardware: portas AND, portas OR, NOT (inversores) e multiplexadores.

As implementações de operações lógicas são as mais simples de serem realizadas, pois elas são mapeadas diretamente com componentes do hardware.

A próxima função a ser incluída é a adição. Supondo que temos apenas um bit para ser somado, necessitamos de um circuito com duas entradas para os operandos, uma saída para a soma resultante, uma entrada relativa ao carry in e uma saída para o carry out. A figura abaixo mostra este somador.



Podemos especificar as saídas soma e carry out através de equações lógicas, que podem ser implementadas a partir dos blocos de hardware mencionados anteriormente. A equação lógica para gerar o bit carry out é dada por:

$$\text{CarryOut} = (b \cdot \text{CarryIn}) + (a \cdot \text{CarryIn}) + (a \cdot b)$$

E, a equação lógica para gerar o bit soma é dada por:

$$\text{Soma} = (a \cdot b \cdot \text{CarryIn}) + (a \cdot \bar{b} \cdot \text{CarryIn}) + (a \cdot b \cdot \bar{\text{CarryIn}}) + (a \cdot \bar{b} \cdot \bar{\text{CarryIn}})$$

Para completar o projeto de uma ALU de n bits podemos conectar n somadores de um bit. Os carry outs gerados pelos bits menos significativos da operação podem ser propagados por toda a extensão do somador, gerando um carry out no bit mais significativo do resultado da operação. Este somador é denominado somador de carry propagado.

A operação de subtração pode ser realizada somando-se o minuendo com a negação do subtraendo. Este efeito é realizado acrescentando uma entrada complementada de  $b$  ao somador e ativando o carry in do bit menos significativo para um. O somador então calcula  $a + b + 1$ . Ao escolhermos a versão invertida de  $b$  obtemos:

$$a + b + 1 = a + (b + 1) = a + (-b) = a - b$$

A simplicidade do projeto do hardware de um somador para números de complemento a 2 demonstra porque esta representação tornou-se um padrão para operações aritméticas inteiras em computadores.

O problema com o somador de carry propagado está relacionado a velocidade de propagação do carry, que é realizada sequencialmente. Num projeto de hardware a velocidade é um fator crítico. Para solucionar este problema existem diversos esquemas para antecipar o carry. Porém, nestes esquemas são utilizadas mais portas lógicas, o que provoca um aumento no custo.

Um dos esquemas para antecipar o carry é denominado carry lookahead. Os somadores que utilizam o esquema de carry lookahead baseiam sua implementação em vários níveis de abstração. Utilizando a abreviação  $c_i$  para representar o  $i$ -ésimo bit de carry, podemos escrever a equação do carry como:

$$c_i = (b_i \cdot c_i) + (a_i \cdot c_i) + (a_i \cdot b_i) = (a_i \cdot b_i) + (a_i + b_i) \cdot c_i$$

Os termos  $(a_i \cdot b_i)$  e  $(a_i + b_i)$  são tradicionalmente chamados de gerador ( $g_i$ ) e propagador ( $p_i$ ), respectivamente. Usando estas relações para definir  $c_{i+1}$ , obtemos:

$$c_{i+1} = g_i + p_i \cdot c_i$$

Qualquer equação lógica pode ser implementada com uma lógica de dois níveis. Mesmo esta formulação mais simplificada pode gerar equações muito grandes e, portanto levar a circuitos lógicos relativamente grandes e caros, dependendo do número de bits a serem somados.

### **Multiplicação**

Para realizar a multiplicação são necessários dois operandos, um multiplicando e um multiplicador, para gerar um operando produto. O algoritmo da multiplicação diz que os dígitos do multiplicando devem ser multiplicados pelos dígitos do multiplicador um de cada vez, da direita para a esquerda, deslocando os produtos intermediários um dígito à esquerda em relação ao imediatamente anterior.

Uma observação importante é que o número de bits do produto final ( $n+m$ ) é maior do que o número de bits do multiplicando ( $n$ ) ou do multiplicador ( $m$ ). Além disso, a multiplicação também precisa tratar a ocorrência de overflow.

Considerando os dígitos binários 0 e 1, temos apenas duas possibilidades de escolha, a cada passo da multiplicação:

1. Coloque uma cópia do multiplicando (multiplicando  $\times$  1) no lugar apropriado, se o dígito do multiplicador for igual a 1, ou
2. Coloque 0 (multiplicando  $\times$  0) no lugar apropriado, se o dígito do multiplicador for igual a 0.

Assim, é necessário desenvolver um algoritmo em hardware que seja eficiente para realizar a multiplicação.

Um método elegante de multiplicar números com sinal recebeu o nome de algoritmo de Booth. Ele foi elaborado a partir da constatação de que com a capacidade de somar e de subtrair números existem várias maneiras de se calcular um produto. Por exemplo, podemos substituir um string de uns no multiplicador por uma subtração quando encontramos o primeiro 1, e por uma soma ao encontrarmos o último 1 do string.

Booth buscou atingir maior velocidade de processamento utilizando operações de deslocamento, que ainda hoje são operações mais rápidas do que operações de soma. Baseado nesta observação, se desejarmos maior velocidade na multiplicação de números inteiros por uma potência de 2, basta que utilizemos operações de deslocamento indicando a quantidade de deslocamentos igual ao expoente.

A grande vantagem do algoritmo de Booth é tratar com facilidade os números com sinal. O raciocínio de Booth foi classificar os grupos de bits como início, meio e fim de um string de uns. Naturalmente um string de zeros não precisa ser considerado.

### **Algoritmo da Multiplicação**

Este algoritmo precisa apenas de dois passos principais: o teste do produto e o seu deslocamento; pois os registradores Produto e Multiplicador podem ser combinados em um só. O algoritmo começa com o Multiplicador na metade à direita do registrador Produto, e 0 na metade à esquerda.

1. Testa se Produto é igual a 0 ou 1.
2. Produto = 0, passa ao item 4.
3. Produto = 1, soma o Multiplicando à metade esquerda do Produto e coloca o resultado na metade à esquerda do registrador Produto.
4. Desloca o registrador Produto 1 bit à direita.
5. Verifica se foram realizadas todas as repetições necessárias de acordo com o tamanho da palavra, se não volta ao item 1.
6. Fim.

### **Divisão**

A divisão é a operação recíproca da multiplicação. Dentre as operações aritméticas é a que aparece menos frequentemente nos códigos dos programas.

No algoritmo da divisão são utilizados dois operandos, o dividendo e o divisor, e produzidos dois resultados o quociente e o resto. A relação entre os componentes da divisão pode ser expressa da seguinte forma:

$$\text{Dividendo} = \text{quociente} \times \text{divisor} + \text{resto},$$

Onde o resto é sempre menor que o divisor.

Às vezes, os programas usam a divisão simplesmente para obter o resto, ignorando o quociente. Além disso, é necessário que seja detectada a divisão por zero, que é matematicamente inválida.

### **Algoritmo da Divisão**

Da mesma forma que foram combinados registradores na multiplicação, também na divisão são combinados dois registradores, o Resto e o Quociente. O algoritmo começa com o Resto na metade à esquerda do registrador Resto, e o Quociente na metade à direita.

1. Desloca o registrador Resto 1 bit à esquerda.
2. Subtrai o registrador Divisor da metade à esquerda do registrador Resto e armazena o resultado na metade esquerda do registrador Resto.
3. Testa se Resto é menor do que 0.
4. Resto < 0, restaura valor original com Divisor + metade esquerda do Resto, armazenando na metade esquerda do registrador Resto e deslocando 1 bit à esquerda, inserindo 0 no novo bit menos significativo, passa ao item 6.
5. Resto ≥ 0, desloca o registrador Resto 1 bit à esquerda, inserindo 1 no novo bit mais à direita.

6. Verifica se foram realizadas todas as repetições necessárias de acordo com o tamanho da palavra, se não volta ao item 1.

7. Desloca a metade a esquerda do registrador Resto 1 bit à direita, Fim.

### **Ponto Flutuante**

Assim como os números decimais podem ser representados em notação científica normalizada os números binários também podem. A aritmética computacional que manipula os números binários em notação científica normalizada é denominada de aritmética de ponto flutuante.

Os projetistas do hardware devem encontrar um compromisso entre a mantissa e o expoente dos números em ponto flutuante. A relação entre a mantissa e o expoente é expressa do seguinte modo: o aumento do número de bits reservados à mantissa aumenta a precisão do número, enquanto o aumento do número de bits reservados ao expoente aumenta o intervalo de variação dos números representados.

Deve ser observado que as interrupções relativas ao overflow e ao underflow também ocorrem na representação em ponto flutuante. Porém, neste caso, overflow e o underflow ocorrem quando o expoente é muito grande ou muito pequeno, respectivamente, para ser armazenado no espaço reservado a ele.

Outra questão que os projetistas devem decidir é se vão ser utilizados os mesmos registradores tanto para números inteiros quanto para números de ponto flutuante. A adoção de registradores diferentes aumenta ligeiramente o número de instruções necessárias a execução do programa.

O impacto maior está na criação de um conjunto de instruções de transferência de dados para mover os dados entre os registradores de ponto flutuante e a memória. Os benefícios estão no fato de não precisar aumentar o tamanho do campo nas instruções para diferenciar os operandos e aumentar a banda passante dos registradores.

A partir de 1980 todos os computadores projetados adotam uma representação padrão para números em ponto flutuante denominada IEEE 754. A adoção deste padrão facilita a portabilidade de programas (precisão simples = 1 bit de sinal, 8 bits de expoente e 23 bits de mantissa + 1 implícito = 24, precisão dupla = 1 bit de sinal, 11 bits de expoente e 52 bits de mantissa + 1 implícito = 53).

A adição e a multiplicação com números de ponto flutuante, na sua essência, utilizam as operações inteiras correspondentes para operar as mantissas, mas é necessária uma manipulação extra nos expoentes e para a normalização do resultado.

### **Algoritmo da Adição em Ponto Flutuante**

1. Compare o expoente dos dois números. Desloque o menor número à direita até que seu expoente se iguale ao maior número.
2. Some as mantissas.
3. Normalize a soma, deslocando à direita e incrementando o expoente ou deslocando à esquerda e decrementando o expoente.
4. Teste se há overflow ou underflow.
5. Sim, gera exceção.
6. Não, arredonde a mantissa para o número de bits apropriado.
7. Testa se resultado está normalizado.
8. Sim, Fim.
9. Não, retorna ao passo 3.

### **Algoritmo da Multiplicação em Ponto Flutuante**



1. Soma os expoentes com peso dos dois números, subtraindo o valor do peso da soma para obter o novo expoente.
2. Multiplique as mantissas.
3. Normalize o produto se necessário, deslocando à direita e incrementando o expoente.
4. Teste se há overflow ou underflow.
5. Sim, gera exceção.
6. Não, arredonde a mantissa para o número de bits apropriado.
7. Teste se resultado está normalizado.
8. Não, retorna ao passo 3.
9. Sim, faça o sinal do produto positivo se ambos os sinais dos operandos originais são os mesmos, caso contrário o sinal é negativo.

### **Características e Recursos dos Processadores**

Apesar do processador ser o componente mais importante do micro, já que é ele quem processa quase todas as informações, ele não é necessariamente o maior responsável pelo desempenho. Na verdade, dependendo da aplicação à qual o micro se destina, o desempenho do processador pode ser menos importante que a quantidade de memória RAM, que o desempenho da placa de vídeo 3D, ou até mesmo que o desempenho do disco rígido.

Tenha em mente que o computador é um conjunto, cada componente depende dos demais para mostrar o seu potencial.

Dizemos que um micro é tão rápido quanto seu componente mais lento. Como estamos falando de um conjunto, apenas um componente que apresente uma baixa performance será suficiente para colocar tudo a perder. Assim como vemos em outras situações, num carro por exemplo, onde um simples pneu furado pode deixar o carro parado na estrada.

Se o micro tiver pouca memória RAM por exemplo, o sistema operacional será obrigado a usar memória virtual, limitando a performance ao desempenho do disco rígido, que é centenas de vezes mais lento que ela. Caso o micro não possua memória cache, o desempenho ficará limitado ao desempenho da memória RAM, que é muito mais lenta que o processador e por aí vai.

Dizemos neste caso, que o componente de baixo desempenho é um gargalo, pois impede que o conjunto manifeste todo o seu potencial. Às vezes, simplesmente aumentar a quantidade de memória RAM, operação que custa relativamente pouco, é capaz de multiplicar a velocidade do micro.

Mas, apesar de tudo, o processador ainda é o componente básico de qualquer PC. Com o avanço cada vez mais rápido da tecnologia, e várias empresas disputando o mercado, os projetistas vem sendo obrigados a desenvolver projetos cada vez mais ousados a fim de produzir os processadores com o melhor desempenho.

Isso é excelente para nós, mas também pode trazer armadilhas, já que com projetos tão diferentes, cada processador acaba saindo-se bem em algumas aplicações, mas muito mal em outras. Não dá para julgar o desempenho do processador apenas pela frequência de operação, como fazíamos na época do 486, os tempos mudaram.

Mas, já que está aqui, que tal conhecermos os avanços pelos quais os processadores passaram até chegar aos dias de hoje? Vamos discutir primeiro algumas características básicas dos processadores, conhecer os pioneiros da década de 70 e avançar pelos anos 80 e 90, até chegar nos dias de hoje.

Características Básicas dos processadores modernos

Existem no mercado vários modelos de processadores, que apresentam preços e desempenho bem diferentes. Este tópico inicial se destina a estabelecer os diferenciais básicos que determinam a performance de um processador, a parte teórica que vai lhe ajudar a compreender a diferença entre os processadores que vamos examinar com detalhes mais adiante.

Quando vamos comprar um processador, a primeira coisa que perguntamos é qual sua frequência de operação, medida em Megahertz (MHz) ou milhões de ciclos por segundo, frequência também chamada de clock.

Acontece, que nem sempre um processador com uma velocidade de operação mais alta é mais rápido do que outro que opera a uma frequência um pouco mais baixa. A frequência de operação de um processador indica apenas quantos ciclos de processamentos são realizados por segundo, o que cada processador é capaz de fazer em cada ciclo já é outra história.

Imagine um processador 486 de 100 MHz, ao lado de um Pentium também de 100 MHz. Apesar da frequência de operação ser a mesma, o 486 perderia feio em desempenho. Na prática, o Pentium seria pelo menos 2 vezes mais rápido. Isto acontece devido à diferenças na arquitetura dos processadores e também no coprocessador aritmético e cache.

### **Coprocessador Aritmético**

Todos os processadores da família x86, usada em micros PC, são basicamente processadores de números inteiros. Muitos aplicativos porém, precisam utilizar valores de maior precisão, assim como funções matemáticas complexas, como Seno, Coseno, Tangente, etc., para realizar suas tarefas. Este é o caso dos programas de CAD, planilhas, jogos com gráficos tridimensionais e de processamento de imagens em geral.

A função do coprocessador aritmético é justamente auxiliar o processador principal no cálculo destas funções complexas, cada vez mais utilizadas, principalmente em jogos. É como um matemático profissional que ajuda o processador a resolver os problemas mais complexos, que ele demoraria muito para resolver sozinho.

Até o 386, o coprocessador era apenas um acessório que podia ser comprado à parte e instalado num encaixe apropriado da placa mãe, sendo que cada modelo de processador possuía um modelo equivalente de coprocessador. O 8088 utilizava o 8087, o 286 o 287, o 386SX e 386DX utilizavam respectivamente o 387SX e o 387DX e o 486SX utilizava 487DX.

O problema nesta estratégia é que como poucos usuários equipavam seus micros com coprocessadores aritméticos, a produção destes chips era baixa, e conseqüentemente os preços eram altíssimos, chegando ao ponto de em alguns casos o coprocessador custar mais caro que o processador principal. Com o aumento do número de aplicativos que necessitavam do coprocessador, sua incorporação ao processador principal a partir do 486DX foi um passo natural. Com isso, resolveu-se também o problema do custo de produção dos coprocessadores, barateando o conjunto.

Atualmente, o desempenho do coprocessador determina o desempenho do micro em jogos e aplicativos gráficos em geral, justamente as aplicações onde os processadores atuais são mais exigidos. Infelizmente, o desempenho do coprocessador é uma característica que varia muito entre os processadores atuais.



Encaixe para o coprocessador aritmético

### **Memória Cache**

Enquanto os processadores tornaram-se quase 10 mil vezes mais rápidos desde o 8088 (o processador usado no XT), a memória RAM, sua principal ferramenta de trabalho, pouco evoluiu em performance.

Quando foram lançados os processadores 386, percebeu-se que as memórias não eram mais capazes de acompanhar o processador em velocidade, fazendo com que muitas vezes ele tivesse que ficar “esperando” os dados serem liberados pela memória RAM para poder concluir suas tarefas, perdendo muito em desempenho.

Se na época do 386 a velocidade das memórias já era um fator limitante, imagine o quanto este problema não atrapalharia o desempenho dos processadores que temos atualmente. Para solucionar este problema, começou a ser usada a memória cache, um tipo ultra-rápido de memória que serve para armazenar os dados mais frequentemente usados pelo processador, evitando na maioria das vezes que ele tenha que recorrer à comparativamente lenta memória RAM.

Sem ela, o desempenho do sistema ficará limitado à velocidade da memória, podendo cair em até 95%! São usados dois tipos de cache, chamados de cache primário, ou cache L1 (level 1), e cache secundário, ou cache L2 (level 2).

O cache primário é embutido no próprio processador e é rápido o bastante para acompanhá-lo em velocidade. Sempre que um novo processador é desenvolvido, é preciso desenvolver também um tipo mais rápido de memória cache para acompanhá-lo.

Como este tipo de memória é extremamente caro (chega a ser algumas centenas de vezes mais cara que a memória RAM convencional) usamos apenas uma pequena quantidade dela. O 486 traz apenas 8 KB, o Pentium traz 16 KB, enquanto o Pentium II e o Pentium III trazem 32 KB, enquanto o Athlon e o Duron da AMD trazem 128 KB.

Para complementar, usamos também um tipo um pouco mais lento de memória cache na forma do cache secundário, que por ser muito mais barato, permite que seja usada uma quantidade muito maior. Nos micros 486 o mais comum é o uso de 128 ou 256 KB de cache L2, enquanto nos micros mais modernos o mais comum é o uso de 512 KB. Dependendo do processador usado, o cache L2 pode vir embutido no próprio processador ou fazer parte da placa mãe.

Sempre que o processador precisar ler dados, os procurará primeiro no cache L1. Caso o dado seja encontrado, o processador não perderá tempo, já que o cache primário funciona na mesma frequência que ele. Caso o dado não esteja no cache L1, então o próximo a ser indagado será o cache L2. Encontrando o que procura no cache secundário, o processador já perderá algum tempo, mas não tanto quanto perderia caso precisasse acessar diretamente a memória RAM.

Por outro lado, caso os dados não estejam em nenhum dos dois caches, não restará outra saída senão perder vários ciclos de processamento esperando que eles sejam entregues pela lenta memória RAM. Para exemplificar, imagine que você estivesse escrevendo um e-mail e de repente precisasse de uma informação que você havia anotado em um papel.

Se o papel estivesse sobre sua mesa, você poderia lê-lo sem perder tempo. Se estivesse dentro de uma gaveta da sua mesa, já seria necessário algum tempo para encontrá-lo enquanto se ele estivesse perdido em algum lugar de um enorme fichário do outro lado da sala, seria preciso um tempo enorme.

Antigamente, era comum as placas mães virem com soquetes apropriados, que permitiam ao usuário adicionar mais memória cache caso quisesse. Os módulos adicionais, chamados de módulos COAST (cache on a stick) eram relativamente acessíveis, levando muita gente a fazer o upgrade.

Entretanto, atualmente esta possibilidade não existe mais, pois a grande maioria dos processadores já trazem o cache L2 integrado, não permitindo qualquer modificação, já que não dá para abrir o processador e soldar mais cache. Mesmo no caso de processadores que ainda usam cache embutido na placa mãe, como o K6-2, não existe mais o encaixe para adicionar mais cache.

Ou seja, atualmente a quantidade de cache que você deseja no processador ou placa mãe deve ser decidida antes da compra, baseado nas opções disponíveis. Uma vez adquiridos o processador e a placa mãe não será possível fazer qualquer alteração.

### **Processadores RISC X Processadores CISC**

Sempre houve uma grande polêmica em torno de qual dessas plataformas é melhor. Talvez você ache inútil eu estar falando sobre isto aqui, mas é interessante que você compreenda a diferença entre estas duas plataformas, para entender vários aspectos dos processadores modernos.

Um processador CISC (Complex Instruction Set Computer, ou “computador com um conjunto complexo de instruções”), é capaz de executar várias centenas de instruções complexas diferentes, sendo extremamente versátil. Exemplos de processadores CISC são o 386 e o 486.

No começo da década de 80, a tendência era construir chips com conjuntos de instruções cada vez mais complexos. Alguns fabricantes porém, resolveram seguir o caminho oposto, criando o padrão RISC (Reduced Instruction Set Computer, ou “computador com um conjunto reduzido de instruções”).

Ao contrário dos complexos CISC, os processadores RISC são capazes de executar apenas algumas poucas instruções simples. Justamente por isso, os chips baseados nesta arquitetura são mais simples e muito mais baratos. Outra vantagem dos processadores RISC, é que, por terem um menor número de circuitos internos, podem trabalhar a frequências mais altas. Um exemplo são os processadores Alpha, que em 97 já operavam a 600 MHz.

Pode parecer estranho que um chip que é capaz de executar algumas poucas instruções, possa ser considerado por muitos, mais rápido do que outro que executa centenas delas, seria como comparar um professor de matemática com alguém que sabe apenas as quatro operações. Mas, um processador RISC é capaz de executar tais instruções muito mais rapidamente.

A idéia principal, é que apesar de um processador CISC ser capaz de executar centenas de instruções diferentes, apenas algumas são usadas frequentemente. Poderíamos então criar um processador otimizado para executar apenas estas instruções simples que são mais usadas. Como de qualquer forma, pouca gente programa diretamente em Assembly, bastaria alterar os compiladores, para que os programas fossem compatíveis com os novos processadores.

É indiscutível, porém, que em muitas tarefas os processadores CISC saem-se melhor, principalmente pelo seu grande número de recursos. Por isso, ao invés da vitória de uma das duas tecnologias, atualmente vemos processadores híbridos, que são essencialmente processadores CISC, mas incorporam muitos recursos encontrados nos processadores RISC (ou vice-versa).

Apesar de por questões de Marketing, muitos fabricantes ainda venderem seus chips, como sendo “Processadores RISC”, não existe praticamente nenhum processador atualmente que siga estritamente uma das duas filosofias. Tanto processadores da família x86, como o Pentium II, Pentium III e AMD Athlon, quanto processadores supostamente RISC, como o MIPS R10000 e o HP PA-8000 misturam características das duas arquiteturas, por simples questão de performance. Por que ficar de um lado ou de outro, se é possível juntar o melhor dos dois mundos? A última coisa que os fabricantes de processadores são é teimosos, sempre que aparece uma solução melhor, a antiga é abandonada.

Examinando de um ponto de vista um pouco mais prático, a vantagem de uma arquitetura CISC é que já temos muitas das instruções guardadas no próprio processador, o que facilita o trabalho dos programadores, que já dispõe de praticamente todas as instruções que serão usadas em seus programas. No caso de um chip estritamente RISC, o programador já teria um pouco mais de trabalho, pois como disporia apenas de instruções simples, teria sempre que combinar várias instruções sempre que precisasse executar alguma tarefa mais complexa.

Seria mais ou menos como se você tivesse duas pessoas, uma utilizando uma calculadora comum, e outra utilizando uma calculadora científica. Enquanto estivessem sendo resolvidos apenas cálculos simples, de soma, subtração, etc. quem estivesse com a calculadora simples poderia até se sair melhor, mas ao executar cálculos mais complicados, a pessoa com a calculadora científica disporia de mais recursos.

Nos chips atuais, que são na verdade misturas das duas arquiteturas, juntamos as duas coisas. Internamente, o processador processa apenas instruções simples. Estas instruções internas, variam de processador para processador, são como uma luva, que se adapta ao projeto do chip. As instruções internas de um K6 são diferentes das de um Pentium por exemplo.

Sobre estas instruções internas, temos um circuito decodificador, que converte as instruções complexas utilizadas pelos programas em várias instruções simples que podem ser entendidas pelo processador. Estas instruções complexas sim, são iguais em todos os processadores usados em micros PC. É isso que permite que um K6 e um Pentium sejam compatíveis entre si.

O conjunto básico de instruções usadas em micros PC é chamado de conjunto x86. Este conjunto é composto por um total de 187 instruções, que são as utilizadas por todos os programas. Além deste conjunto principal, alguns processadores trazem também instruções alternativas, que permitem aos programas executar algumas tarefas mais rapidamente do que seria possível usando as instruções x86 padrão. Alguns exemplos de conjuntos alternativos de instruções são o MMX (usado a partir do Pentium MMX), o 3D-NOW! (usado pelos processadores da AMD, a partir do K6-2), e o SSE (suportado pelo Pentium III).

### **PCs x Macs**

Continuando na discussão de processadores RISC e CISC, vamos estudar um pouco sobre a arquitetura de dois processadores atuais, o G4, utilizado nos micros Macintosh e o AMD Athlon, usado em micros PC.

Existe uma idéia geral de que o G4, usado nos Macs é um processador RISC, enquanto os processadores usados em micros PC, incluindo o Pentium III e o Athlon são todos CISC. Ambas as afirmações estão erradas. Na verdade, tanto o G4, quanto o Athlon e o Pentium III são considerados processadores Post-RISC, processadores que possuem um conjunto de instruções gigantesco, maior do que o conjunto de instruções de um processador CISC típico.

A diferença é que toda essa gigantesca gama de instruções diferentes, podem ser decodificadas em instruções RISC simples, estas sim que serão processadas. A "conversão" das instruções é feita por um componente especial do processador, chamado de Hardware Decoder, encontrado tanto no G4 quanto no Athlon.

O G4 possui um enorme conjunto de instruções, assim como os processadores x86, mas todas as instruções que podem ser convertidas pelo Hardware decoder e em seguida processadas. O Hardware Decoder é extremamente rápido, por isso não compromete o desempenho do processador.

De fato, a perda de desempenho por usar este grande conjunto de instruções que precisam ser quebradas em instruções menores é de menos de 1%. É por isso que os processadores atuais abandonaram a idéia RISC original: a perda de desempenho é ínfima perto do ganho de flexibilidade.

O Athlon por sua vez, tem que ser compatível com o conjunto de instruções x86, caso contrário não poderia ser usado em micros PC. As instruções x86 consistem em basicamente dois tipos de instruções, as instruções simples, que podem ser diretamente processadas pelo Hardware decoder, sem perda de tempo, e as instruções complexas, que são quebradas em instruções simples por outro componente, chamado Microcode decoder.

As instruções simples, que podem ser diretamente processadas, são as mais frequentemente usadas nos programas. De fato, num programa atual típico, é composto de entre 95 e 97% destas instruções simples.

O restante são as instruções complexas, que apesar de raramente usadas são as que dão mais trabalho, pois precisam passar por um processo de decodificação muito mais lento, feito pelo Microcode Decoder.

Para amenizar Este problema, a AMD incluiu um buffer de pré extração no Athlon, que funciona como uma espécie de fila por onde as instruções já decodificadas passam antes de ser processadas. Graças a isto, o processador pode processar outras instruções enquanto aguarda o Microcode Decoder decodificar cada instrução complexa, sem perder muito tempo.

Com isto, mesmo mantendo compatibilidade com o conjunto de instruções x86, o Athlon perde muito pouco em desempenho em relação ao G4, isto naturalmente comparando dois processadores de mesma frequência. O IPC, ou seja, o número de instruções processadas por ciclo de ambos é muito próximo, o que garante que um Athlon de 500 MHz apresente um desempenho muito parecido com um G4 também de 500 MHz.

### **Front End e Back End**

Qualquer processador atual pode ser dividido em dois blocos básicos, o Front End e o Back End.

O Front End corresponde aos circuitos que decodificam as instruções, no caso o Hardware decoder, Microcode decoder e buffer de pré extração que acabei de explicar, junto com mais alguns componentes, como os circuitos de Branch Prediction (que ordenam as instruções de forma que o processador possa processar o maior número possível de instruções por ciclo e o cache L1. Estes componentes são a “porta de entrada” do processador, tendo a função de preparar as instruções para serem processadas.

O Back End é a parte do processador que finalmente processa as instruções, sendo composto basicamente pelas unidades de execução.

Como vimos, o fato como ambos os processadores decodificam as instruções, contando com o Hardware decoder é bastante semelhante, mas o Athlon possui alguns componentes a mais para garantir compatibilidade com as instruções x86. Isto não atrapalha o desempenho do processador, mas o torna um projeto mais complexo.

Em termos de unidades de execução, ou seja, o Back End, é que os processadores mostram mais algumas diferenças na forma como processam as instruções já decodificadas.

O Athlon possui um total de 9 unidades de execução, enquanto o G4 possui apenas 6. A diferença parece grande, mas na prática o desempenho é quase o mesmo, veja por que:

O Athlon possui 3 unidades de execução para leitura/gravação de dados na memória, enquanto o G4 possui apenas uma. O ponto é que todas as instruções, tanto de inteiros, quanto de ponto flutuante no Athlon, vem com um espaço reservado para uma instrução de leitura/gravação, espaço que nem sempre é preenchido, fazendo com que as 3 unidades fiquem ociosas na maior parte do tempo, apesar de agilizarem algo de vez em quando.

No G4, só existe uma unidade de leitura/gravação, mas que em compensação fica ocupada na maior parte do tempo. Na prática, esta única unidade acaba fazendo o mesmo volume de trabalho das três do Athlon, que ficam boa parte do tempo ociosas. Sem dúvida, o G4 perde alguma coisa em termos de desempenho, mas muito pouco.

Em termos de unidades de execução de inteiros e de ponto flutuante, que são as mais importantes, temos especificações parecidas em ambos:

O Athlon possui três unidades de ponto flutuante (que formam o coprocessador aritmético), o mesmo número encontrado no G4. Apenas para efeito de comparação, o Pentium 3 possui apenas duas. Com o mesmo número de unidades, o desempenho dos dois processadores no quesito ponto flutuante é quase igual.

Já em termos de unidades de processamento de inteiros, o cenário muda um pouco de figura, pois o Athlon possui três unidades de execução contra apenas duas do G4.

Isto garante que o Athlon tenha um desempenho um pouco melhor que o G4 em aplicativos de escritórios, mas a diferença é pequena, pois o desempenho real também depende do cache, velocidade de acesso à memória, etc.

Em termos de instruções 3D, o Athlon conta com o 3D-Now, o famoso conjunto de instruções, embutido nos processadores AMD que permite melhorar o desempenho do processador em jogos e aplicativos 3D. O Athlon traz também o velho MMX, que garante algum ganho em aplicativos multi-mídia.

O G4 por sua vez traz um conjunto unificado, o Altivec, que inclui tanto instruções 3D (como no 3D-Now!), quanto instruções multimídia (como no MMX), isto garante que tanto o Athlon quanto o G4 possuam armas semelhantes neste quesito, o resto fica por conta dos programadores.

### **Do 8086 ao Pentium MMX**

O primeiro microprocessador foi lançado pela Intel em 1971 e se chamava i4004. Este era um processador extremamente simples, formado por pouco mais de 2000 transistores, mas que foi o precursor dos processadores que temos atualmente. A chamada lei de Moore, que leva o nome do fundador da Intel, Gordon Moore, prega que a potência dos processadores dobra a cada 18 meses. Apesar desta previsão ter sido feita no final da década de 70, continuou mantendo-se verdadeira até os dias de hoje, com uma precisão notável.

De lá pra cá, foi um longo caminho. Enormes investimentos foram feitos e muitos dos maiores gênios do planeta trabalharam em busca de soluções para questões cada vez mais complexas. Vamos agora examinar os avanços feitos desde o 8088, usado no XT, até o Pentium, onde estudaremos quando e porque recursos como o modo protegido e a multiplicação de clock foram introduzidos, e no que eles afetam o funcionamento do processador. Entendendo estes conceitos, você poderá facilmente entender as diferenças entre os processadores Pentium III, Athlon, K6-3 etc. que temos atualmente e veremos com mais detalhes adiante, assim como dos processadores que vierem a ser lançados futuramente que, pode ter certeza, continuarão utilizando os mesmos conceitos básicos.

### **8088**

O 8088 era na verdade uma versão econômica do processador 8086, que havia sido lançado pela Intel em 78. Quando a IBM estava desenvolvendo seu computador pessoal, chegou a ser cogitado o uso do 8086, mas acabou sendo escolhido o 8088 devido ao seu baixo custo.

Tanto o 8086 quanto o 8088 são processadores de 16 bits e eram considerados avançadíssimos para a época, apesar de serem extremamente simples para os padrões atuais. A diferença entre eles é que o 8088, apesar de internamente trabalhar com palavras binárias de 16 bits, usava um barramento de apenas 8 bits, o que permitiu à IBM utilizar os mesmos componentes usados nos computadores de 8 bits da época, que eram muito mais baratos do que os periféricos de 16 bits.

Esta arquitetura permitiu ao primeiro PC competir na mesma faixa de preço dos computadores de 8 bits mais populares e, ao mesmo tempo, possuir um desempenho bem superior devido ao seu processador de 16 bits. O 8088 é capaz de acessar até 1 MB de memória RAM, e funciona a 4.77 MHz, recursos incríveis para a época, já que estamos falando de um processador lançado no final de 1979.

Falando em recursos, só para matar sua curiosidade, o PC original da IBM, lançado em Agosto de 1981 possuía apenas 64 KB de memória RAM (a versão mais simples vinha com apenas 16 KB), monitor MDA mono de 12 polegadas, usava uma unidade de disquetes de 5 1/4 de apenas 160 KB e vinha sem disco rígido. O sistema operacional usado era o MS-DOS 1.0 (na época ainda chamado de PC-DOS), que foi desenvolvido pela Microsoft com base num sistema operacional mais simples, chamado QDOS, comprado da Seattle Computers, uma pequena empresa desenvolvedora de sistemas. Na verdade, a Microsoft foi a segunda opção da IBM, depois de ter sua proposta de licença recusada pela Digital Research, que na época desenvolvia versões do seu CP/M para várias arquiteturas diferentes.

Dois anos depois, foi lançado o PC XT, que apesar de continuar usando o 8088 de 4.77 MHz, vinha bem mais incrementado, com 256 KB de RAM, disco rígido de 10 MB, monitor CGA e o MS-DOS 2.0.

Mesmo com o surgimento dos micros 286, o XT ainda continuou sendo bastante vendido, pois era mais barato. Fabricantes de clones criaram projetos de micros XTs mais avançados, equipados com processadores 8088 de 8 MHz, discos rígidos maiores e até 640 KB de memória RAM.

### **Segmentação de Endereços**

Um recurso bem interessante, usado no 8088, é a segmentação de endereços, que permitiu aumentar a quantidade de memória RAM suportada pelo processador.

Para que o processador possa acessar a memória RAM, é preciso que a memória seja dividida em endereços. Cada byte depositado na memória recebe um endereço único, assim como cada rua do Brasil tem um CEP diferente. Como o 8088 pode lidar apenas com palavras binárias de 16 bits, a princípio não seria possível para ele acessar mais do que 64 Kbytes de memória RAM, já que 16 bits permitem apenas 65,536 combinações diferentes (2 elevado à 16ª potência).

Se o 8088 pudesse acessar apenas 64 KB de memória RAM, os micros baseados nele seriam muito limitados e poderiam apenas rodar programas muito simples. Para você ter uma idéia, 64 KB não dariam nem mesmo para carregar o DOS 3.0.

Para solucionar este problema, foi adotada uma solução bastante engenhosa: apesar do processador continuar podendo acessar apenas 64 KB de memória de cada vez, foram criados mais 4 bits de endereçamento, que permitem o acesso a 16 blocos de memória. Como cada bloco possui 64 KB, chegamos a 1 MB inteiro de capacidade total. Basicamente criamos 16 áreas diferentes de memória, cada uma com 64 KB, que é o máximo que o 8088 pode endereçar.

O processador pode acessar uma única área de cada vez. Se por exemplo, está sendo usado o bloco 1, e de repente é preciso ler um dado gravado no bloco 2, é preciso limpar todos os endereços relativos ao bloco 1 e carregar os endereços do bloco 2. Neste momento, o processador perde o acesso ao bloco 1 e passa a enxergar apenas o segundo bloco.

Quando novamente for preciso ler ou gravar dados no bloco 1 (ou qualquer outro bloco), novamente são carregados os endereços relativos a ele, e o acesso ao bloco 2 será perdido. É mais ou menos como se você precisasse fazer anotações em várias páginas de um caderno. Como só é possível ler ou escrever em uma página de cada vez, você precisaria ficar continuamente virando as páginas.

## **286**

O processador 286 foi lançado em Fevereiro de 1982, apenas 6 meses após a IBM ter lançado o seu primeiro PC. Porém, o 286 passou a ser utilizado apenas em 1984, quando a IBM lançou o seu PC AT. Esta demora é justificável, pois, para lançar um computador usando o novo processador da Intel, foi preciso desenvolver toda uma nova arquitetura. Da placa de vídeo ao gabinete, praticamente tudo foi mudado, o que somado à burocracia e a longos períodos de testes antes do lançamento, demandou um certo tempo.

Atualmente, o período de desenvolvimentos dos periféricos é muito mais curto. Quase sempre quando um novo processador é lançado, já temos placas mãe para ele disponíveis quase que imediatamente, pois o desenvolvimento é feito de forma simultânea.

O 286 trouxe vários avanços sobre o 8088. Ele utilizava palavras binárias de 16 bits tanto interna quanto externamente, o que permitia o uso de periféricos de 16 bits, muito mais avançados do que os usados no PC original e no XT. O custo destes periféricos desta vez não chegou a ser um grande obstáculo, pois enquanto o PC AT estava sendo desenvolvido, eles já podiam ser encontrados com preços mais acessíveis.

O principal avanço trazido pelo 286 são seus dois modos de operação, batizados de “Modo Real” e “Modo Protegido”. No modo real, o 286 se comporta exatamente como um 8086 (apesar de mais rápido), oferecendo total compatibilidade com os programas já existentes. Já no modo protegido, ele manifesta todo o seu potencial, incorporando funções mais avançadas, como a capacidade de acessar até 16 Megabytes de memória RAM (usando os 24 bits de endereçamento do 286), multitarefa, memória virtual em disco e proteção de memória.

Assim que ligado, o processador opera em modo real, e com uma certa instrução, passa para o modo protegido. O problema é que trabalhando em modo protegido, o 286 deixava de ser compatível com os programas escritos para o modo real, inclusive com o próprio MS-DOS. Para piorar, o 286 não possuía nenhuma instrução que fizesse o processador voltar ao modo real, isto era possível apenas resetando o micro.

Isso significa que um programa escrito para rodar em modo protegido, não poderia usar nenhuma das rotinas de acesso a dispositivos do MS-DOS, tornando inacessíveis o disco rígido, placa de vídeo, drive de disquetes memória, etc., a menos que fossem desenvolvidas e incorporadas ao programa todas as rotinas de acesso a dispositivos necessárias. Isso era completamente inviável para os



desenvolvedores, pois para projetar um simples jogo, seria praticamente preciso desenvolver todo um novo sistema operacional. Além disso, o programa desenvolvido rodaria apenas em micros equipados com processadores 286, que ainda eram minoria na época, tendo um público alvo muito menor. De fato, apenas algumas versões do UNIX e uma versão do OS/2 foram desenvolvidas para utilizar o modo protegido do 286.

Basicamente, os micros baseados no 286 eram usados para rodar aplicativos de modo real, que também podiam ser executados em um XT, aproveitando apenas a maior velocidade do 286. Falando em velocidade, a primeira versão do 286 funcionava a apenas 6 MHz, sendo lançada logo depois uma nova versão de 8 MHz, que foi usada no PC AT. Posteriormente, foram desenvolvidas versões de até 20 MHz.

Devido às várias mudanças na arquitetura, destacando o acesso mais rápido à memória e alterações no conjunto de instruções do processador, que permitiam realizar muitas operações de maneira mais rápida e eficiente, um 286 consegue ser quase 4 vezes mais rápido que um 8088 do mesmo clock.

### **386**

O 386 foi lançado apenas em Outubro de 85, três anos e meio depois do 286. Desta vez, a diretoria da IBM demorou muito para chegar à um acordo e desenvolver um sistema baseado no 386, dando tempo para a Compaq sair na frente. Este foi um verdadeiro marco pois, de repente, as companhias perceberam que não eram mais obrigadas a seguir a IBM.

Qualquer um que tivesse tecnologia suficiente poderia sair na frente, como fez a Compaq. A partir daí, a IBM começou a gradualmente perder a liderança do mercado, tornando-se apenas mais um entre inúmeros fabricantes de PCs.

O 386 trouxe vários recursos novos. Para começar, o 386 trabalha tanto interna quanto externamente com palavras de 32 bits e é capaz de acessar a memória usando um barramento de 32 bits, permitindo uma transferência de dados duas vezes maior. Como o 386 pode trabalhar com palavras binárias de 32 bits, é possível acessar até 4 GB de memória (2 elevado à 32ª potência), mesmo sem usar a segmentação de endereços, como no 8088 e no 286.

Assim como o 286, o 386 continua possuindo os dois modos de operação. A diferença é que no 386 já é possível alternar entre o modo real e o modo protegido livremente. Um programa que rode sobre DOS, pode chavear o processador para o modo protegido, para beneficiar-se de suas vantagens, e voltar ao modo real sempre que precisar usar alguma sub-rotina do DOS, de maneira transparente ao usuário. Neste caso, é usado um programa de DPMI ("DOS Protected Mode Interface", ou "interface DOS de modo protegido") para fazer o chaveamento entre os dois modos.

Toda vez que o programa precisa usar alguma sub-rotina do DOS, ele passa o comando ao chaveador e fica esperando. O chaveador por sua vez, passa o processador para o modo real, executa o comando, chaveia o processador para o modo protegido e entrega o resultado ao aplicativo, que continua trabalhando como se nada tivesse acontecido. Um bom exemplo de programa de DPMI é o DOS4GW, que é usado por muitos jogos que rodam sobre o MS-DOS, como o Doom, Sim City 2000 e vários emuladores de vídeo-games.

O esquema de chaveamento também é utilizado pelo Windows 3.x, que já inclui todas as rotinas necessárias, dispensando qualquer programa de DPMI. O Windows 95/98 também pode chavear para o modo real caso precise carregar algum driver de dispositivo de modo real. Porém, devido ao modo virtual 8086, que veremos logo a seguir, não é preciso colocar o processador em modo real para executar aplicativos MS-DOS dentro do Windows 95/98

Ter um processador 386 é o requisito mínimo para rodar qualquer sistema operacional ou aplicativo de modo protegido moderno. Com um 386, um mínimo de memória RAM e espaço em disco suficiente, você pode rodar o Windows 95 e a maioria dos aplicativos para ele, embora bem lentamente devido à pouca potência do processador.

Com um simples 286, no máximo você poderá rodar o DOS e aplicativos mais simples, que trabalhem somente com o modo real. Também é possível rodar o Windows 3.0, porém em modo "Standard", onde é possível acessar todos os 16 MB de memória permitidos pelo 286, mas sem memória virtual nem multitarefa.

## A Introdução do Cache

Os processadores 386 acima de 20 MHz eram muito rápidos para as memórias RAM existentes na época. Por isso, a cada acesso, o processador tinha que ficar “esperando” os dados serem liberados pela memória RAM para poder concluir suas tarefas, perdendo muito em desempenho. Para solucionar esse problema, passaram a ser usadas pequenas quantidades de memória cache na grande maioria das placas mãe para micros 386 e superiores.

A memória cache é um tipo de memória ultra-rápida, que armazena os dados mais usados pelo processador, evitando na grande maioria dos casos, que ele precise perder tempo buscando dados diretamente na lenta memória RAM. Mesmo uma pequena quantidade de memória cache é capaz de melhorar bastante a velocidade da troca de dados entre o processador e a RAM.

Apesar de já ser bem mais rápido que a memória RAM, o 386 ainda não era um processador muito rápido, justamente por isso, ainda não era tão dependente do desempenho da memória cache quanto os processadores atuais. Um 386 equipado com memória cache é de 20 a 30% mais rápido que um 386 da mesma frequência, mas sem memória cache, enquanto um processador moderno pode ficar até 20 vezes mais lento caso sejam desabilitados tanto o cache L1 quanto o cache L2.

### 386SX

Como o 386 era um processador de 32 bits, foi preciso desenvolver toda uma nova categoria de chipsets e circuitos de apoio para trabalhar com ele, o que acabou encarecendo bastante os sistemas baseados no 386 e afastando muitos compradores em potencial.

Para contornar este problema, a Intel optou por lançar uma versão de baixo custo do 386, batizada de 386SX, que apesar de continuar funcionando internamente com palavras de 32 bits, comunicava-se com a memória RAM e os demais periféricos usando palavras de 16 bits (como o 286). Apenas para diferenciar os dois processadores, a Intel passou a chamar o 386 original de 386DX.

Esta arquitetura permitiu que fossem aproveitados os mesmos periféricos usados em placas de micros 286, tornando as máquinas baseadas no 386SX muito mais acessíveis. Pra você uma idéia, um PC básico equipado com um 386SX, chegava a custar menos de 1,000 dólares, quase metade de um equipamento com uma configuração parecida baseado no 386DX.

Apesar de, devido ao preço, o 386SX ter tornado-se uma boa opção em termos de custo-benefício, em termos de performance ele fica bem atrás de um 386DX da mesma frequência, pois apesar de internamente os processadores serem idênticos, o SX usa praticamente os mesmos componentes usados nos micros 286, acessa a memória usando palavras de 16 bits e, para completar, as placas mãe para ele não possuem memória cache.

### Modo Real x Modo Protegido

Operando em modo real, o processador funciona exatamente como um 8086, apenas trabalhando com uma velocidade maior. Não somente o 386, mas todos os processadores atuais podem alternar entre o modo real e o modo protegido livremente, sempre que necessário. No modo real, rodamos o MS-DOS e outros aplicativos de modo real mais antigos, enquanto no modo protegido rodamos o Windows e seus programas.

Com certeza, alguma vez ao tentar rodar um programa antigo, você já se deparou com uma enigmática mensagem de falta de memória, apesar dos manuais do programa dizerem que ele precisa apenas de 500 ou 600 KB de memória e você ter instalado bem mais do que isso. Estas mensagens surgem por que estes programas rodam com o processador operando em modo real onde, como o 8086, ele é capaz de reconhecer apenas o primeiro Megabyte da memória RAM. Este primeiro Megabyte por sua vez, é subdividido em dois blocos, chamados de memória convencional e memória estendida.

A memória convencional corresponde aos primeiros 640 Kbytes da memória, e é a área de memória usada pelos programas que operam em modo real. Os 384 Kbytes restantes são chamados de memória superior, e são reservados para armazenar uma cópia do BIOS, que passa a ser executado mais rapidamente, já que a memória RAM é muito mais rápida do que o chip de memória ROM ou

Flash onde ele é originalmente armazenado. Esta cópia do BIOS é chamada de "Shadow", ou sombra, e serve para aumentar o desempenho geral do sistema. A memória superior também é usada para armazenar sombras dos BIOS de outros dispositivos, como placas de vídeo, aumentando também a velocidade de operação destes periféricos.

Apesar de existirem 640 Kbytes de memória convencional, protos para ser usada por qualquer programa que opere em modo real, nem toda esta memória fica disponível, já que parte dela é usada pelo MS-DOS e drivers de dispositivos de modo real. É possível liberar mais memória convencional, editando os arquivos de inicialização do DOS, conseguindo assim rodar estes programas.

Quando o computador é ligado, o processador está operando em modo real. Quem dá o comando para que ele mude para o modo protegido é o sistema operacional. No caso do Windows, este comando é dado durante o carregamento do sistema.

Em modo protegido, o processador é capaz de reconhecer toda a RAM instalada no sistema, além de incorporar recursos como a multitarefa e a memória virtual em disco. É neste modo que usamos a interface gráfica do Windows e rodamos seus aplicativos.

### **Recursos do Modo Protegido**

Apesar de, em nome da compatibilidade retroativa com programas desenvolvidos para micros PC XT e 286, tanto o 386 como todos os processadores atuais poderem operar em modo real, apenas no modo protegido eles incorporam os recursos mais avançados, que permitem a existência dos softwares que temos atualmente.

A partir do 386, poucas funções novas foram incorporadas aos novos processadores. Basicamente, evoluímos apenas em termos de velocidade. Tanto que, com um simples 386, é possível rodar praticamente qualquer aplicativo mais atual, apenas com uma velocidade menor.

O modo protegido traz basicamente quatro novos recursos: memória virtual, multitarefa, proteção de memória e o modo virtual 8086.

### **Memória Virtual**

A capacidade do 386 de trabalhar com vários aplicativos ao mesmo tempo (multitarefa) é realmente muito útil, mas esta característica traz um pequeno problema: abrindo vários aplicativos sucessivamente, logo a memória RAM do sistema se esgota. Para corrigir este problema, o modo protegido traz também a memória virtual, que permite criar um arquivo temporário no disco rígido, chamado de Swap File, ou arquivo de troca, que funciona como uma extensão da memória RAM, permitindo abrir quantos aplicativos forem necessários, até que o espaço do disco rígido se esgote.

Por exemplo, só o Windows 2000 Professional, junto com os serviços básicos ocupa cerca de 40 MB de memória. Se você abrir o Word 97, serão necessários mais 10 Megabytes, um total de quase 50 MB. Caso o micro em questão possua apenas 32 MB de memória, seria criado um arquivo temporário de 18 MB no disco rígido, que armazenaria os dados que não couberam na memória RAM.

O problema em usar memória virtual é que o disco rígido é centenas de vezes mais lento do que a memória RAM. Um disco rígido razoável possui um tempo de acesso em torno de 10 milissegundos (milésimos de segundo) enquanto um módulo de memória PC-100 possui um tempo de acesso inferior a 10 nanossegundos (bilionésimos de segundo) ou seja, um tempo de acesso um milhão de vezes menor! Em termos de taxa de transferência, novamente temos um contraste marcante: 800 MB para o módulo de memória e de 5 a 20 MB (dependendo do modelo) para o disco rígido.

Graças a este abismo, apesar dos programas funcionarem normalmente usando memória virtual, o sistema vai ficando cada vez mais lento. Experimente, por exemplo, tentar trabalhar em um PC com apenas 4 MB de RAM (seja qual for o processador) rodando o Windows 95. A lentidão é insuportável.

No Windows 3.x, era necessário reservar uma quantidade espaço do disco rígido para a memória virtual, quantidade que podia ser configurada livremente através do Painel de Controle. O problema é que este espaço ficava indisponível. Se você possuísse um disco de 800 MB, e reservasse 200 para a memória virtual, ficaria com apenas 600 MB para instalar programas e guardar arquivos. Se por

outro lado, você reservasse pouco espaço para a memória virtual, ficaria com pouca memória para abrir vários programas e trabalhar com arquivos grandes.

Apartir do Windows 95 este problema foi resolvido com a adoção de um arquivo de troca dinâmico, que vai aumentando ou diminuindo de tamanho conforme a necessidade de memória, evitando o desperdício de espaço em disco que tínhamos no Windows 3.x.

Apartir do Windows 95, existe também uma administração mais racional dos recursos do sistema, movendo os arquivos mais importantes, acessados com mais frequência para memória RAM (ou memória cache, dependendo da importância do arquivo), e deixando apenas arquivos usados mais raramente no arquivo de troca. Esta simples medida diminui bastante a perda de performance causada pelo uso da memória virtual.

No Windows 2000 é possível determinar um valor inicial e um valor máximo para um arquivo de troca. No caso do Linux, a fim de melhorar o desempenho, os desenvolvedores optaram por criar um sistema de arquivos próprio para a memória virtual.

### **Multitarefa**

Multitarefa significa executar mais de uma tarefa de cada vez, como assobiar e chupar cana ao mesmo tempo. Apesar de na vida real não ser muito fácil fazer duas coisas ao mesmo tempo, do ponto de vista de um computador este processo é relativamente simples.

Todos os aplicativos são carregados na memória e o processador passa a executar algumas instruções de cada aplicativo por vez. Como o processador é capaz de executar vários milhões de instruções por segundo, esta troca é feita de maneira transparente, como se os aplicativos estivessem realmente sendo executados ao mesmo tempo. Enquanto o processador dá atenção para um aplicativo, todos os demais ficam paralisados, esperando sua vez.

### **Memória Protegida**

Usando a multitarefa, quase sempre teremos vários aplicativos carregados na memória, seja na memória RAM ou no arquivo de troca. Se não houvesse nenhum controle por parte do processador, um aplicativo poderia expandir sua área de memória, invadindo áreas de outros aplicativos e causando travamentos no micro.

Um editor de imagens, por exemplo, precisa ocupar mais memória conforme as imagens vão sendo abertas ou criadas. Sem nenhuma orientação por parte do processador, simplesmente seriam ocupadas as áreas adjacentes, que poderiam tanto estar vazias, quanto estar ocupadas pelo processador de textos, por exemplo.

Para colocar ordem na casa, foi desenvolvido o recurso de proteção de memória, que consiste no processador isolar a área de memória ocupada por cada aplicativo, impedindo que ele ocupe outras áreas ao seu bel prazer. Se, por acaso, o programa precisar de mais memória, o próprio processador irá procurar uma área vazia de memória e ordenar ao aplicativo que ocupe a área reservada.

Existem basicamente dois tipos de multitarefa, denominadas multitarefa preemptiva e multitarefa cooperativa, que diferem justamente pelo uso ou não da proteção de memória.

O Windows 3.x, apesar de ser considerado um sistema operacional multitarefa, não é capaz de usar o recurso de proteção de memória, nele é usada a multitarefa cooperativa, que consiste em cada aplicativo usar os recursos do processador por um certo tempo, passar para outro programa e esperar novamente chegar sua vez para continuar executando suas tarefas.

A alternância entre os programas neste caso não é comandada pelo sistema e sim pelos próprios aplicativos. Neste cenário, um aplicativo mal comportado poderia facilmente monopolizar o sistema, consumindo todos os recursos do processador por um longo período, ou mesmo invadir áreas de memória ocupadas por outros aplicativos, causando em qualquer um dos casos o famoso GPF, ("General Protection Fault", ou "falha geral de proteção") que tanto atormentava os usuários do Windows 3.x.

Experimente tentar fazer dois irmãos dividirem os mesmo brinquedo; pode funcionar durante um certo tempo, mas uma hora um não vai querer deixar o outro brincar e vai sair briga, exatamente como acontece com os aplicativos dentro da multitarefa cooperativa.

O Windows 95/98 por sua vez, usa a multitarefa preemptiva, isolando as áreas de memória ocupadas pelos aplicativos. Isto garante uma estabilidade bem maior do que a que temos no Windows 3.11. Porém, o modo como a multitarefa preemptiva é implementada no Windows 95 assim como do Windows 98 e do Windows Millennium, que são baseados no mesmo kernel (núcleo) do Windows 95, ainda possui dois problemas graves:

O primeiro é que, quando é executado um programa de 16 bits, o Windows 95 cai em multitarefa cooperativa para poder rodar o programa, deixando de proteger as áreas de memória e tornando-se tão vulnerável quanto o Windows 3.11.

Porém, mesmo usando apenas aplicativos de 32 bits os travamentos ainda são comuns, pois o Windows 95 os serviços do sistema não tem prioridade sobre os aplicativos. Isto significa que caso um aplicativo qualquer entre em loop, poderá consumir todos os recursos do processador, neste caso o sistema operacional ficará paralisado, simplesmente sem ter como fechar o aplicativo e restaurar o sistema, obrigando o usuário a resetar o micro e perder qualquer trabalho que não tenha sido salvo. Na verdade costuma-se dizer que o Windows 95/98 utiliza multitarefa semi-preemptiva, pois não utiliza todos os recursos de uma verdadeira multitarefa.

A solução para este problema veio com o Windows NT. Desde suas primeiras versões, o Windows NT é bem estável neste aspecto, pois implementa a multitarefa preemptiva de forma completa. As tarefas executadas pelo sistema operacional, são priorizadas sobre as de qualquer outro aplicativo. Isto significa que em nenhuma situação, um aplicativo terá como passar por cima do sistema operacional e consumir todos os recursos do processador como acontece no Windows 95/98.

Na prática, significa que o sistema até pode travar devido a algum bug, mas se algum aplicativo travar ou tentar invadir uma área de memória não designada para ele, simplesmente será fechado, permitindo que todos os demais aplicativos continuem trabalhando sem problemas. Você logo notará quais aplicativos costumam dar problemas, bastando substituí-los por versões mais recentes que corrijam seus bugs ou mesmo passar a usar um programa concorrente.

Tanto o Windows 2000, quanto o XP são baseados no kernel do Windows NT e mantém o mesmo sistema de funcionamento. Por ter sido inspirado no Unix, o Linux utiliza multitarefa preemptiva desde suas primeiras versões, é por isso que o Linux é considerado um dos sistemas mais estáveis, a ponto de ser usado em vários dos mais importantes servidores do planeta.

O MacOS por sua vez, utilizou a multitarefa cooperativa durante muito mais tempo, até a versão 9.x. Os usuários dos Mac só passaram a ter disponível um sistema com multitarefa preemptiva a partir do MacOS X, que é baseado no FreeBSD, um sistema Unix de código aberto, semelhante ao Linux em vários aspectos. A Apple usou o FreeBSD para construir o Darwin, que é a base do sistema e completou a obra com a interface Aqua, que mantém a idéia de facilidade de uso das versões anteriores do MacOS.

### **Modo Virtual 8086**

Apesar de, operando em modo real, o processador ser totalmente compatível com qualquer programa antigo, seria impossível executar um aplicativo de modo real dentro do Windows 95 ou qualquer outro sistema operacional que utilize o modo protegido. Seria preciso fechar o Windows e fazer o processador voltar para o modo real para poder executar o aplicativo.

Pensando nesta possível limitação, os projetistas da Intel desenvolveram o modo virtual 8086 onde o processador, operando em modo protegido, é capaz de simular vários ambientes de modo real, cada um com 1 MB de memória e total acesso ao hardware do micro, chamados de máquinas virtuais. É como se dentro do 386 fossem abertos vários XTs completos, um para cada programa de modo real a ser executado. É justamente o modo virtual 8086 que permite abrir janelas DOS dentro do Windows 95/98.

Como o processador continua em modo protegido, cada máquina virtual tem sua área isolada na memória. O programa roda sem prejudicar a estabilidade do sistema.

**486**

O 386 foi o grande marco dos processadores para micros PC, pois foi o primeiro processador a trazer o conjunto de instruções x86, que são suportadas por todos os processadores modernos. A partir dele, surgiram vários melhoramentos, mas apenas em termos de desempenho.

Apesar de não trazer instruções novas, o 486 conquistou seu lugar na história, por trazer vários recursos que continuam sendo usados até os processadores atuais. Em primeiro lugar, o 486 foi o primeiro processador a trazer cache integrado. Eram 8 Kbytes, mas que eram capazes de entregar dados a cada ciclo do processador. Como os fabricantes continuaram incluindo cache na placa mãe, um pouco mais lentos, mas em maior quantidade, surgiu também a distinção entre o cache L1 e o L2.

Outra evolução foi o coprocessador aritmético. Ao invés do caríssimo componente que deveria ser adquirido separadamente, o coprocessador passou a ser um item de série. Este foi o impulso que faltava para a popularização de vários programas e o surgimento de jogos bem mais elaborados.

Com tudo isso, um 486 é quase duas vezes mais rápido do que um 386 da mesma frequência. Em alguns aplicativos, que dependem do coprocessador aritmético, um 486 chega a ser 10 vezes mais rápido.

Como fez anteriormente com o 386, a Intel criou um 486 de baixo custo chamado de 486SX. A diferença entre o SX e o 486 original, que passou a ser chamado de 486DX. Os dois compartilhavam a mesma arquitetura, mas o SX vinha sem o coprocessador aritmético, o que o tornava muito mais lento em aplicativos gráficos e científicos.

Para os proprietários, existia a opção de posteriormente comprar um 80487SX, um coprocessador aritmético que era vendido separadamente. O problema era que comprado separadamente, o coprocessador custava quase tanto quanto um processador 486DX que já vinha com o coprocessador embutido, definitivamente um péssimo negócio. Para evitar confusão, o 486 original passou a ser chamado de 486DX.

Foram lançadas versões do 486 rodando à 25 MHz, 33 MHz e 40 MHz, porém, criou-se uma barreira, pois não haviam na época circuitos de apoio capazes de trabalhar a mais de 40 MHz. Para solucionar esse problema, foi criado o recurso de Multiplicação de Clock, através do qual o processador trabalha internamente à uma velocidade maior do que a da placa mãe. Foram lançados então os processadores 486DX2 (que trabalhavam ao dobro da frequência da placa mãe) e logo depois os 486DX4 (que trabalhavam ao triplo da frequência da placa mãe):

Processador	Placa mãe	Multiplicador
<b>486DX-2 50 MHz</b>	25 MHz	2x
<b>486DX-2 66 MHz</b>	33 MHz	2x
<b>486DX-2 80 MHz</b>	40 MHz	2x
<b>486DX-4 75 MHz</b>	25 MHz	3x
<b>486DX-4 100 MHz</b>	33 MHz	3x
<b>486DX-4 120 MHz</b>	40 MHz	3x

Com isso, surgiram também as placas mãe upgradable, que permitem atualizar o processador, apenas configurando alguns jumpers da placa.

Os processadores 486, a partir do DX-33 foram os primeiros a utilizar cooler, que naquela época eram dissipadores com menos de um centímetro de altura, com exaustores minúsculos. Conforme os processadores passaram a dissipar cada vez mais calor, os coolers foram crescendo na mesma proporção, até chegar nos exageros que vemos atualmente 😊

### **Multiplicação de Clock**

Dentro de qualquer computador, os dados são transmitidos e processados na forma de sinais elétricos. O processador é muito pequeno, não mede mais do que 1, ou 1,2 centímetros quadrados. A placa mãe por sua vez é muito maior que isso.

Graças a esta diferença de proporções, acaba sendo muito mais fácil desenvolver um processador capaz de operar a, digamos, 2 gigahertz, do que uma placa mãe capaz de acompanhá-lo. Apesar dos sinais elétricos percorrerem os circuitos a uma velocidade próxima da da luz, estamos falando de bilhões de transmissões por segundo.

O recuso de multiplicação de clock surgiu para evitar que os processadores ficassem limitados à frequência da placa mãe. Num Pentium III de 800 MHz por exemplo, a placa mãe opera a apenas 100 MHz. O multiplicador é de 8x.

Hoje em dia os processadores trazem tanto cache L1, quanto cache L2 integrados, operando na mesma frequência do restante do processador, o que diminui muito a dependência da velocidade da memória RAM, que sempre opera na mesma frequência de a placa mãe, meros 100 ou 133 MHz. Mesmo assim, quanto maior for o multiplicador, maior será a perda de desempenho. Um bom exemplo disso, é uma comparação entre o Celeron 766 (que usa bus de 66 MHz) e o Celeron 800 (que já usa bus de 100 MHz). Apesar da frequência de operação ser quase a mesma, o Celeron 800 chega a ser 20% mais rápido, graças ao acesso mais rápido à memória.

Apesar das limitações, o recurso de multiplicação de clock é indispensável atualmente, pois sem ele seria impossível desenvolver processadores muito rápidos, já que não é possível aumentar a frequência das placas mãe e dos demais periféricos na mesma proporção do aumento do clock nos processadores. Se o Pentium III, por exemplo, tivesse que trabalhar na mesma frequência da placa mãe, não passaríamos de 100 ou 133 MHz.

Nos PCs 486, Pentium, MMX e K6 é necessário configurar o multiplicador manualmente, através de alguns jumpers da placa mãe. É uma maravilha, principalmente quando você não tem o manual da placa em mãos. Mas, a partir do Pentium II, a placa é capaz de detectar automaticamente o multiplicador. Na verdade, a partir do Pentium II, todos os processadores Intel têm o seu multiplicador travado ainda na fábrica. Não é possível alterá-lo mesmo que queira.

### **Pipeline**

Até o 386, os processadores da família x86 eram capazes de processar apenas uma instrução de cada vez. Uma instrução simples podia ser executada em apenas um ciclo de clock, enquanto instruções mais complexas demoravam vários ciclos de clock para serem concluídas. Seria mais ou menos como montar um carro de maneira artesanal, peça por peça.

Para melhorar o desempenho do 486, a Intel resolveu usar o pipeline, uma técnica inicialmente usada em processadores RISC, que consiste em dividir o processador em vários estágios distintos. O 486, possui um pipeline de 5 níveis, ou seja, é dividido em 5 estágios.

Quando é carregada uma nova instrução, ela primeiramente passa pelo primeiro estágio, que trabalha nela durante apenas um ciclo de clock, passando-a adiante para o segundo estágio. A instrução continua então sendo processada sucessivamente pelo segundo, terceiro, quarto e quinto estágios do processador. A vantagem desta técnica, é que o primeiro estágio não precisa ficar esperando a instrução passar por todos os demais para carregar a próxima, e sim carregar uma nova instrução assim que se livra da primeira, ou seja, depois do primeiro pulso de clock.

As instruções trafegam dentro do processador na ordem em que são processadas. Mesmo que a instrução já tenha sido processada ao passar pelo primeiro ou segundo estágio, terá que continuar seu caminho e passar por todos os demais. Se por acaso a instrução não tenha sido completada mesmo após passar pelos 5, voltará para o primeiro e será novamente processada, até que tenha sido concluída.

Desta maneira, conseguimos que o processador seja capaz de processar simultaneamente, em um único ciclo de clock, várias instruções que normalmente demorariam vários ciclos para serem processadas. Voltando ao exemplo do carro, seria como se trocássemos a produção artesanal por uma linha de produção, onde cada departamento cuida de uma parte da montagem, permitindo montar vários carros simultaneamente.

O uso dos 5 estágios de pipeline no 486 não chega a multiplicar por cinco a performance do processador, na verdade a performance não chega nem mesmo a dobrar, mas o ganho é bem significativo.

## Pentium

Assim como o 486, o Pentium é um processador de 32 bits, capaz de acessar até 4 GB de memória RAM. Mas, novamente o processador trouxe várias melhorias que o tornaram muito mais rápido que a geração anterior. Não é à toa que o primeiro Pentium operava a apenas 60 MHz, e era, na época do lançamento, muito mais caro que um 486DX4-100. O Pentium é de 65 a 100% mais rápido que um 486 do mesmo clock. Como o processador aritmético também foi completamente remodelado, o Pentium acaba sendo ainda mais rápido em aplicativos que demandam um grande número de cálculos.

Os processadores Pentium existiram em versões de 60 a 200 MHz, sempre utilizando multiplicação de clock (com exceção apenas para as versões de 60 e 66 MHz):

Processador	Placa Mãe	Multiplicador
Pentium 60, 90, 120, 150 e 180 MHz	60 MHz	Respectivamente 1x, 1.5x, 2x, 2.5x e 3x
Pentium 66, 100, 133, 166 e 200 MHz	66 MHz	Respectivamente 1x, 1.5x, 2x, 2.5x e 3x
Pentium 75 MHz	50 MHz	1.5 x
Pentium 200 MHz	66 MHz	3 x

Como na época dos micros 486, as placas mãe para processadores Pentium (com exceção de placas muito antigas) suportam várias frequências de barramento e vários multiplicadores distintos. Na maioria dos casos é possível configurar a placa mãe para utilizar qualquer processador da família.

### Melhorias no Cache L1

A primeira mudança trazida pelo Pentium foi o aumento da quantidade de cache L1, que passou a ser de 16 KB, o dobro do 486. Além do aumento da capacidade, foram implementados três novos recursos, que tornaram o cache ainda mais eficiente:

A primeira medida foi a divisão do cache em dois blocos de 8 KB, um dedicado a armazenar dados e outro dedicado a armazenar instruções. Esta divisão permite que tanto as instruções a serem executadas pelo processador (comparação, multiplicação, soma, decisão, etc.) quanto os dados a serem processados, possam ser acessados simultaneamente no cache, aumentando sua eficiência.

Se, por exemplo, um programa qualquer ordena que o processador leia um número gravado na memória e verifique se ele é maior ou menor que 10, temos duas instruções (ler o número e compará-lo com o número 10) e duas variáveis (o número 10 e o número a ser lido).

Com um cache unificado, como no 486, primeiro seriam lidas as instruções e em seguida as variáveis. No cache dividido do Pentium, ambos podem ser lidos no mesmo ciclo de clock, poupando tempo.

Outra modificação foi a ampliação do barramento de dados entre o processador e o cache. Enquanto no 486 podem ser lidos ou gravados até 128 bits de dados por ciclo de clock, no Pentium podem ser lidos ou gravados até 256 bits no cache de instruções e mais 256 no cache de dados. Como ambos os caches podem ser acessados simultaneamente, temos um barramento total de 512 bits, o quádruplo do que tínhamos no 486! Este barramento mais largo permite que quantidades maiores de dados possam ser lidos a partir do cache no mesmo espaço de tempo, permitindo ao cache acompanhar a maior velocidade de processamento do Pentium.

A última medida foi a adoção de um cache Write Back, que é capaz de cachear tanto as operações de leitura de dados na memória RAM, quanto as operações de escrita.

O cache usado no 486, cacheia apenas as operações de leitura, o que permite ao processador ganhar tempo ao ler dados, mas não ajuda na hora de gravar dados, quando são perdidos vários ciclos até que a memória RAM torne-se disponível.



### Previsão de Desvio Dinâmico

Após concluída uma instrução do programa, para que o processador não perca um único ciclo de clock aguardando que o cache ou a memória RAM enviem a próxima instrução a ser processada, foi incluído no Pentium um buffer de pré extração. Este pequeno circuito armazena as próximas instruções a serem processadas, formando uma espécie de fila. Na verdade, o buffer de pré extração funciona como uma espécie de cache L0, ficando entre o processador e o cache L1.

A maior utilidade deste buffer é prever o resultado de operações de tomada de decisão. Se chega ao processador uma instrução como “Se  $X > Y$  então  $Z = K$ , senão  $Z = Q$ ” o buffer irá carregar tanto a instrução seguinte para  $X < Y$  quanto para  $X > Y$ , fazendo com que seja qual for o resultado da operação anterior, a próxima instrução já esteja carregada no buffer.

O buffer de pré extração também ajuda a memória cache a carregar antecipadamente os dados que o processador possa precisar. No caso do exemplo anterior, seriam carregados na memória cache tanto o valor de K quanto de Q.

### Coprocessador Aritmético Mais Rápido

O coprocessador aritmético do Pentium foi completamente remodelado. Foram alteradas algumas das instruções, que passaram a ser executadas muito mais rapidamente e, como o processador principal, o coprocessador do Pentium também passou a utilizar um pipeline para aumentar a velocidade de execução das instruções.

Somadas com as brutais melhorias na arquitetura, o coprocessador do Pentium tornou-se cerca de 5 vezes mais rápido do que o utilizado no 486, tornando o processador muito mais rápido em aplicativos que demandem um grande número de cálculos.

#### Arquitetura Superescalar

Mais um aperfeiçoamento do Pentium e um dos principais motivos de seu maior desempenho, é a adoção de uma arquitetura superescalar.

Internamente, o Pentium trabalha como dois processadores de 32 bits distintos (chamados de canaleta U e canaleta V), sendo capaz de processar duas instruções por ciclo de clock (uma em cada processador). Cada processador possui acesso total ao cache, à memória RAM, e aos demais componentes do micro. Foi incluída também, uma unidade de controle, com a função de comandar o funcionamento dos dois processadores e dividir as tarefas entre eles.

Teoricamente, o uso de dois processadores distintos dobraria o desempenho do Pentium, já que ao invés de uma, poderiam ser executadas duas instruções por ciclo de clock. Mas, na prática existem algumas limitações.

Se por exemplo, um programa ordena que o processador some 4 números,  $X + Y + W + K$ , o processador poderia no primeiro ciclo de clock usar a canaleta U para somar X e Y e a canaleta V para somar W, mas no segundo ciclo, haveria apenas mais um cálculo para ser executado, o resultado das duas somas. Neste caso, apenas uma das canaletas poderia ser usada; a segunda ficaria ociosa. No final das contas, houve um ganho de 33%, já que ao invés do processador demorar 3 ciclos para executar o cálculo, demorou apenas 2.

Caso a continuação do cálculo dependesse da conclusão do cálculo anterior, como em “ $(X + Y) \times 3$ ”, a segunda canaleta novamente não poderia ser usada, pois o processador teria primeiro que somar X e Y para depois multiplicar o resultado por 3. Neste caso, não haveria ganho algum, pois o processador demoraria os mesmos dois ciclos que seriam necessários com apenas uma canalização.

Em média, a segunda canalização permite um desempenho 30 ou 40% superior ao desempenho obtido com apenas uma canalização. Caso o software seja alterado e otimizado para rodar em um processador com duas canalizações, ordenando as instruções de modo a deixar a segunda canaleta ocupada durante a maior parte do tempo, podemos conseguir mais 10 ou 15% de desempenho, chegando a algo entre 40 e 50%, mas, jamais será possível conseguir o dobro de desempenho.

Isto pode ser notado por exemplo ao recompilar o kernel do Linux. É possível orientar o utilitário a otimizar o código para qualquer processador, de um 486 a um Pentium III. Esta otimização permite que o Linux utilize todos os recursos do processador, obtendo um desempenho bastante superior.

Otimizar um programa para rodar em um processador Pentium, não o torna incompatível com processadores com apenas uma canalização (como o 486), nem torna seu desempenho menor nestes processadores, já que as mesmas instruções serão executadas, apenas em ordem diferente.

### **Execução Especulativa**

Se o processador tivesse que processar uma instrução de tomada de decisão, como em “Se  $X > 0$  então  $Y = 30$ , senão  $Y = 20$ ”, enquanto a primeira canaleta do processador verifica se  $X$  é maior ou menor que 0, a segunda ao invés de ficar ociosa, pode executar uma das duas opções seguintes (atribuir o valor 30 ou atribuir o valor 20 a  $Y$ ). No próximo ciclo, quando a primeira canaleta tivesse terminado de processar sua instrução, teríamos 50% de chance da segunda canaleta ter adivinhado qual seria a instrução seguinte e já tê-la executado. O nome “execução especulativa” foi dado por que a segunda canaleta escolhe aleatoriamente a instrução a executar, entre as duas possíveis.

### **Acesso mais rápido à Memória**

O Pentium é capaz de acessar a memória usando palavras binárias de 64 bits, o dobro do 486, que acessa a 32 bits. Este recurso permite que sejam lidos 8 bytes por ciclo, ao invés de apenas 4, dobrando a velocidade de acesso à memória. Isto diminuiu bastante o antigo problema de lentidão das memórias, mas apenas provisoriamente, pois logo surgiram processadores Pentium utilizando multiplicadores de clock cada vez mais altos.

Como a maioria das placas para processadores Pentium utiliza módulos de memória de 72 vias, que são módulos de 32 bits, é preciso usa-los em pares. O processador acessa cada dupla como se fosse um único módulo, chegando aos 64 bits necessários.

Mesmo com a capacidade de acessar a memória a 64 bits e sendo composto internamente por dois processadores de 32 bits, o Pentium continua sendo um processador de 32 bits. Estes novos recursos servem apenas para melhorar o desempenho do processador.

### **Multiprocessamento**

Visando o mercado de Workstations (máquinas muito rápidas, destinadas a aplicações pesadas como processamento de imagens 3D ou vídeo) e servidores de rede, a Intel incluiu no Pentium o recurso de multiprocessamento simétrico, que permite o uso de dois processadores na mesma placa mãe. Neste caso, é preciso adquirir uma placa mãe especial, com encaixe para dois processadores e um chipset com suporte ao multiprocessamento.

Como a divisão das tarefas entre os dois processadores não é feita automaticamente pelo chipset, é preciso que o sistema operacional seja capaz de reconhecer os dois processadores e acessá-los individualmente, dividindo as tarefas entre eles da melhor maneira possível.

Caso o sistema operacional não ofereça suporte ao multiprocessamento, como é o caso do Windows 95 e do Windows 98, apenas um dos processadores será usado pelo sistema, ficando o outro inativo. Neste caso, será como se tivéssemos apenas um processador instalado. A maioria dos sistemas operacionais, incluindo o Windows NT, 2000 e XP, Linux e a maioria das versões do Unix suportam multiprocessamento, as exceções mais notáveis ficam por conta do Windows 95,98 e ME.

Apesar de, pela lógica, o desempenho dobrar com dois processadores trabalhando em paralelo, na prática o ganho dificilmente passa de 40 ou 60%, pois dificilmente será possível organizar a execução das tarefas de modo a deixar ambos os processadores ocupados todo o tempo, assim como é difícil manter as duas canaletas do Pentium cheias 100% do tempo.

Até pouco tempo atrás, o recurso de multiprocessamento foi exclusividade dos processadores Intel. Tanto o Pentium, quanto o MMX, o Pentium II e o Pentium III suportam o uso de até dois processadores simultaneamente, enquanto o Xeon suporta o uso de até quatro processadores (8 com a adição de um circuito especial na placa mãe). Com exceção do Athlon MP, todos os demais processadores da AMD e Cyrix não suportam multiprocessamento, o que não chega a ser uma grande desvantagem

para um usuário doméstico, já que fora do ramo dos servidores de alto desempenho, este recurso raramente é usado.

### **Clock e Overclock**

Ao contrário do que muitos pensam, velocidade de operação dos processadores não é fixa, mas sim determinada pela placa mãe.

Na placa mãe temos um pequeno cristal de Quartzo, chamado gerador de clock, que vibra alguns milhões de vezes por segundo, com uma precisão quase absoluta. As vibrações deste cristal são usadas para sincronizar os ciclos da placa mãe, que sabe que a cada vibração do cristal deve gerar um certo número de ciclos de processamento.

É mais ou menos como um farol, que abre e fecha algumas vezes por minuto. Quando o farol está fechado, o trânsito fica parado, voltando a fluir quando a farol abre. Um pulso de clock é justamente a abertura do farol, um “já!” que faz todos os periféricos trabalharem simultaneamente e de forma sincronizada. O funcionamento de todos os periféricos, da placa de vídeo ao disco rígido, é coordenado por este relógio.

O processador não possui um gerador de clock, e por isso trabalha usando o sinal recebido da placa mãe. Num Pentium MMX de 200 MHz, por exemplo, a placa mãe funciona a 66 MHz, e o multiplicador é 3x, o que significa que para cada ciclo da placa mãe, o processador gerará 3 ciclos.

Justamente por estar limitada à frequência indicada pela placa mãe, a frequência do processador não é fixa; pode ser maior ou menor do que o especificado, dependendo de como a placa mãe estiver configurada.

Como as placas mãe atuais, para manter compatibilidade com vários processadores podem operar a várias frequências diferentes, é possível fazer o processador trabalhar mais rápido simplesmente configurando a placa mãe para trabalhar a uma frequência maior. Esta técnica é chamada de Overclock, uma gíria que significa “acima do clock” numa tradução livre.

Um Pentium 120 por exemplo, usa bus 60 MHz e multiplicador de 2x. Se configurássemos a placa mãe para trabalhar a 66 MHz, mantendo o multiplicador em 2x, o processador passaria a trabalhar a 133 MHz. Se a frequência da placa mãe fosse aumentada para 75 MHz, o processador funcionaria a 150 MHz.

Em muitos casos, o processador também aceita um multiplicador maior. Um AMD K6 de 266 MHz por exemplo, trabalha com a placa mãe funcionando a 66 MHz e usando multiplicador de 4x. Se aumentássemos o multiplicador para 4.5x, mantendo a placa mãe funcionando a 66 MHz, faríamos o processador funcionar a 300 MHz.

A performance de um processador trabalhando em overclock é idêntica à de um processador “normal” funcionando a essa velocidade. Um Pentium 120 overclockado para 133 MHz por exemplo, apresenta exatamente a mesma performance de um Pentium 133 “de verdade”.

Quando um fabricante desenvolve um projeto de processador, testa-o a várias frequências diferentes, a fim de determinar sua frequência ideal de operação. Geralmente, os fabricantes adotam uma certa margem de segurança, vendendo o processador com uma frequência ligeiramente inferior à frequência máxima. É justamente esta margem de segurança que permite o overclock; estaríamos então simplesmente fazendo o processador funcionar na sua frequência máxima.

Esta margem muda de fabricante para fabricante e de processador para processador. Por isso, alguns processadores aceitam overlocks maiores que outros. Existem casos de processadores que aceitam trabalhar sem problemas a uma frequência 50% maior que a original, assim como existem casos de processadores que apresentam instabilidade operando a uma frequência apenas 10% maior que a original.

Obviamente, o overclock também traz algumas desvantagens. Fazendo o processador trabalhar a uma frequência maior do que a ideal, podemos ter problemas de travamentos, superaquecimento e sempre haverá alguma diminuição de sua vida útil.

## Outros Processadores

Além dos processadores “principais” que vimos até agora, existiram alguns modelos lançados como processadores de baixo custo, alternativa de upgrade para quem tinha um processador antigo e não queria gastar muito.

### 486DLC e 486SLC

Estes dois processadores foram a tentativa da Cyrix de entrar no mercado de processadores de baixo custo, oferecendo uma opção barata de upgrade para usuários de micros 386 e também uma opção de processador para micros de baixo custo, especialmente micros de menos de 1.000 dólares.

Estes processadores são basicamente processadores 386 (respectivamente o DX e o SX), que incorporam um pequeno cache L1 de apenas 1 KB. O cache não fazia milagres, mas já era o suficiente para aumentar um pouco o desempenho do processador, o que somado ao baixo preço de venda, foi suficiente para vários usuários investirem no upgrade, já que os 486DLC e SLC eram totalmente compatíveis com as placas para micros 386.

Vale lembrar que, como o 386 padrão, estes processadores não possuem coprocessador aritmético, podendo ser acoplados a eles o 387DCL ou o 387SLC, que deviam ser comprados separadamente.

Sob licença da Cyrix, a Texas Instruments desenvolveu versões próprias do 486DLC e SLC, preservando a mesma arquitetura, mas aumentando a quantidade de cache L1 para 8KB.

### AMD 5x86

No início, a AMD produzia clones de processadores Intel, utilizando os projetos desenvolvidos pela Intel e pagando royalties em troca. Porém, devido a várias divergências, a aliança acabou sendo desfeita e a AMD passou a batalhar seus próprios projetos de processadores. Apesar de, durante muito tempo, a AMD ter tido que se contentar com um distante segundo lugar, produzindo basicamente processadores de baixo custo, atualmente ela vem competindo diretamente com a Intel também no ramo de processadores de alto desempenho com seu Athlon, conseguindo na maioria das vezes manter preços mais baixos que a concorrente.

Mas, voltando à nossa aula de história, depois que a Intel lançou o 486DX4-100, abandonou o desenvolvimento de processadores 486 para se dedicar somente ao desenvolvimento do Pentium. Com a intenção de apresentar um processador que possuísse um desempenho semelhante a um Pentium low end (os modelos mais lentos e baratos), mas que ao mesmo tempo tivesse um preço competitivo, a AMD continuou o desenvolvimento do seu processador 486, lançando uma versão de 120 MHz (que opera usando barramento de 40 MHz e multiplicador de 3x), e logo em seguida também uma versão de 133 MHz.

Por questões de Marketing, a AMD batizou este 486 de 133 MHz de “AMD 5x86” o que confundiu alguns usuários, que pensaram tratar-se de um processador semelhante ao Pentium. O AMD 5x86 utiliza placas mãe para 486, necessita apenas que a placa seja capaz de sinalizar o multiplicador de 4x. O clock fica em 33 MHz, totalizando seus 133 MHz.

Como o AMD 5x86 não passa de um 486 funcionando a 133 MHz, seu desempenho é pouco menos de 33% superior a um 486DX4-100, sendo mais ou menos equivalente ao de um Pentium de 75 MHz. Aliás, outra medida de marketing tomada pela AMD na época, foi criar um índice Pr, ou “Pentium Rating”, comparando o desempenho do 5x86 ao do Pentium. O 5x86 de 133 MHz recebeu o índice Pr 75, indicando possuir um desempenho semelhante ao apresentado por um Pentium de 75 MHz.

A AMD conseguiu fazer um razoável sucesso com este processador, já que além de ser sido muito usado em micros de baixo custo, o 5x86 passou a ser uma alternativa barata de upgrade para usuários de micros 486 com processadores mais lentos.

### Cyrix Cx5x86

Além de desenvolver projetos de processadores 486, que foram fabricados pela Texas Instruments, a Cyrix lançou um processador que mistura recursos do 486 e do Pentium, oferecendo um desempenho bastante superior a um 486 padrão.

Este processador foi batizado como Cx5x86, e apresenta um cache L1 de 16 KB, além de algumas outras melhorias que tornam seu desempenho cerca de 35% superior ao de um 486 do mesmo clock. A versão de 100 MHz do Cx5x86 possui um desempenho equivalente ao 5x86 de 133 MHz da AMD e ao Pentium 75, enquanto a versão de 120 MHz rivaliza em desempenho com um Pentium 90.

Como o 5x86 da AMD, Cx5x86 é totalmente compatível com as placas mãe para 486, bastando configurar a placa com multiplicador de 3x e bus de 33 MHz para instalar a versão de 100 MHz e, 3x 40 MHz para utilizar a versão de 120 MHz.

### AMD K5

Depois de muitos atrasos, a AMD finalmente conseguiu lançar um processador que pudesse concorrer diretamente com o Pentium. O K5, porém, não chegou a tornar-se muito popular devido ao seu lançamento atrasado. Quando finalmente saíram as versões Pr 120 e Pr 133 do K5, a Intel já havia lançado as versões de 166 e 200 MHz do Pentium, ficando difícil a concorrência. Ao invés de simplesmente tentar copiar o projeto da Intel, a AMD optou por desenvolver um processador completamente novo, tecnicamente superior ao Pentium.

O K5 também utiliza uma arquitetura superescalar, mas ao invés de duas, possui quatro canalizações. O cache L1 também foi ampliado, passando a ser de 24 KB, dividido em dois blocos, um de 16 KB para instruções e outro de 8 KB para dados.

O coprocessador aritmético porém não foi muito melhorado, apresentando um desempenho quase 50% inferior ao apresentado pelo coprocessador do Pentium, devido principalmente à ausência de Pipeline. Este acabou sendo o calcanhar de Aquiles do K5, que a AMD sempre fez o possível para tentar esconder. Mas, como na maioria das aplicações o K5 era bem mais rápido que o Pentium, a AMD optou novamente por vender seu processador segundo um índice Pr, que compara seu desempenho com o dos processadores Pentium:

Processador	Frequência Real de Operação
<b>K5-Pr 120</b>	90 MHz (1.5x 60 MHz)
<b>K5-Pr 133</b>	100 MHz (1.5x 66 MHz)
<b>K5-Pr 166</b>	116 MHz (1.75x 66 MHz)

### Pentium Overdrive

Como fez com os antigos 386 SX, a Intel lançou (ou pelo menos tentou, pois este processador nunca chegou a ser muito vendido) também um Pentium "low cost". Este processador, apesar de internamente ter um funcionamento idêntico a um Pentium, utiliza placas mãe para processadores 486, sendo por isso chamando de Overdrive.

A Intel lançou o Overdrive em versões de 63 MHz (25 MHz x 2.5) e 83 MHz (33 MHz x 2.5) mas, por utilizarem placas de 486, que operam a frequências muito mais baixas e acessam a memória a apenas 32 bits, estes processadores perdem feio em performance se comparados com um Pentium "de verdade". O Overdrive de 63 MHz apresenta performance idêntica ao 486DX4-100, enquanto o de 83 MHz empata com o 5x86 de 133 MHz da AMD.

Além da baixa performance, o Overdrive era extremamente caro (por isso usei o low cost entre aspas no parágrafo anterior :-), e acabou sendo uma péssima opção de compra. Em termos de custo-benefício, o 5x86 da AMD foi uma opção muito melhor.

Mesmo após este primeiro fracasso, a Intel continuou tentando lançar sucessivamente vários processadores Overdrive, entre eles uma versão do MMX que funciona em placas soquete 7 antigas e uma versão do Pentium II que funciona em placas mãe para Pentium Pro. Apesar da propaganda feita por alguns "especialistas" nenhum destes modelos de Overdrive foi uma opção de compra que sequer

merecesse ser considerada, pois devido à baixa procura e conseqüentemente à baixa produção, sempre custaram muito mais caro do que processadores equivalentes.

A única vantagem de se utilizar um processador Overdrive seria continuar utilizando a mesma placa mãe, trocando apenas o processador. Porém, na maioria vezes, por causa da baixa produção o processador Overdrive custa mais caro que um processador normal mais uma placa mãe para ele, tornando-se uma opção lusitana de upgrade.

É bem possível que no futuro sejam lançados outros modelos de processadores Overdrive, mas não espere ver boas opções. Adianta muito pouco trocar apenas o processador, mantendo a mesma velha e ultrapassada placa mãe, a mesma quantidade de memória de RAM, o mesmo velho e lento disco rígido etc. e vale menos à pena ainda pagar caro por isso. Seria como comprar uma Brasília e trocar o motor, você vai continuar com o mesmo carro velho. Na maioria das vezes, é mais sensato gastar em mais memória RAM, num disco rígido maior e mais rápido, ou numa placa 3D, caso o micro seja utilizado para jogos.

### Um tal de 186

O 8086 foi o pai do 8088, que equipou os primeiros PCs. Depois vieram os processadores 286, que também conhecemos bem. Mas, entre as duas gerações, existiu um modelo intermediário, pouco conhecido, mas igualmente importante. O 80186 é uma evolução do 8086, que trouxe algumas instruções novas e um sistema de tolerância à falhas. Apesar de não ter sido usado como processador em micros PC, o 80186 tornou-se um componente bastante popular, sendo usado em controladores de HDs, controladores de interrupção entre vários outros periféricos. Nestas aplicações, o 80186 continua em uso até hoje.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---



## **A Plataforma Java**

Você já deve ter ouvido falar que Java é tanto uma linguagem quanto uma plataforma. Isso quer dizer que, além da linguagem, o programador Java também conta com um conjunto de APIs que facilitam o desenvolvimento ao oferecerem soluções para situações comuns de desenvolvimento.

Confira como a plataforma Java foi organizada para atender as necessidades de diferentes tipos de aplicações:

- Java SE: Representa a base do Java, sendo composta pelas APIs e bibliotecas básicas para possibilitar o desenvolvimento de aplicativos de linha de comando e desktop. Já ouviu falar em Swing? Sim, ele faz parte do Java SE (Java Standard Edition);
- Java ME: A Java Micro Edition é a plataforma voltada para dispositivos móveis e embarcados, com capacidade de processamento reduzida, como os utilizados na criação de produtos para a Internet das Coisas;
- Java EE: Voltada para o desenvolvimento de soluções web e corporativas, a Java Enterprise Edition é uma especificação que agrupa outras, sendo, por isso, conhecida como especificação “guarda-chuva”.

## **Conhecendo As Plataformas Java**

Com o objetivo de apresentar algumas das plataformas Java, este artigo aborda os conceitos de cada uma delas, onde podem ser utilizadas e quais os tipos de aplicação que podem ser desenvolvidas. Também serão apresentadas suas características e algumas especificações aplicadas e disponíveis para cada plataforma, descrevendo suas utilidades.

Este tema é útil para quem deseja conhecer e utilizar as plataformas Java EE, Java SE e JavaFX no desenvolvimento de aplicações web e desktop. É válido também para profissionais que desejam conhecer algumas das especificações que podem ser aplicadas nessas plataformas.

Diversas plataformas de desenvolvimento estão presentes fortemente no mercado de TI, como é o caso do Java. Ela foi lançada pela empresa Sun Microsystems em 1995 e atualmente está sob a responsabilidade da Oracle. Seu objetivo é fornecer ferramentas para o desenvolvimento de aplicativos que contemplam áreas como mobilidade, sistemas web, sistemas desktop, dentre outros.

A plataforma Java é composta atualmente por quatro edições para segmentos específicos de aplicações, a saber: Java EE, Java SE, Java ME e JavaFX. Uma característica forte e presente em todas elas é a contribuição significativa de seus usuários, que ajudam cada vez mais em seu processo de evolução e divulgação.

Além da colaboração da comunidade, uma das grandes vantagens do Java é ser multiplataforma, o que possibilita a execução de aplicativos em qualquer sistema operacional ou hardware, desde que o interpretador Java esteja instalado. Os programas desenvolvidos são emulados por meio de uma máquina Virtual, conhecida como JVM (Java Virtual Machine), que converte os bytecodes (codificação do programa) para uma linguagem que a máquina entenda.

Outra vantagem do Java é a oferta de um vasto conjunto de tecnologias (ou especificações) que estão disponíveis e ajudam a enriquecer cada vez mais as plataformas. Alguns conceitos dessas tecnologias serão abordados ao longo do artigo.

## **Java Enterprise Edition**

A Java EE, ou Java Platform, Enterprise Edition, é a plataforma que disponibiliza recursos para o desenvolvimento de aplicações corporativas voltadas para web e servidores de aplicação.

Java EE foi projetada para suportar sistemas de uso em larga escala, ou seja, para uma quantidade significativa de usuários, possibilitando o desenvolvimento de aplicações escaláveis, robustas e multicamadas. Toda essa estrutura incorpora características como segurança e confiabilidade, muitas vezes consideradas difíceis de serem implementadas. Com o objetivo de amenizar as dificuldades de

implementação dessas características, o Java EE fornece um conjunto de tecnologias que reduz significativamente o custo e a complexidade de desenvolvimento.

Em uma aplicação multicamadas, por exemplo, tem-se a estrutura do aplicativo separada em camadas, onde cada uma possui uma responsabilidade específica. Atualmente é possível classificar essa estrutura em duas partes: arquitetura lógica e arquitetura física.

A arquitetura física refere-se à infraestrutura sobre a qual o sistema ou aplicação é executado, que pode ser dividida em três camadas, a saber:

- Cliente (Client Machine): uma máquina cliente pode ser um dispositivo móvel ou um equipamento com algum dispositivo inteligente.

Entretanto, um Cliente não necessariamente é considerado uma máquina, podendo ser também softwares que acessam outras camadas ou que recebem notificações delas e, no contexto da Java EE, um exemplo pode ser o acesso ao servidor. Diversos tipos de softwares podem ser considerados clientes Java EE, desde um navegador web até uma aplicação que não é desenvolvida em Java;

- Servidor de aplicação (Application Server): é onde as aplicações web e corporativas são instaladas. Quando alguma regra de negócio é alterada na aplicação, para que os clientes tenham acesso a essa alteração, basta atualizar o sistema no servidor de aplicações. Feito isso, todos os usuários passam a ter acesso à nova versão.

Os servidores de aplicação são acessados pelos clientes através de uma conexão de rede. É o servidor de aplicação que controla a conexão com o banco de dados em função das requisições do cliente;

- Servidor de banco de dados (Database Server): é onde residem os registros da aplicação desenvolvida, por exemplo, informações de cadastros de usuários. Os dados são acessados somente pelo servidor de aplicação e não diretamente pela aplicação cliente.

Em relação à arquitetura lógica, a mesma é caracterizada pela organização dos componentes do sistema ou aplicação, podendo ser dividida em quatro camadas, a saber:

- Apresentação (Web Pages): representa a interface gráfica (botões, janelas, formulários) que provém opções que permitem ao usuário interagir com o aplicativo. Uma camada de apresentação mal projetada pode resultar em uma alta complexidade, ocasionando uma experiência frustrante ao usuário;

- Controladora (Web Tier): objetiva controlar as requisições realizadas pelos usuários. Suas principais tarefas ou funções são: coletar as informações dos usuários na camada de apresentação, gerar conteúdo dinâmico para cada cliente e manter o estado dos dados para a sessão de cada usuário.

Basicamente esta camada controla o fluxo de interação que ocorre na camada de apresentação, servindo como intermediária entre a camada de apresentação e a camada de negócios;

- Negócios (Business Tier): objetiva implementar toda a lógica de negócio da aplicação, desde funções simples como validação de dados de entrada (por exemplo, CPF, CNPJ), até funções mais complexas como o controle de estoque.

Essa camada encapsula as funcionalidades da aplicação em EJBs (classes de negócio) sem se preocupar com a interface com o usuário ou a camada de persistência;

- Persistência (EIS Tier): é onde são implementados os métodos responsáveis por buscar, remover, editar e inserir informações na base de dados. Não é o próprio mecanismo de persistência (banco de dados), mas sim um front-end que manipula o acesso a ele.

Para compreender melhor como funciona essa arquitetura observe a Figura 1 que exemplifica graficamente a estrutura de uma aplicação Java EE.



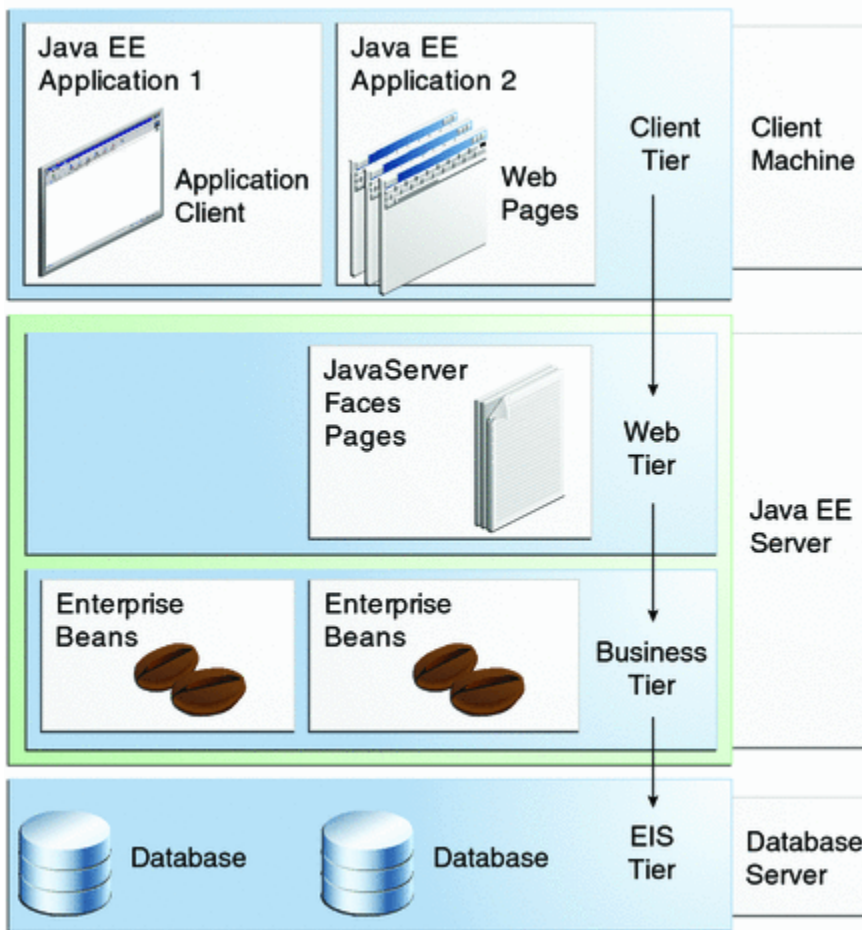


Figura 1. Estrutura de uma aplicação Java EE.

As quatro camadas lógicas interagem entre si através do seguinte fluxo: a camada de apresentação tem a função de interagir com o usuário e apresentar resultados. Ela dispara eventos gerados a partir das ações dos usuários (cliques ou seleções de menus) para a camada de controle. Esta gerencia todo o fluxo de informação recebida e ativa processos da camada de negócio. A camada de negócios, por sua vez, mantém toda a regra de negócios da aplicação, e de acordo com a ação recebida da camada de controle, executa a lógica necessária e se comunica com a camada de persistência, responsável pelo acesso às informações no banco de dados. Esta última camada, enfim, provê a criação, atualização, busca ou remoção de dados solicitados na camada de negócios, que são processados e devolvidos por meio da camada de controle à camada de apresentação.

### Tecnologias Aplicadas Na Camada De Apresentação

Na camada de apresentação estão contidos todos os componentes visuais (formulários, menus, botões, por exemplo) presentes na aplicação. Para a construção da interface gráfica, tecnologias como HTML, JavaScript, CSS e Ajax são utilizadas.

Essas tecnologias trabalham em conjunto com algumas especificações Java, a saber:

- **JavaServer Faces (JSF):** tecnologia para desenvolvimento web que utiliza um conjunto de componentes de interface gráfica baseados em eventos. O JSF busca aprimorar a construção de interfaces com o usuário ricas e amigáveis, possibilitando aos desenvolvedores trabalhar na camada de apresentação sem precisar conhecer a fundo, por exemplo, JavaScript, pois a tecnologia JSF oferece componentes prontos;
- **JavaServer Pages (JSP):** é uma linguagem de script que tem como objetivo a geração de conteúdo dinâmico para páginas da internet, pois permite adicionar código Java a páginas estáticas, como as páginas HTML.

Embora presentes na camada de apresentação, as tecnologias JSP e JSF também possuem recursos que permitem o seu uso na camada de controle.

Com o objetivo de enriquecer ainda mais o client-side de Java, foi criada a plataforma JavaFX, que vem conquistando o mercado, provendo uma nova e moderna coleção de controles de interface com o usuário. Esta coleção permite aos desenvolvedores construir aplicações RIA (Rich Internet Applications) que podem ser executadas em vários navegadores, desktops e dispositivos móveis. Mais informações sobre ela serão abordadas em seu respectivo tópico.

### **Tecnologias Aplicadas Na Camada De Controle**

Na camada de controle, que intermedia a interação entre a camada de apresentação e de negócios, o Java EE oferece tecnologias para atender as necessidades das tarefas descritas anteriormente. Dentre estas tecnologias podem-se citar alguns dos componentes fundamentais do Java EE, que são:

- **Servlets:** é um componente com o objetivo de receber requisições HTTP, processar e devolver respostas dinamicamente para a camada de apresentação, em formato de páginas HTML e XML, por exemplo;
- **Expression Language:** são tags padrões utilizadas dentro de páginas JSP e Facelets para referenciar componentes Java EE. As tags EL possuem algumas funções, a saber: possibilidade de invocar métodos estáticos e públicos, acesso dinâmico a dados armazenados em componentes JavaBeans e basicamente todas as operações de lógica e aritmética da linguagem Java (+, -, \*, /, < =, ||);
- **JavaServer Pages Standard Tag Lib:** São bibliotecas de tags que encapsulam funcionalidades comuns dentro de páginas, como controle de laços (for), controle de fluxos (do tipo if else), dentre outros. O JSTL permite escrever páginas JSPs sem código Java, usando somente tags.
- **Componentes JavaBeans:** são componentes de software que são projetados para serem reutilizáveis. Para uma classe ser considerada um JavaBeans, deve possuir as seguintes características: implementar a interface Serializable, possuir um construtor sem argumentos e conter métodos get() e set() para acesso aos campos. Apesar da semelhança nos nomes, os JavaBeans não devem ser confundidos com os Enterprise JavaBeans, ou EJBs, que são componentes utilizados em servidores.

### **Tecnologias Aplicadas Na Camada De Negócios**

A camada de negócio é responsável por implementar toda a lógica de negócios para o aplicativo. Nesta camada se concentra todo o código com as regras desenvolvidas para um caso específico, como um setor financeiro, um site de comércio eletrônico, entre outros. Em um aplicativo empresarial devidamente projetado, o núcleo da aplicação é implementado nos componentes da camada de negócio. As tecnologias que são oferecidas nesta camada são:

- **Enterprise JavaBeans (EJB):** é um modelo de componentes com foco em arquiteturas multicamadas, responsável por encapsular a lógica de negócios da aplicação. Fazendo uso de EJBs é possível construir aplicações escaláveis, transacionais, seguras e portáteis;
- **JAX-RS RESTful web services:** é uma API para criação de web services que usa a arquitetura REST (Representational State Transfer), sendo adotada no projeto de aplicações web que contam com recursos nomeados (URL, URI, URN). O objetivo da API é fornecer um conjunto de anotações para expor uma classe POJO como um serviço RESTful, a fim de simplificar o desenvolvimento de web services;
- **JAX-WS web service endpoints:** é uma API para criar e consumir web services SOAP. SOAP (Simple Object Access Protocol) é um protocolo para troca de informações estruturadas que se baseia em XML para seu formato de mensagem. Uma mensagem SOAP encapsula o conteúdo e pode ser trafegada via HTTP, JMS ou outro protocolo;
- **Java Persistence API (JPA):** é uma API que possibilita aos desenvolvedores gerenciar os dados dos usuários utilizando o mapeamento relacional de objetos (ORM). Tipicamente, uma entidade (classe)

representa uma tabela em um banco de dados relacional, e cada instância dessa entidade corresponde a uma linha (registro ou tupla) nessa tabela. A Java Persistence API possibilita criar essas entidades, modificá-las e removê-las do banco de dados. Os dados encapsulados pelos campos persistidos por essa API podem ser consultados usando, por exemplo, JPQL (Java Persistence Query Language), uma linguagem similar à SQL;

- Java EE managed beans: são componentes que podem prover a lógica de negócio da aplicação, mas não exigem os recursos transacionais ou de segurança dos EJBs.

**Tecnologias Aplicadas Na Camada De Persistência**

A terceira camada consiste em receber requisições da camada de negócio e executar essas requisições na base de dados. A seguir, são demonstradas algumas tecnologias que podem ser aplicadas para comunicação com o banco de dados a partir de aplicações Java EE:

- Java Database Connectivity API (JDBC): é um conjunto de APIs de baixo nível que tem como função estabelecer uma conexão, realizar consultas, inserção, remoção e atualização de dados, por meio de instruções SQL, em uma base de dados;
- Java EE Connector Architecture (JCA): é uma API para conexão de um servidor de aplicação com sistemas legados, geralmente sistemas que não evoluíram com o passar do tempo;
- Java Transaction API (JTA): API para definição e gerenciamento de transações. Por meio de interfaces é possível padronizar o uso de transações distribuídas, feitas por aplicativos Java. Basicamente é usada para verificar se uma transação é confirmada (commit), ou não (rollback). Um exemplo do uso de JTA é quando se necessita acessar dois SGBDs;
- Java Persistence API (JPA): A JPA, além de fazer parte da camada de negócios, como descrito anteriormente, também faz parte da camada de persistência.

Algumas das tecnologias supracitadas continuam sendo evoluídas nas novas versões das plataformas. Atualmente, a Java EE, por exemplo, está na versão 7, lançada oficialmente em 2013. Alguns dos principais marcos da evolução da Java EE estão ilustrados na Figura 2.

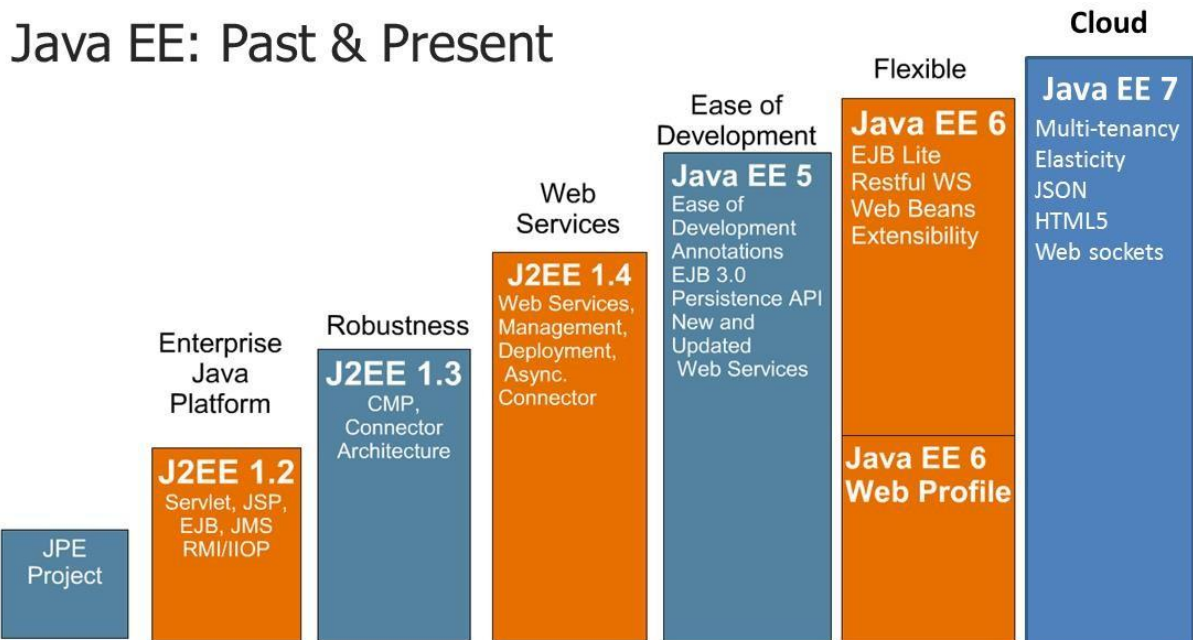


Figura 2. Evolução da plataforma Java EE.

A Java EE 7 introduziu um conjunto de novas APIs, bem como melhorou algumas das APIs já existentes. Seu maior objetivo foi facilitar a implementação e execução de aplicações em ambientes na nuvem, sejam elas públicas e/ou privadas. Um dos aspectos mais importantes do crescimento do Java é seu foco, que busca sempre acompanhar tendências com as novidades presentes em cada

nova versão de plataforma. As tecnologias e suas respectivas versões, que compõem essa plataforma, estão ilustradas na Figura 3.

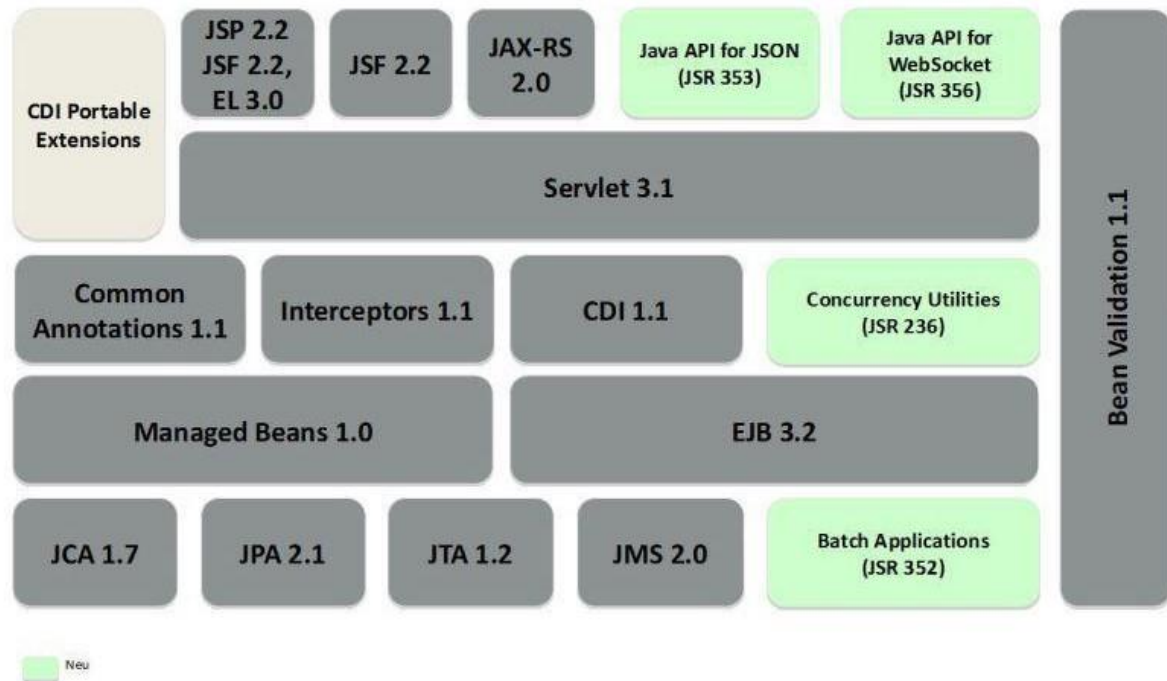


Figura 3. Especificações da Java EE 7.

Dentre as diversas melhorias projetadas para a Java EE 7, pode-se citar a tecnologia Bean Validation. Com ela é possível realizar validações no código fazendo uso de anotações. Bean Validation permite que a mesma validação de um campo obrigatório aplicada na camada de apresentação, por exemplo, possa ser utilizada nas demais camadas. Esta regra em todas as camadas fornece diversos benefícios ao desenvolvedor, gerando produtividade e evitando possíveis falhas.

### Java SE

A Java SE (Java Platform, Standard Edition) é a plataforma de programação voltada para criação de applets e desenvolvimento de softwares para desktop, destinados a computadores pessoais, notebooks ou outras arquiteturas com maior capacidade de processamento e memória.

Os aplicativos podem ser executados em Windows, Mac OS, Linux, Solaris ou outros sistemas operacionais, contanto que estes tenham instalado o ambiente de execução JRE (Java Runtime Environment).

### Tecnologias Aplicadas À Java SE

A plataforma Java SE fornece um conjunto de tecnologias para a criação de aplicações, sejam elas desktop ou não. Dentre as que ela oferece, são citadas algumas a seguir:

- Java 3D: API que oferece um conjunto de interfaces que possibilitam a construção e controle de gráficos 3D. Com esta API é possível incorporar gráficos de alta qualidade a aplicativos e applets;
- >Java Sound: fornece uma API para operações de áudio, como reprodução, captura (gravação) e mixagem;
- Java Advanced Imaging (JAI): provê um conjunto de interfaces que suportam um modelo de programação simples e de alto nível para manipulação de imagens;

- AWT (Abstract Window Toolkit): é uma API cuja função é fornecer suporte à criação de janelas, botões e outros elementos visuais usando métodos nativos. Ou seja, ela faz uso do ambiente gráfico do sistema operacional onde o software desenvolvido está sendo executado. Por exemplo, se você usa o Windows, toda a interface do seu aplicativo ficará com a aparência do Windows;
- JDesktop Integration Components (JDIC): é um projeto que fornece acesso a funcionalidades e facilidades oferecidas pelo ambiente nativo onde a aplicação é executada. Por exemplo, uma aplicação sendo executada em um ambiente desktop poderá abrir uma página web usando o navegador padrão definido pelo usuário em seu perfil na máquina. O JDIC suporta uma grande variedade de recursos, como criar ícones na área de trabalho do usuário, acesso ao editor de e-mails, dentre outros.

## JavaFX

JavaFX é uma plataforma projetada para oferecer um rico conjunto de APIs que simplificam o desenvolvimento de softwares que utilizam conteúdo de multimídia. Para que isso seja possível, são utilizados avançados engines de mídia e gráficos de aceleração por hardware.

Essa tecnologia provê uma nova e moderna coleção de controles de interface com o usuário, tornando-se uma ótima opção para o desenvolvimento RIA com Java. As aplicações RIA (Rich Internet Applications) podem ser acessadas de qualquer computador que esteja conectado a internet. O termo RIA é usado para descrever Aplicações Ricas para Internet, que são executadas em ambiente web, mas que possuem características similares a softwares desenvolvidos para execução em ambiente desktop.

Investir nesta plataforma pode significar muitas vantagens para os desenvolvedores e empresas que já trabalham, principalmente, com Java. Por ser baseada nesta linguagem, é possível utilizar todos os seus poderosos recursos, como multithreading e suporte a anotações.

Outra vantagem é que a tecnologia utiliza FXML, uma linguagem de marcação como opção à codificação em Java. FXML permite modelar a UI separadamente da lógica da aplicação, não sendo necessário recompilar o código toda vez que uma mudança no layout ocorrer. Com essa tecnologia é possível criar tabelas, botões, aplicar efeitos em campos, criar animações, fazer uso de CSS, exibir conteúdo multimídia, dentre muitas outras funcionalidades que contribuem para fornecer ao usuário uma melhor usabilidade. Na Figura 4 é exibida a tela de uma aplicação disponibilizada pela Oracle com exemplos de aplicações desenvolvidas em JavaFX.



Figura 4. Aplicação com exemplos do JavaFX.

Para incrementar ainda mais a plataforma JavaFX, algumas tecnologias e projetos vêm contribuindo para o desenvolvimento de aplicações desse tipo. São eles:

- e(fx) eclipse: plugin do Eclipse para a construção de aplicações JavaFX;
- FxForm2: API para criação de formulários utilizando JavaBeans. FXForm2 usa a API de Bean Validation para validação dos dados;
- DataFX: Atualmente, muitos dos web services têm seus dados disponibilizados em formatos como XML e/ou JSON. O objetivo do projeto DataFX é facilitar a leitura desses dados usando ListViews, TableViews e TreeViews de uma forma mais simples, produtiva e sem muitas linhas de código;
- JFXtras: projeto que tem como objetivo oferecer diversos componentes extras aos desenvolvedores, enriquecendo ainda mais a interface gráfica das aplicações e, conseqüentemente, sua usabilidade.

### **Conclusão**

Neste artigo foi possível conhecer as principais plataformas Java (Java SE, Java EE e JavaFX) e abordar diversos conceitos de tecnologias que podem ser aplicadas especificamente em cada uma delas.

A edição Java SE é considerada a base da plataforma Java, sendo voltada, principalmente, para a criação de software para ambiente desktop. Como é a base da plataforma, Java EE estende Java SE, e é usada para a construção de aplicações corporativas e web. JavaFX, por sua vez, tem por objetivo focar no desenvolvimento de aplicações multimídia, proporcionando uma interface visual muito mais agradável que enriquece o desenvolvimento client-side da plataforma Java.

Para o futuro, espera-se que sejam implementadas ainda mais tecnologias que venham a fortalecer as plataformas analisadas, buscando uma melhora contínua para tornar o processo de desenvolvimento de aplicações Java ainda mais produtivo.

### **Máquina Virtual Java**

Na máquina virtual Java, ou JVM, é onde a sua aplicação será executada. É ela, também, a responsável pela característica multiplataforma do Java. Um programa escrito nessa linguagem será executado em qualquer plataforma, seja ela um notebook, smartphone ou torradeira, que possua uma máquina virtual Java implementada.

Um dos recursos mais conhecidos da JVM é o Garbage Collection. É ele que é acionado com certa frequência para limpar da memória objetos que não estão sendo utilizados, evitando desperdício de espaço e que sua aplicação deixe de funcionar por falta dela.

Algumas vezes, a depender da proposta de sua aplicação, pode ser necessário fazer customizações na JVM visando melhoria na performance.

### **Primeiros Passos**

Assim como em qualquer outra linguagem, no Java você encontrará tópicos de conhecimento fundamental. Dominá-los permite programar um código mais limpo e fácil de ser compreendido.

Diferente de outras linguagens, no Java o seu primeiro código será escrito dentro de um método de uma classe.

Após aprender sobre classes você deve estar curioso para saber como declarar variáveis e realizar operações com elas.

No Java as enums são um poderoso recurso para a criação de objetos imutáveis.

Você já ouviu falar que tudo no Java são objetos? Se ainda não, não tem problema.

Agora que você já conheceu os componentes básicos de um programa Java, que tal olhar um pouco mais de perto a criação de uma classe?

## **Strings**

A manipulação de texto é algo bastante comum no desenvolvimento de aplicações. Em Java, a estrutura principal para isso é a classe String, que nos fornece vários métodos para sua manipulação. Nos posts abaixo você encontrará conteúdos que abordam desde os métodos básicos, até outras opções para que possa manipular texto da maneira mais adequada para sua solução.

Além da classe String, temos também as classes StringBuilder e StringBuffer.

Algumas vezes precisamos de uma ferramenta mais específica para processar Strings. É nesse momento que as expressões regulares podem ajudar.

## **Estruturas De Condição E Repetição No Java**

Duas ferramentas importantes e presentes na lógica de programação de qualquer código são as estruturas de condição e repetição. A estrutura de condição nos permite criar um bloco de código a ser executado apenas se determinada condição for atendida. Já a estrutura de repetição nos permite criar um bloco de código que pode ser executado várias vezes, enquanto a condição especificada for verdadeira.

## **Entrada E Saída De Dados**

Saber apresentar dados na tela e ler dados do teclado é um passo fundamental no aprendizado da programação. Lembre-se que o principal objetivo de um programa é processar os dados do usuário e exibir os resultados. Para isso temos os recursos de Entrada/Saída, ou In/Out da linguagem.

## **Módulos Do Código Java**

Saber como organizar o código é fundamental para sua manutenção e reutilização. Em Java, o código é organizado em pacotes, classes e métodos. Dominados esses conceitos, e utilizando modificadores de acesso, podemos controlar como cada classe ou método poderá ser reaproveitado pelos demais.

## **Estrutura De Dados**

Ao começar a desenvolver sistemas um pouco mais avançados, você se deparará com a necessidade de organizar os dados que são manipulados no código. Para isso, é comum o uso de estruturas de dados, que, como o nome indica, fornecem uma forma padrão para agrupar e simplificar a gerência de dados com características em comum.

## **Tratamento De Exceções**

É comum a ocorrência de exceções durante a execução de um programa. O que não é comum é não nos prepararmos para elas. Uma exceção é um evento diferente daquilo que normalmente o software espera que aconteça. Quando não tratamos exceções, o software fica sem saber o que fazer com ela e como continuar sua execução, o que normalmente leva à paralisação do sistema.

Todo programa pode e vai falhar em algum momento. Aplicações consistentes se preparam adequadamente para esse momento.

## **Serialização E Fluxo De Dados**

Logo você poderá se perguntar: Como salvar os dados que utilizo em minha aplicação para que consiga utilizá-los posteriormente? A resposta é simples: Serialização. Quando serializamos um objeto, estamos o transformando em um array de bytes, formato utilizado para salvar arquivos.

## **Sockets**

Caso seu interesse seja aprender como enviar e receber dados e arquivos pela rede, saber programar com Sockets é fundamental. Compreender sockets é a primeira etapa para iniciar na programação distribuída.

## Threads

Com a multiplicação do número de núcleos presentes em um processador, saber como programar com threads é algo de grande relevância para fornecer sistemas com alto desempenho, que explorem ao máximo o poder computacional de servidores e desktops.

## Collections

Neste momento é bem provável que você já tenha tido o primeiro contato com as coleções, afinal, uma coleção é uma estrutura de dados. Mas, por que só agora falar delas? É importante compreender os fundamentos por trás de um conceito antes de começar a utilizar algo mais avançado, com um nível mais alto de abstração.

De forma simples, uma coleção é uma lista de objetos. Como quando lidamos com dados é comum nos depararmos com diferentes necessidades de organização dos mesmos, o Java facilita esse trabalho nos fornecendo diferentes opções para mantê-los - de listas que se preocupam apenas em armazenar os dados em memória, a listas que evitam dados repetidos e os mantém ordenados.

## Generics

Generics é uma importante funcionalidade da linguagem Java que foi criada para facilitar o reuso do código e tornar a implementação mais segura e menos propensa a erros. Generics são bastante utilizados quando lidamos com coleções, por exemplo, e sua declaração é feita com os caracteres '<>'. Assim, quando declaramos uma lista, podemos especificar que essa lista somente poderá receber objetos de um tipo.

Com isso, caso o programador tente inserir um objeto de tipo diferente, o erro será identificado em tempo de compilação, e não mais em tempo de execução.

## Expressões Lambda

Um dos recursos mais jovens da linguagem Java agrega um toque de programação funcional a ela. Uma definição simples e um pouco abstrata é entender que uma expressão lambda é uma opção para programar uma função sem que para isso seja necessário declarar um nome e um tipo de retorno. Em geral, a expressão lambda é declarada no mesmo lugar em que será utilizada e possibilita um código menor e mais simples de manter.

A sintaxe de uma expressão lambda é bastante simples. Começamos declarando os parâmetros. Em seguida utilizamos o operador arrow.

Por fim, declaramos o bloco de código a ser executado.

(Zero ou mais parâmetros) -> { /\* bloco de código \*/ }

## Streams

Com o intuito de aprimorar a forma como manipulamos coleções de dados, o Java 8 trouxe a Streams API, conjunto de classes e interfaces que faz uso de conceitos da programação funcional e das expressões lambda para que escrevamos código mais claro e em menor quantidade.

Para isso, obtemos uma stream, a partir de uma coleção, por exemplo, e, sem nos preocuparmos com a forma como os dados serão percorridos, programamos algum tipo de processamento sobre os dados, como filtros, mapeamentos, entre outros, já fornecidos pela API.

Como exemplo, considere que temos uma lista de produtos e desejamos selecionar apenas aqueles cujo valor seja maior do que 5. O código para solucionar esse problema poderia ser:

```
List<produto> produtosFiltrados = produtos
```

```
.stream()
```

```
.filter((p) -> p.getValor() > 5)
```



```
.collect(Collectors.toList());
```

## Reflection

O Java também dispõe de um recurso conhecido como Reflection. Com ele, em tempo de execução conseguimos acessar informações de uma classe, como os atributos e métodos, assim como instanciar essa classe e invocar um método. Isso nos permite estender funcionalidades de uma aplicação, por exemplo.

Mas antes de prosseguir, sem querer dar spoiler, você vai precisar de um pouco de familiaridade com as anotações em Java para entender esse poderoso conceito na prática.

Reflection também é muito utilizado por IDEs, bibliotecas e frameworks, os quais precisam lidar com classes criadas por terceiros. Você já imaginou como aquela biblioteca que faz o parsing de objetos para JSON e vice-versa funciona? Como ela consegue converter qualquer objeto para JSON e o JSON para seu respectivo objeto?

## Orientação A Objetos

A Orientação a Objetos é o paradigma de programação mais utilizado para o desenvolvimento de sistemas e tem como principal característica o planejamento e implementação do software a partir da representação de 'coisas' da vida real por meio de objetos.

Antes desse paradigma era comum criarmos sistemas utilizando o paradigma de programação estruturada. Porém, devido a limitações, aos poucos essa opção começou a ser substituída.

## Pilares Da Orientação A Objetos

Ao planejar a programação baseada em objetos, esse paradigma trouxe consigo uma série de conceitos, os quais levam a um código mais fácil de entender, manter e reutilizar.

O encapsulamento é o pilar da Orientação a Objetos que nos permite ocultar informações e, ao mesmo tempo, atribuir segurança ao código. Quando encapsulamos o acesso a uma variável, por exemplo, não importa para o objeto que consome essa informação (o cliente), se o valor é obtido lendo o dado em memória ou se foi necessário acessar um banco de dados.

Imagine que você esteja em um restaurante e faça um pedido. É importante para você saber qual panela está sendo utilizada, a ordem de preparo dos nutrientes, entre outras coisas? Então, encapsular é a "arte" de deixar acessível ao cliente apenas aquilo que ele precisa saber.

E a herança? Como assim programar uma herança? Você herdou características de seus pais, avós e de outras gerações passadas, não é mesmo? Na Orientação a Objetos, quando precisamos fazer com que a classe B possua características de outra, definimos via código que a classe B herda dessa outra. Interessante, não? :) Ainda está achando um pouco confuso?

O polimorfismo é um pilar da Orientação a Objetos que tem forte ligação com a herança. Algumas vezes, quando herdamos uma característica, precisamos adicionar a ela um comportamento diferente. Por exemplo: todo animal sabe correr, porém, cada um corre de uma maneira diferente.

## Coesão E Acoplamento

Você já deve ter ouvido falar que um bom código precisa ter alta coesão e baixo acoplamento. No entanto, o que é a coesão e o acoplamento? A coesão é uma medida que indica quão bem delimitado está cada módulo do nosso código. Caso ele faça mais coisas do que deveria, dizemos que temos um módulo com baixa coesão. Já o acoplamento mede o grau de dependência de um módulo em relação aos demais. Caso ele dependa de muitos módulos, dizemos que ele está com acoplamento alto. Essas situações devem ser evitadas, pois tornam o código muito mais difícil de manter e evoluir.

## Princípios SOLID

Após compreender os pilares da Orientação a Objetos na teoria e na prática, um importante passo na busca pelo código de qualidade é aprender sobre os princípios SOLID.

Esses princípios nos ajudam a pensar no problema a ser modelado de forma orientada a objetos, propondo definições que nos levam a um design de código coeso e com baixo acoplamento, o que o torna fácil de ler, manter e reutilizar.

E por que esse nome, por que SOLID? Esse termo é um acrônimo para as cinco "regras" que o compõem:

- S - Single Responsibility Principle - O Princípio da Responsabilidade Única define que cada classe deve ter apenas uma responsabilidade;
- O - Open Closed Principle - O Princípio do Aberto Fechado define que uma classe deve ser extensível (aberta para extensões) por outras classes, sem que para isso precise ser modificada (fechada para modificações);
- L - Liskov Substitution Principle - O Princípio da Substituição de Liskov está relacionado ao uso da herança e dita que devemos ser capazes substituir a classe filha pela classe pai sem que o funcionamento do software seja prejudicado;
- I - Interface Segregation Principle - O Princípio da Segregação de Interfaces preza pela correta definição de interfaces, de forma que os clientes que a consomem tenham acesso apenas àquilo que de fato é necessário;
- D - Dependency Inversion Principle - O Princípio da Inversão de Dependências sinaliza que devemos depender apenas de classes abstratas, e não de classes concretas;

### **Orientação A Objetos Na Prática**

Agora que você já conhece os pilares da Orientação a Objetos, que tal pôr as mãos na massa? Pode dizer! Já estava na hora, não é mesmo? :)

Para isso reunimos uma seleção de artigos que abordam esse tema na prática.

Como apoio a esse conteúdo, não deixe de verificar os posts abaixo e aprenda como solucionar e/ou evitar os erros mais cometidos quanto adotamos o paradigma OO.

### **Boas Práticas**

Até o momento você aprendeu as principais tecnologias que compõem o Java, bem como utilizá-las de forma consistente. Que tal agora colocar em prática esses conhecimentos enquanto aprende técnicas avançadas no desenvolvimento com Java?

Lembre-se que o aprendizado da linguagem Java precisa ser constante. Somente programando você conseguirá implementar códigos melhores.

Ao prosseguir com a busca pelo código de qualidade, em algum momento você se deparará com o termo Clean Code. Mas, o que é isso? O Clean Code, ou Código Limpo, reúne uma série de práticas a serem aplicadas para obter um código fácil de compreender e evoluir.

A qualidade do código também influencia na performance.

---

---

---

---

---

---

---

---

---

---

## Programação PHP

Para começarmos a programar em PHP primeiro abra o seu editor de texto ou IDE preferido, analise entre as melhores IDE para PHP e tabela comparativo entre as melhores IDE para PHP ambos em inglês. Uma boa saída é o Aptana e o plugin para desenvolvimento em PHP mais você pode utilizar o seu preferido.

### Delimitadores De Código PHP

Assim como o HTML temos as tags no PHP temos os delimitadores de código, que são `<?php` e `?>` respectivamente a tag de abertura e a tag de fechamento, em que o código deve ser inserido. Veja o exemplo a seguir:

1	<code>&lt;?php</code>
2	
3	<code>// Código PHP aqui!!!</code>
4	
5	<code>?&gt;</code>

Além dos delimitadores de código `<?php` e `?>` todas as suas páginas devem possuir a extensão `.php` por exemplo: `meu-primeiro-programa.php`

Ou ainda como mencionado no artigo anterior sobre o que é o PHP podemos **misturar o PHP com o HTML** como pode ser observado no exemplo a seguir.

01	<code>&lt;?php</code>
02	
03	<code>// Sim também podemos ter código PHP antes do DocType.</code>
04	
05	<code>?&gt;</code>
06	<code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"&gt;</code>
07	<code>&lt;html xmlns="http://www.w3.org/1999/xhtml"&gt;</code>
08	<code>&lt;head&gt;</code>
09	<code>&lt;meta http-equiv="Content-Type" content="text/html; charset=utf-8" /&gt;</code>
10	<code>&lt;title&gt;Titulo da minha página&lt;/title&gt;</code>
11	<code>&lt;?php</code>
12	
13	<code>// Código PHP aqui dentro do head.</code>
14	
15	<code>?&gt;</code>
16	<code>&lt;/head&gt;</code>

17	
18	<body>
19	<?php
20	
21	// Código PHP aqui dentro do body!!!
22	
23	?>
24	</body>
25	</html>
26	<?php
27	
28	// Código PHP aqui até depois de fecharmos a tag html!
29	
30	?>

Como você pode perceber podemos escrever código PHP misturado com o HTML em qualquer parte do código. Além do exemplo que eu mostrei imagine que você queira gerar o título da sua página pelo PHP dinamicamente o que você faria? Isso mesmo colocaria as tags do PHP dentro da tag title e realizaria uma rotina para gerar o título dinamicamente.

Caso você já tenha programado algo em PHP ou baixado algum script em PHP talvez já tenha se deparado com as tags respectivamente de abertura e fechamento assim:

1. <? ?>
2. <% %>
3. <script language="PHP"> </script>

O mais utilizado é o primeiro caso, no entanto **não o utilize**. Pois com o avanço do PHP este tipo de escrita pode ser desabilitado e ainda pode ocorrer a possibilidade de seu servidor web não aceitar este tipo de tag.

Por isso prefira sempre as tags <?php e ?>.

### Conceito De Variável

Variáveis como você já deve estar imaginando apenas pelo nome é tudo aquilo que é sujeito a **variações**, que é **incerto**, **instável** ou **inconstante**, ou seja, pode **mudar durante sua existência**. Vamos a uma analogia para entendermos o conceito de variáveis. Imagine a seguinte situação: Joãozinho mora na casa de seus pais desde que nasceu e passa a maior parte do dia em seu quarto.

Com o passar dos anos o quarto de Joãozinho foi mudando, seus brinquedos foram trocados por um computador, seu berço foi trocado por uma cama, seu guarda-roupa de criança foi trocado por um de adulto, as paredes foram pintadas de cores diferentes e Joãozinho ganhou uma televisão em seu quarto. Como você já deve ter imaginado a nossa variável aqui foi o quarto de Joãozinho, que com o passar do tempo foi variando seu conteúdo.

Trazendo o conceito de variável ao PHP e a nossas páginas de internet. Voltemos ao exemplo anterior em que misturamos PHP e HTML. O conteúdo que ficaria dentro da tag body gerado pelo PHP seria uma variável, que dependo da página que estaríamos acessando este valor, ou seja, ele seria variável. A página inicial teria um valor naquela variável e página de contato já teria um valor diferente em sua variável.

### As Variáveis No PHP

Agora que você já possui um conceito de variável iremos ver como criar nossas variáveis no PHP. As variáveis servem para armazenar dados que podem ser utilizado em qualquer parte do programa. Para criar uma variável em PHP você deve seguir algumas regras, são todas bem fáceis de se acostumar, veja a seguir:

1. As variáveis são representadas por um cifrão (\$) seguido pelo nome da variável
2. O PHP é case sensitive, ou seja, ele diferencia minúsculas de maiúsculas. Sendo então \$nome diferente de \$Nome e \$NoMe
3. Nomes de variáveis devem ser iniciados apenas com uma letra ou \_ (sublinhado) e podem ser seguido de letras ou algarismos, ou seja, números podem aparecer em outras posições exceto na primeira
4. Nunca utilize caracteres especiais em nome de uma variável como acentos (é í ó) cedilha (ç)

### Exemplo De Variáveis Validas No PHP

Veja a seguir exemplos de nomes válidos para variáveis, baseadas nas regras que estudamos anteriormente.

1	<?php
2	
3	\$nome = 'Mauro George';
4	\$nascimento = '06/09/1989';
5	\$sobre_nome = 'Oliveira Tavares';
6	
7	?>

### Exemplo de variáveis invalidas no PHP

Agora alguns nomes de variáveis invalidas no PHP que **não** devem ser utilizadas.

1	<?php
2	
3	\$20_anos_e_nome = 'Mauro George';
4	\$ nascimento = '06/09/1989';
5	\$*este_é_meu_sobrenome = 'Oliveira Tavares';
6	
7	?>

Como você já deve ter imaginado o sinal de = (igual) é utilizado para atribuir um valor a variável, aprenderemos mais sobre eles quando chegarmos em operadores de atribuição.

### Separação De Instruções

Você também deve ter reparado no exemplo anterior que depois definirmos um valor a variável terminamos utilizando o sinal de ; (ponto e vírgula). Este é o separador de instruções do PHP, ou seja, sempre que definirmos uma variável ou imprimirmos algo na tela devemos utilizar o ;.

Enviando resultados ao browser, o uso do comando echo

Você deve estar imaginando acabei de criar varias variáveis mais como que eu faço para exibir o conteúdo que guardei em cada uma delas? E neste ponto que aprenderemos exibir o valor de nossas variáveis na tela. Mais primeiro exibiremos o mais clichê de todas as linguagens de programação o famoso "Hello World".

Aconselho a partir deste ponto criar uma pasta com o nome de **estudos** em seu diretório HTDOCS e vá testando os exemplos que serão mostrados a seguir.

Não se esquecendo de salvar os arquivos com a extensão .php

1	<?php
2	
3	echo 'Hello World';
4	
5	?>

Repare que ainda não exibimos os dados de uma variável apenas exibimos um texto comum.

### Exibindo Dados Das Variáveis No PHP

Agora iremos exibir os valores de nossas variáveis no browser utilizando o PHP, para isso utilizaremos as variáveis que criamos anteriormente.

01	<?php
02	
03	\$nome = 'Mauro George';
04	\$nascimento = '06/09/1989';
05	\$sobre_nome = 'Oliveira Tavares';
06	
07	
08	echo \$nome;
09	echo \$sobre_nome;
10	echo \$nascimento;
11	

12	?>
----	----

Você deve ter percebido que como exibimos varias variáveis quando terminou o dado de uma ele “co-lou” com o dado da próxima variável mais isto será concertado quando chegarmos a concatenação de strings, mais neste ponto o interessante é vermos que conseguimos exibir o valor da variável.

### Posso Exibir Dados Das Variáveis No PHP Misturado Com O HTML?

SIM! Como o PHP se mistura ao HTML podemos exibir os dados em conjuntos vejam um exemplo pratico.

01	<?php
02	
03	\$titulo = 'Aqui vai o titulo da minha página';
04	\$css = '<link rel="stylesheet" type="text/css" href="css/estilos.css" />';
05	\$conteudo = 'Aqui é o conteudo mais como não tenho nenhum...   vai apenas um   <strong>Hello World</strong>';
06	
07	?>
08	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
09	<html xmlns="http://www.w3.org/1999/xhtml">
10	<head>
11	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
12	<title><?php echo \$titulo; ?></title>
13	
14	<?php
15	echo \$css;
16	?>
17	</head>
18	
19	<body>
20	<?php
21	
22	echo \$conteudo;
23	
24	?>

25	
26	<p>Eu posso repetir o valor da variavel sempre que eu quiser sabia? Veja aqui o nosso titulo denovo "<?php echo \$titulo; ?>"</p>
27	<p>E não necessariamente deveria imprimir \$titulo apenas porque eu a declarei primeiro poderia imprimir \$css antes e depois \$titulo ou qualquer uma variavel isso vai de acordo com sua necessidade</p>
28	<p>Agora irei exibir \$conteudo novamente olhe ela ai</p>
29	<p><?php echo \$conteudo; ?></p>
30	
31	</body>
32	</html>

Repare que além do PHP esta misturado ao HTML também a tags de HTML no valor da minha variável em PHP.

As variáveis podem ser exibidas sempre que necessário na tela, veja que repeti algumas variáveis mais de uma vez.

E ainda independente da ordem de declaração das variáveis elas podem ser chamadas a sua escolha. Mesmo eu tendo declarado \$titulo, \$css e \$conteudo eu poderia exibir na tela primeiro \$conteudo e \$css e por ultimo exibir \$titulo a ordem de exibição sou eu que faço.

No entanto o seguinte exemplo geraria um erro:

01	<?php
02	
03	echo \$nome;
04	echo \$sobre_nome;
05	echo \$nascimento;
06	
07	\$nome = 'Mauro George';
08	\$nascimento = '06/09/1989';
09	\$sobre_nome = 'Oliveira Tavares';
10	
11	?>

Você sabe me dizer o por quê? Se você respondeu que devido a tentarmos exibir uma variável sem antes declará-la você acertou. Ou seja, antes de exibirmos qualquer dado antes devemos o ter declarado previamente.



**Comentários no PHP**

O PHP nos fornece um suporte a comentários que vem a ser muito útil quando estamos desenvolvendo sistemas. Temos os seguintes tipos de comentários no PHP:

1. // comentário de uma linha apenas
2. # também comentário de uma linha apenas
3. /\* \*/ comentário de múltiplas linhas

Veja todos eles em funcionamento abaixo:

01	<?php
02	
03	// Comentario de uma linha apenas
04	# Outro modo de escrever um comentario de uma linha
05	
06	/* Comentario de varias linhas com
07	este podemos quebra linha e escrever
08	quantas linhas quisermos.
09	*/
10	
11	// Apenas um nome
12	\$nome = 'Mauro George';
13	# A data de nascimento de uma pessoa
14	\$nascimento = '06/09/1989';
15	// O Sobre nome de uma pessoa
16	\$sobre_nome = 'Oliveira Tavares';
17	
18	/* Aqui exibimos o nome
19	o sobre nome e a data de nascimento de
20	uma pessoa
21	*/
22	echo \$nome;
23	echo \$sobre_nome;
24	echo \$nascimento;
25	

26	?>
----	----

### Constantes no PHP

Como você já deve ter imaginado as constantes no PHP guardam valores que nunca serão alterados. Diferente das variáveis que possuem valores que podem ser alterados, sendo assim após definida uma constante ela não pode ser alterada ou removida.

Para definirmos uma constante utilizamos o comando define(); que tem sua sintaxe a seguir:

```
define( 'NOME_DA_CONSTANTE', 'VALOR DA CONSTANTE' );
```

O nome de uma constante tem a mesma regra de qualquer identificador PHP, ou seja, as mesmas regras de nomes de variáveis exceto pelo fato de constantes **não** iniciarem o nome com cifrão (\$). Veja um exemplo a seguir em que utilizamos uma constante.

01	<?php
02	// Defino o titulo da minha página
03	\$titulo = 'Exemplo utilizando Constantes';
04	// Apenas um nome
05	\$nome = 'Mauro George';
06	// A data de nascimento de uma pessoa
07	\$nascimento = '06/09/1989';
08	// O Sobre nome de uma pessoa
09	\$sobre_nome = 'Oliveira Tavares';
10	// Defino o ESTADO da pessoa que é uma constante
11	define( 'ESTADO', 'Rio de Janeiro' );
12	
13	?>
14	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15	<html xmlns="http://www.w3.org/1999/xhtml">
16	<head>
17	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
18	<title><?php echo \$titulo; ?></title>
19	</head>
20	
21	<body>
22	<p><?php echo \$nome; ?> <?php echo \$sobre_nome; ?>, nascido em <?php echo \$nascimento; ?> nasceu no <?php echo ESTADO; ?></p>

23	</body>
24	</html>

Agora que você entendeu o funcionamento de uma constante deve estar se perguntando: qual a diferença real entre variáveis e constantes?

Acompanhe o exemplo a seguir e observe que variáveis podem ter seus valores modificados durante a execução de uma página, já as constantes seus valores nunca podem ter seus valores alterados.

01	<?php
02	// Defino o titulo da minha página
03	\$titulo = 'Exemplo utilizando Constantes';
04	// Apenas um nome
05	\$nome = 'Mauro George';
06	// A data de nascimento de uma pessoa
07	\$nascimento = '06/09/1989';
08	// O Sobre nome de uma pessoa
09	\$sobre_nome = 'Oliveira Tavares';
10	// Defino o ESTADO da pessoa que é uma constante
11	define( 'ESTADO', 'Rio de Janeiro' );
12	
13	?>
14	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15	<html xmlns="http://www.w3.org/1999/xhtml">
16	<head>
17	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
18	<title><?php echo \$titulo; ?></title>
19	</head>
20	
21	<body>
22	<p><?php echo \$nome; ?> <?php echo \$sobre_nome; ?>, nascido em <?php echo \$nascimento; ?> nasceu no <?php echo ESTADO; ?></p>
23	<?php
24	/**
25	* Redefino os valores das seguintes variaveis

26	*
27	*/
28	\$nome = 'Jéssica';
29	\$nascimento = '12/07/1990';
30	\$sobre_nome = 'Monteiro da Silva';
31	define( 'ESTADO', 'São Paulo' );
32	
33	?>
34	<p><?php echo \$nome; ?> <?php echo \$sobre_nome; ?>, nascido em <?php echo \$nascimento; ?> nasceu no <?php echo ESTADO; ?></p>
35	</body>
36	</html>

Por padrão sempre escreva o nome de constantes em maiúsculo e separado por \_ (sublinhado). Exemplo:

MINHA\_CONSTANTE, UMA\_CONSTANTE, CONSTANTE

E não se esqueça que constantes não são iniciadas com o cifrão (\$).

### PHP: Como Dar Os Primeiros Passos

Se você sempre teve vontade de programar em PHP, mas nunca soube por onde começar, saiba que não está sozinho. Embora seja possível encontrar praticamente qualquer tipo de conteúdo na internet, muitas vezes os aspirantes a programadores ficam um pouco perdidos com tanta informação e não conseguem encontrar um ponto de partida para a sua empreitada.

Às vezes, ter alguém para apontar o caminho certo é tudo o que uma pessoa precisa para se encontrar no mundo da programação. Por isso, o Tecmundo resolveu criar este guia, mostrando onde encontrar as informações necessárias para começar a programar em PHP e também o que é importante saber e fazer antes de avançar para os níveis mais complexos. Confira!

### Por Onde Começar?

Antes de começar a codificar uma página, é importante configurar o seu computador com os aplicativos necessários. Normalmente, quando um programador vai realizar alguma alteração em um site, o trabalho é feito primeiro em uma cópia local dos endereços para só então ser enviado para o servidor. Dessa forma, não há o risco de o portal ficar fora do ar devido a alguma falha durante a manutenção. Para ter essa cópia local funcionando corretamente no seu computador, é preciso instalar e configurar ferramentas como o Apache, PHP e MySQL. Assim, você consegue simular um ambiente de servidor na máquina e abrir a página no navegador mesmo sem estar conectado à internet.

A parte de configuração das ferramentas citadas é de extrema importância, pois garante a exibição correta do conteúdo programado. Para quem nunca realizou a tarefa de instalação dos aplicativos, ter um guia mostrando todos os passos por perto é uma ótima ideia.

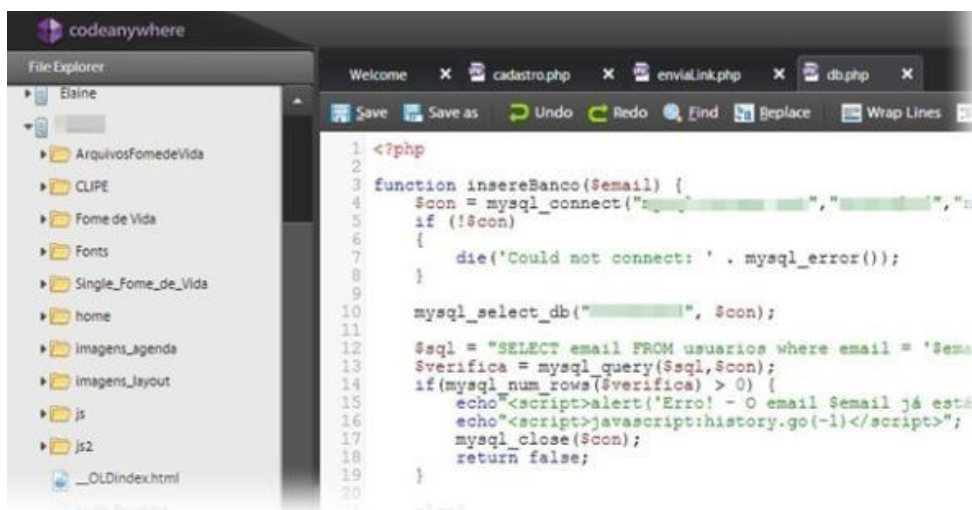


Se você digitar no Google os termos “instalar apache php mysql windows”, certamente vai encontrar diversos fóruns e tutoriais mostrando todo o processo de forma detalhada. Além disso, existem alguns pacotes que facilitam bastante a tarefa de instalação, pois oferecem todos os recursos necessários para rodar sua página localmente em uma só aplicação. Confira abaixo algumas dessas ferramentas.

- XAMPP
- EasyPHP
- PHP Triad
- WampServer

### Escolhendo Um Bom Editor

Outra ferramenta indispensável para um programador é o editor ou IDE utilizado na hora de mexer no código. Uma ferramenta não precisa ter dezenas de funcionalidades para ser boa, executando as funções básicas de forma satisfatória pode ser o suficiente.



Cada programador possui a sua ferramenta preferida. Tenha em mente que funcionalidades como autocompletar e destaque de sintaxe não são obrigatórias, mas ajudam muito na hora de programar, principalmente se você está começando e não lembra muito bem os comandos de cabeça. Boas opções de editor para PHP (e outras linguagens também) são:

- PHP Editor
- PHP Anywhere
- Bluefish
- Eclipse PDT
- Notepad++

### Agora Sim, Programar!

Se você decidiu criar um site em PHP, mas não tem ideia de como começar a programar, uma boa ideia é entender melhor alguns conceitos de programação, saber bem como funciona a linguagem PHP e procurar alguns tutoriais bem básicos na internet.

Apesar de já instalado e configurado o Apache, PHP e MySQL, entender a utilidade de cada um desses itens pode ser de grande ajuda, pois, caso haja algum problema no futuro, fica mais fácil diagnosticar o que o está causando.

O **Curso de PHP em Vídeo Aulas**, disponível para download no Baixaki, pode ser uma ótima forma de estudar alguns conceitos e aprender na prática os passos básicos para programar em PHP. No site **Apostilando.com** também é possível encontrar diversas apostilas gratuitas que explicam muito bem os conceitos para quem deseja embarcar nesse mundo do desenvolvimento de sites.

Há também o portal **PHP.net**. Além de explicar bem o que é o PHP (não a linguagem), você encontra diversos tutoriais ensinando a instalar e configurar o serviço no seu computador.

Outro serviço que certamente é muito útil para os programadores web é o **w3schools.com**. Nele, você encontra tutoriais, exemplos e explicações detalhadas de utilização para quase todos os comandos utilizados nas linguagens de desenvolvimento mais comuns para a internet. O conteúdo encontra-se apenas em inglês.

Apesar de ter um nome um tanto peculiar, o **Dummies.com** é um bom lugar para encontrar informações básicas a respeito de programação e desenvolvimento com PHP. Infelizmente, o site não possui tradução para o português, mas não é preciso dominar muito o inglês para compreender os textos disponíveis.

Não há muitos segredos para desenvolver um site em PHP, mas é preciso ter um pouco de paciência no começo e insistir. Se você não entender algum conceito ou termo presente nas apostilas, livros ou portais, procure os fóruns de programadores e não tenha vergonha de perguntar.

### O que é o PHP?

O PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.

Ótimo, mas o que isso significa? Por exemplo:

#### Exemplo #1 Um exemplo introdutório

```
<!DOCTYPE HTML>
<html>
<head>
<title>Exemplo</title>
</head>
<body>

<?php
echo "Olá, eu sou um script PHP!";
?>
</body>
</html>
```

Em vez de muitos comandos para mostrar HTML (como acontece com C ou Perl), as páginas PHP contêm HTML em código mesclado que faz "alguma coisa" (neste caso, mostra "Olá, eu sou um script PHP!"). O código PHP é delimitado pelas instruções de processamento (tags) de início e fim `<?php` e `?>` que permitem que você pule para dentro e para fora do "modo PHP".

O que distingue o PHP de algo como o Javascript no lado do cliente é que o código é executado no servidor, gerando o HTML que é então enviado para o navegador. O navegador recebe os resultados da execução desse script, mas não sabe qual era o código fonte. Você pode inclusive configurar seu servidor web para processar todos os seus arquivos HTML com o PHP, e então não haverá realmente nenhum modo dos usuários descobrirem se você usa essa linguagem ou não.

A melhor coisa em usar o PHP é que ele é extremamente simples para um iniciante, mas oferece muitos recursos para um programador profissional. Não tenha medo de ler a longa lista de funções do PHP. Pode entrar com tudo, o mais rápido que puder, e comece a escrever scripts simples em poucas horas.

Apesar do desenvolvimento do PHP ser focado nos scripts do lado do servidor, você pode fazer muito mais com ele. Veja sobre isso na seção *O que o PHP pode fazer?*, ou vá diretamente para tutorial introdutório se você estiver interessado em programação web.

### **O que o PHP pode fazer?**

Qualquer coisa. O PHP é focado principalmente nos scripts do lado do servidor, portanto, você pode fazer qualquer coisa que outro programa CGI pode fazer: coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies. Mas o PHP pode fazer muito mais.

Esses são os maiores campos onde os scripts PHP podem ser utilizados:

- Scripts no lado do servidor (server-side). Este é o mais tradicional e principal campo de atuação do PHP. Você precisa de três coisas para isto funcionar. O interpretador do PHP (CGI ou módulo do servidor), um servidor web e um navegador web. Você precisa rodar o servidor web conectado em uma instalação PHP. Você pode acessar os resultados de seu programa PHP com um navegador web, visualizando a página PHP através do servidor web. Tudo isso pode rodar na sua máquina pessoal se você estiver apenas experimentando programar com o PHP. Veja a seção das instruções de instalação para mais informações.

- Scripts de linha de comando. Você pode fazer um script PHP para executá-lo sem um servidor ou navegador. A única coisa necessária é o interpretador PHP. Esse tipo de uso é ideal para script executados usando o cron (Unix, Linux) ou o Agendador de Tarefas (no Windows). Esses scripts podem ser usados também para rotinas de processamento de texto simples. Veja a seção *Utilizando o PHP em linha de comando* para mais informações.

- Escrever aplicações desktop. O PHP provavelmente não é a melhor linguagem para criação de aplicações desktop com interfaces gráficas, mas se você conhece bem o PHP, e gostaria de usar alguns dos seus recursos avançados nas suas aplicações do lado do cliente, você pode usar o PHP-GTK para escrever programas assim. Você também tem a possibilidade de escrever aplicações multi-plataformas desse jeito. O PHP-GTK é uma extensão do PHP, não disponibilizada na distribuição oficial. Caso esteja interessado no PHP-GTK, visite » o site do projeto.

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes do Unix (como HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS e provavelmente outros. O PHP também é suportado pela maioria dos servidores web atualmente. Isso inclui o Apache, o IIS e muitos outros. Também qualquer servidor web que pode utilizar o binário FastCGI do PHP, como o lighttpd e o nginx. O PHP trabalha tanto como módulo quanto como um processador CGI.

Com o PHP, portanto, você tem liberdade de escolha de sistema operacional e de servidor web. Do mesmo modo, você pode escolher entre utilizar programação estruturada ou programação orientada a objeto (OOP), ou ainda uma mistura das duas.

Com PHP você não está limitado a gerar somente HTML. As habilidades do PHP incluem geração de imagens, arquivos PDF e até animações Flash (utilizando libswf ou Ming) criados dinamicamente, on the fly. Você pode facilmente criar qualquer padrão texto, como XHTML e outros arquivos XML. O PHP pode gerar esses padrões e os salvar no sistema de arquivos, em vez de mostrá-los em tela, formando um cache no lado do servidor para seu conteúdo dinâmico.

Uma das características mais fortes e mais significativas do PHP é seu suporte a uma ampla variedade de banco de dados. Escrever uma página web consultando um banco de dados é incrivelmente simples usando uma das extensões específicas de um banco de dados (e.g., mysql), ou usando uma camada de abstração como o PDO ou conectar a qualquer banco de dados que suporte o padrão "Open Database Connection" usando a extensão ODBC. Outros bancos de dados podem utilizar cURL ou sockets, como o CouchDB.

O PHP também tem suporte para comunicação com outros serviços utilizando protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (em Windows) e incontáveis outros. Você também pode abrir sockets de rede e interagir diretamente com qualquer protocolo. O PHP também suporta o





## Programação Python

As linguagens de programação exercem um papel fundamental dentro da internet como a conhecemos hoje. E não somente no desenvolvimento de sites e softwares, suas aplicações mais óbvias, mas também em áreas mais complexas como ciência de dados, interações entre humanos e máquinas por meio de inteligência artificial e coleta de informações, entre outros tipos de soluções.

Mas, ao contrário do que pode parecer para o público leigo, quanto mais complexa as funções exercidas por um sistema, mais intuitiva deve ser sua utilização.

Esta é uma das razões pelas quais as linguagens de programação de entendimento mais simples têm ganhado notoriedade nos últimos anos e claro que Python está entre as mais amadas.

Criado sob a filosofia de valorizar o esforço humano sobre o computacional, Python é uma linguagem de programação acessível e muito popular em setores emergentes da indústria de tecnologia.

Python é uma linguagem de programação de alto nível — ou High Level Language —, dinâmica, interpretada, modular, multiplataforma e **orientada a objetos** — uma forma específica de organizar softwares onde, a grosso modo, os procedimentos estão submetidos às classes, o que possibilita maior controle e estabilidade de códigos para projetos de grandes proporções.

Por ser uma linguagem de sintaxe relativamente simples e de fácil compreensão, ganhou popularidade entre profissionais da indústria tecnológica que não são especificamente programadores, como engenheiros, matemáticos, cientistas de dados, pesquisadores e outros.

Um de seus maiores atrativos é possuir um grande número de bibliotecas, nativas e de terceiros, tornando-a muito difundida e útil em uma grande variedade de setores dentro de desenvolvimento web, e também em áreas como análise de dados, machine learning e IA.

Qual a origem da linguagem Python?

Idealizada e desenvolvida por Guido Van Rossum, matemático holandês, no início dos anos 90, o Python foi criado com o objetivo de otimizar a leitura de códigos e estimular a produtividade de quem os cria, seja este um programador ou qualquer outro profissional.

A ideia surgiu, como quase todas as boas ideias, de uma necessidade: a de economizar tempo no desenvolvimento e melhorar a eficiência em um projeto desenvolvido no instituto onde Guido era pesquisador.

Para que esta melhoria pudesse ser feita de forma mais rápida e eficaz, Guido desenvolveu uma linguagem muito descomplicada e flexível: o Python.

Uma vez que esta linguagem passou a possibilitar a criação desde scripts muito simples até sistemas extremamente poderosos, profissionais de várias áreas começaram a progressivamente utilizá-la cada vez mais.

Hoje, além dos desenvolvedores de software, temos biólogos, contadores, físicos e outros profissionais potencializando suas habilidades através dela.

Estes são alguns motivos que tem feito o uso do Python crescer consideravelmente nos últimos anos em detrimento de outras linguagens.

Para que é usado o Python?

Como já citado acima, o Python é uma linguagem muito popular nas áreas da tecnologia relacionadas à análise de dados, pesquisa, desenvolvimento de algoritmos e IA. Mas, afinal, o que pode ser feito em Python?

Abaixo, detalharei um pouco mais suas utilizações em cada uma destas áreas.

### **Scripting E Automação**

Automatizar tarefas é um dos maiores objetivos de um profissional de tecnologia.

E desenvolver scripts de automação com Python é totalmente possível e rápido graças às suas várias bibliotecas nativas, disponíveis junto com sua instalação.

### **Desenvolvimento Web**

Uma das aplicações mais comuns do Python é o desenvolvimento de aplicações para web. Desde sites simples, como hotpages para ações de marketing, quanto ERPs capazes de administrar empresas e realizar vendas de produtos e serviços.

A linguagem possui uma extensa variedade de frameworks para todos os tipos de gosto, entre eles os famosos Django, Flask e FastAPI.

### **Enquadramento De Testes**

Se o momento de realizar testes era um momento desagradável no seu dia de trabalho, saiba que com o Python a sua vida será muito mais animada.

Ele possui um grande volume de estruturas de testes integradas, além de diversos módulos voltados exclusivamente para o enquadramento de testes.

Então, utilizar o Python é poder ter a certeza que sua aplicação poderá ser testada com muita eficiência.

Isso justifica por que desenvolvedores de software em Python adoram utilizar TDD (Test Driven Development), trazendo mais segurança e confiança aos seus projetos.

### **Big Data**

A análise e interpretação de grandes volumes de dados, área de conhecimento que chamados de **Big Data**, é um dos mais populares campos onde Python tem se tornado a linguagem favorita.

Por ser facilmente compreendida por profissionais de diversas especialidades, o Python é, antes de tudo, um facilitador.

Além disso, duas características principais o levam a ser ótimo para essa tarefa: criação de scripts e a facilidade em utilizar as várias bibliotecas gratuitas desenvolvidas em parcerias com especialistas de várias áreas.

É possível analisar, processar e também exibir os dados com muita eficiência e clareza. Embora ainda haja grandes concorrentes, como o R, por exemplo, o Python, assim como o próprio Big Data, está com a sua notoriedade no auge.

### **Ciência De Dados**

Acompanhando o próprio avanço da indústria tecnológica, o crescente número de bibliotecas disponíveis em Python voltadas à análise de dados oferece funções e métodos de otimização para praticamente quaisquer objetivos.

A grande comunidade de Python, não apenas internacional mas também no Brasil, é um poderoso atrativo: o compartilhamento de soluções e informações entre profissionais da área diminui consideravelmente as chances de um programador precisar lidar com um problema sem solução aparente.

### Computação Gráfica

Se você já assistiu aos mais recentes filmes da saga Star Wars, saiba que todos os seus efeitos de computação gráfica, produzidos pela Industrial Light & Magic, envolveu a linguagem Python.

Além de haver vários pacotes de soluções para esta área, como PyOpenGL e PyGame, há ainda o poderoso software de criação de gráficos 3D, o Blender, que utiliza o Python como linguagem principal.

### Inteligência Artificial

Quando você faz uma busca no Google e encontra exatamente a resposta que esperava, o Python desempenhou exatamente o papel que se esperava de sua atuação.

Isso porque os algoritmos e modelos de inteligência artificial presentes no buscador, elaborados para prever a sua intenção de busca, são em sua maioria desenvolvidos nesta linguagem.

Entre as bibliotecas mais comuns voltadas ao aprendizado de máquina estão o TensorFlow, PyTorch, Theano, Keras e outras.

Quais as vantagens de programar em Python?

Como você pode estar percebendo, as vantagens de se dedicar ao aprendizado de Python são muitas, e entre elas está o fato de que os profissionais especializados nesta linguagem de programação no mercado são escassos.

Ou seja: ao se destacar como um bom programador Python, sua concorrência profissional será baixíssima.

### Outros Benefícios Do Python São:

**1. Simples e fácil de aprender:** a curva de aprendizado de um estudante de Python é, de modo geral, relativamente baixa. A linguagem, por ter uma sintaxe muito acessível e ter sido criada em prol da agilidade e da produtividade de quem a utiliza, é absorvida rápida e facilmente.

**2. Portátil, extensível e multiplataforma:** por ser uma linguagem portátil e multiplataforma, o Python roda com tranquilidade em diversos sistemas operacionais, desde que seu interpretador esteja instalado. Além disso, o Python também é conhecido por suas propriedades extensíveis, tendo à sua disposição mais de 125.000 bibliotecas super versáteis.

**3. Licença de uso público:** em outras palavras: o Python é totalmente gratuito! Para instalar, utilizar e desenvolver em Python, basta simplesmente fazê-lo.

Além disso, a maior pesquisa realizada na área da programação, a StackOverflow Survey, perguntou este ano para desenvolvedores do mundo inteiro em qual linguagem eles mais gostam de programar e adivinhem: Python ficou em 1º lugar!

Quais empresas usam Python?

O Python é extremamente presente na vida de quem utiliza a internet com regularidade, principalmente por sua colaboração na criação de algoritmos, desde os menos complexos até funções de aprendizado de máquina.



## Gerência de Configuração de Software

Os sistemas de software estão em constante evolução. A manutenção do software, isto é, modificações em artefatos existentes, chega a consumir 75% do custo total do seu ciclo de vida. Aproximadamente, 20% de todo o esforço de manutenção é usado para consertar erros de implementação e os outros 80% são utilizados na adaptação do software em função de modificações em requisitos funcionais, regras de negócios e na reengenharia da aplicação.

A Gerência de Configuração de Software surgiu da necessidade de controlar estas modificações, por meio de métodos e ferramentas, com o intuito de maximizar a produtividade e minimizar os erros cometidos durante a evolução.

É uma disciplina que controla e notifica as inúmeras correções, extensões e adaptações aplicadas durante o ciclo de vida do software de forma a assegurar um processo de desenvolvimento e evolução sistemático e rastreável, sendo indispensável quando equipes manipulam, muitas vezes em conjunto, artefatos comuns.

Apesar de existir um forte apelo para o uso da Gerência de Configuração de Software durante a etapa de manutenção, a sua aplicação não se restringe somente a essa etapa do ciclo de vida do software.

O uso dos sistemas de Gerência de Configuração é fundamental para prover controle sobre os artefatos produzidos e modificados por diferentes recursos desde o planejamento e levantamento de requisitos até a construção e entrega do produto. O motivo da sua importância está geralmente associado aos problemas identificados quando a Gerência de Configuração não é utilizada no desenvolvimento de software. Abaixo, vamos analisar alguns destes problemas.

Imagine que uma organização desconhece o que seja Gerência de Configuração e que em um determinado projeto um desenvolvedor esteja modificando os artefatos C1, C2 e C3 em um diretório compartilhado na rede. Simultaneamente, um segundo desenvolvedor modifica os artefatos C4, C5 e também o artefato C3, como exemplifica a Figura 1.

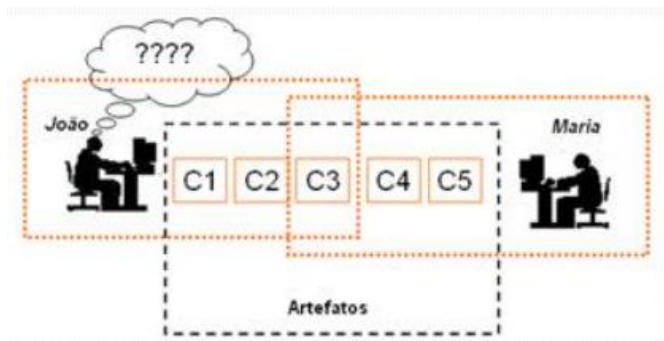


Figura 1. Espaço de Trabalho compartilhado por vários desenvolvedores

Neste cenário, o segundo desenvolvedor não notifica o primeiro desenvolvedor sobre o impacto que a modificação do artefato C3 pode causar no código. Conseqüentemente, o primeiro desenvolvedor, que está usando o mesmo espaço de trabalho, não conseguirá identificar, de forma rápida, o motivo que levou sua implementação a falhar. Este problema acontece pela falta de notificação e pelo compartilhamento de artefatos de software por diversos desenvolvedores.

Imagine que agora foi acordado entre os desenvolvedores que o ideal seria centralizar os artefatos em um repositório e que cada desenvolvedor implementaria suas modificações em um espaço de trabalho privado. Após cada modificação, o artefato seria devolvido ao repositório. Considerando este cenário, frequentemente ocorreriam sobreposições ou perdas de modificações implementadas nos artefatos comuns nas organizações sem a prática da Gerência de Configuração.

Um desenvolvedor poderia implementar sua modificação em uma versão desatualizada do artefato e sobrepor a versão mais atual disponibilizada por outro. Este problema ocorre devido à atualização simultânea, quando dois desenvolvedores compartilham o mesmo repositório e não existe controle ou restrição quanto ao acesso a este repositório (ver Figura 2).

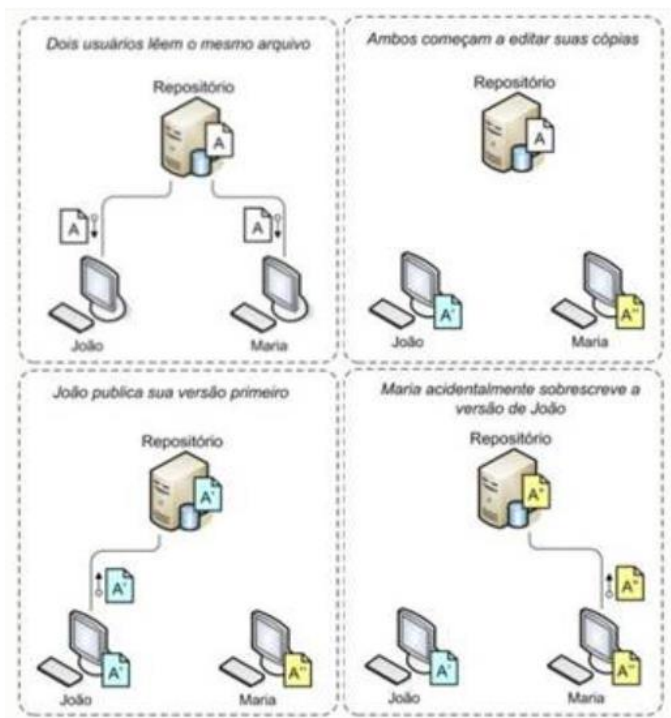


Figura 2. Repositório centralizado compartilhado por vários desenvolvedores

Sob a perspectiva de desenvolvimento, a Gerência de Configuração de Software abrange três sistemas principais: controle de modificações, controle de versões e controle de gerenciamento de construção.

O sistema de controle de versões permite que os artefatos sob Gerência de Configuração evoluam de forma distribuída, concorrente e disciplinada, evitando perdas ou sobreposições durante o desenvolvimento e a manutenção do artefato. Podemos citar como exemplos de ferramentas de mercado: CVS, Subversion, IBM Rational ClearCase e Microsoft Visual Source Safe.

O sistema de controle de modificações armazena todas as informações geradas durante o andamento das solicitações de modificação e relata essas informações aos participantes interessados e autorizados. Podemos citar como exemplos de ferramentas de mercado: Bugzilla, Jira, Trac e IBM Rational ClearQuest.

O sistema de gerenciamento de construção automatiza o processo de transformação dos diversos artefatos do software que compõem um projeto em um sistema executável propriamente dito. Este processo é nomeado construção do software que, por exemplo, testa e empacota a aplicação java como um arquivo jar. Este processo ocorre de forma aderente às normas, procedimentos, políticas e padrões definidos para o projeto. Podemos citar como exemplos de ferramentas de mercado: Maven e Apache Ant.

As vantagens da utilização da Gerência de Configuração de Software são inúmeras. Dentre elas, podemos listar:

- ganho de produtividade e eficiência;
- diminuição do retrabalho e dos erros;
- aumento da disciplina no processo de desenvolvimento;
- aumento da memória organizacional;

acesso às informações qualitativas e quantitativas referentes ao processo de desenvolvimento, como por exemplo, medida de esforço para efetuar uma alteração e frequência de modificações por componente;

possibilidade de estabelecer uma trilha de auditoria indicando por que, quando e por quem um artefato foi alterado;

auxílio à gerência de projetos e

garantia de ambiente estável no qual o produto deve ser desenvolvido.

Neste artigo, veremos alguns conceitos da área de Gerência de Configuração de Software e como esta área se relaciona com o processo de desenvolvimento de software. Também serão apresentadas algumas estratégias de organização do trabalho.

**Terminologia**

O sistema de controle de versões permite que os artefatos sejam obtidos, por meio de uma operação conhecida como check-out, modificados dentro do espaço de trabalho do desenvolvedor e, depois, retornados ao repositório, por meio de uma operação conhecida como check-in, como exemplifica a Figura 3. O repositório é o local de armazenamento dos artefatos que estão sob controle da Gerência de Configuração de Software. Estes artefatos recebem o nome de itens de configuração. A cada operação de check-in realizada, a versão do item de configuração é incrementada de uma unidade. Quando o item é adicionado pela primeira vez no repositório, este item passa a ter a versão igual a 1. Para cada item de configuração armazenado, são anexadas informações como: datas da criação ou alteração, comentários e versões.

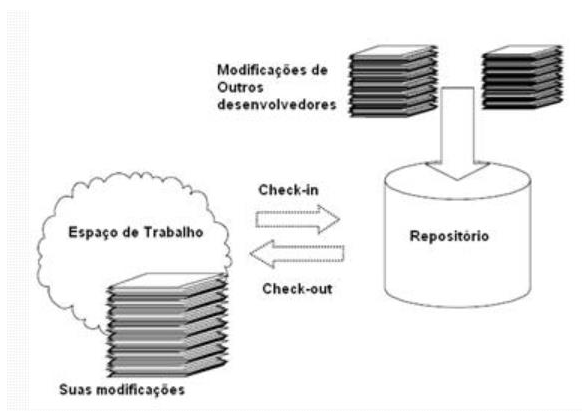


Figura 3. Operações Check-in e Check-out

Neste cenário, não há perdas ou sobreposições porque políticas de trabalho foram estabelecidas, restringindo ou controlando as modificações no repositório. As ferramentas de controle de versões normalmente suportam a definição de diferentes políticas de trabalho. Dentre essas políticas, podemos citar a política pessimista, que enfatiza o uso de check-out reservado, fazendo bloqueio e inibindo o paralelismo do desenvolvimento sobre o mesmo artefato. O uso desta política pode ser exemplificado na Figura 4.

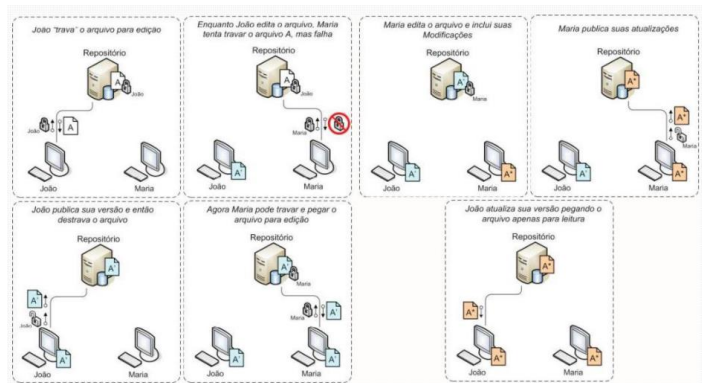


Figura 4. Política Pessimista

Outra política amplamente utilizada é a otimista. Neste cenário, se um artefato for alterado simultaneamente por dois desenvolvedores, a política assume que a quantidade de conflitos será naturalmente baixa e que será mais fácil tratar cada conflito individualmente, caso eles venham a ocorrer. A política otimista usa um mecanismo conhecido como junção (merge), que une as modificações efetuadas em paralelo sobre um mesmo artefato e produz uma nova versão deste artefato que contém a soma das modificações. Os conflitos ocorrem quando a mesma região ou linha do arquivo é modificada. A junção é automática, na maioria dos casos, mas quando ocorre um conflito, ela deve ser feita de forma manual. A Figura 5 ilustra o processo.

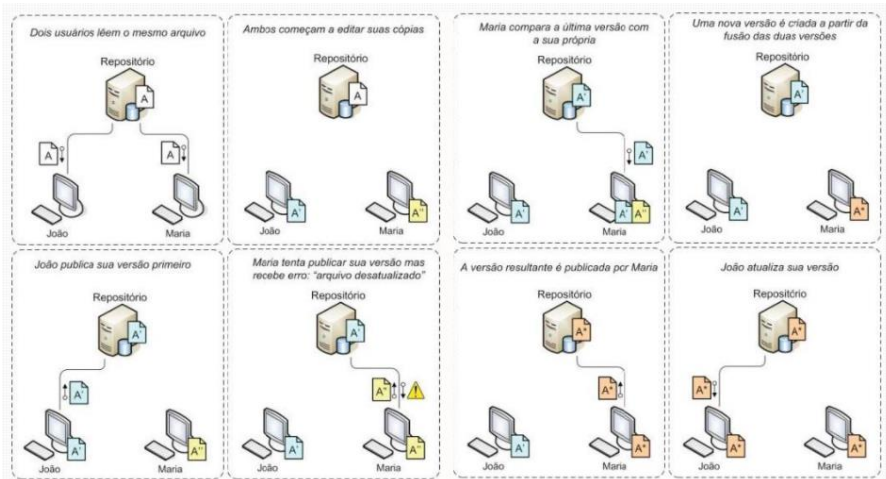


Figura 5. Política Otimista

Existem situações onde uma determinada política é mais indicada do que as demais. Nos casos onde a junção tende a ser complexa, quando, por exemplo, os artefatos não são textuais e a ferramenta não dá apoio automatizado para junções, é mais indicado trabalhar usando políticas pessimistas. Contudo, na grande maioria das situações referentes ao desenvolvimento de software, as políticas otimistas atendem satisfatoriamente.

Além desses recursos fundamentais presentes no sistema de controle de versão, outros recursos mais elaborados também são encontrados com frequência. Em determinados momentos do ciclo de vida de desenvolvimento e manutenção do software, os itens de configuração são agrupados e verificados, constituindo configurações do software voltadas para propósitos específicos. Neste momento, cria-se marcos no versionamento de artefatos que são denominados baselines ou releases.

A diferença entre baselines e releases é sutil. As baselines representam conjuntos de itens de configuração formalmente aprovados que servem de base para as etapas seguintes de desenvolvimento. Mas, quando uma entrega formal é feita ao cliente, no final de uma iteração, por exemplo, denominamos esta entrega de release. Baselines e releases são identificadas no repositório, na grande maioria das vezes, pelo uso de etiquetas (tags). A Figura 6 representa a baseline criada ao final da fase de codificação, considerando quatro itens de configuração. Cada item de configuração apresenta seu respectivo histórico de versões no repositório.

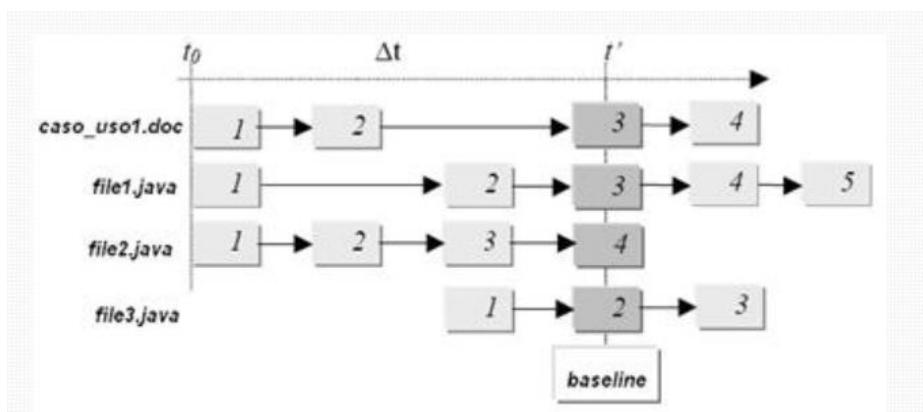




Figura 6. Definição de Baselines

A Gerência de Configuração de Software também permite que a implementação de novas funcionalidades por uma equipe seja realizada em paralelo, mas de forma isolada e independente das modificações de outros desenvolvedores. O isolamento é obtido com o uso de ramo (branch). As linhas de desenvolvimento (codelines) são designadas para cada projeto e são compartilhadas por vários desenvolvedores. A primeira linha de desenvolvimento definida no projeto é, por convenção, nomeada mainline. O ramo é criado no repositório e representa uma linha secundária de desenvolvimento que pode ser unida novamente à linha principal (mainline) por meio da operação de junção (merge). Atualmente, a necessidade de atender, ao mesmo tempo, as múltiplas demandas do projeto torna o uso de ramos um diferencial (ver Figura 7).

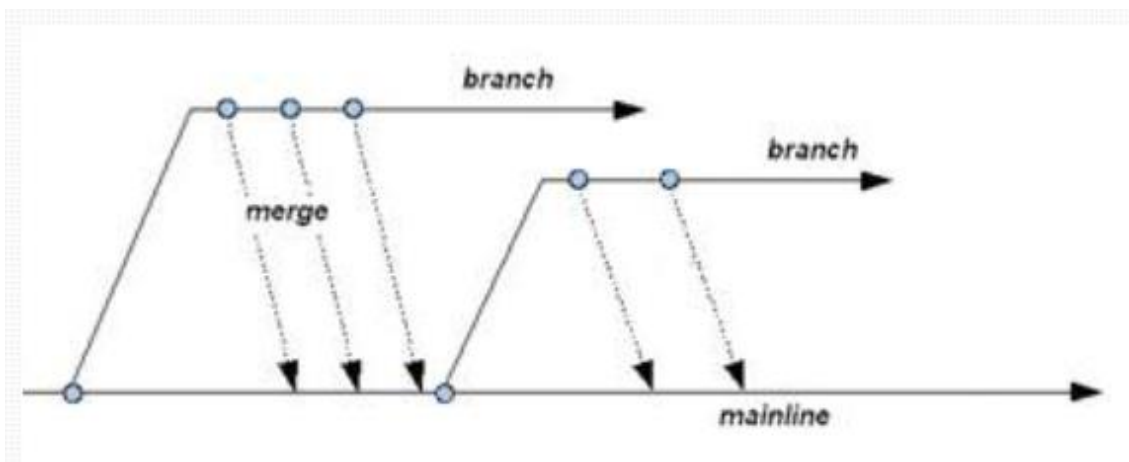


Figura 7. Definição de ramos

A Figura 8 exemplifica a criação de um ramo e a operação de junção. A junção é efetuada para cada artefato do ramo e todas as modificações efetuadas desde o ancestral em comum são levadas em consideração. Imagine que um artefato foi modificado na mainline e no ramo. Digamos que, quando o ramo foi criado, este artefato estava na mainline com a versão 1. Mas quando ocorreu a junção do ramo, ele estava na mainline com a versão 2. O processo de junção soma as modificações efetuadas em paralelo, criando uma nova versão do artefato. Portanto, ao final, a versão 3 do artefato contém as modificações  $\beta$  efetuadas na mainline desde a versão 1 deste artefato e também as modificações realizadas no ramo.

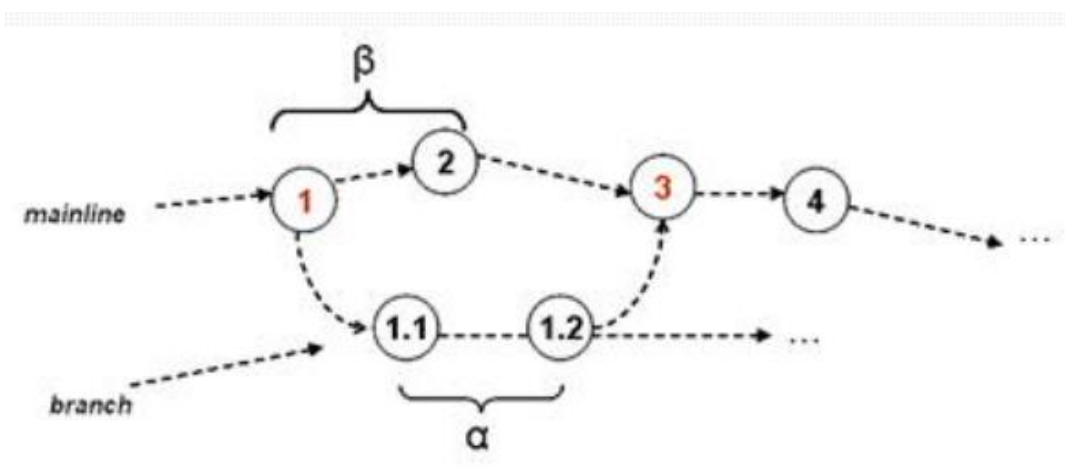


Figura 8. Junção

É importante ressaltar que o isolamento longo dificulta a operação de junção, portanto, é necessário realizar junções periodicamente.

**Estratégias de Organização**

Estratégias podem ser adotadas para organizar o trabalho dos desenvolvedores no projeto.

De forma genérica, criar um ramo é um meio de organizar o trabalho, pois isola o desenvolvimento de outras modificações e possibilita que o trabalho seja executado sem que uma modificação específica cause impacto nas demais alterações do software.

No entanto, o isolamento que o ramo promove, tem um custo associado. Mesmo com ótimas ferramentas, a junção pode ser uma atividade difícil, em função dos conflitos que ocorrem. Portanto, deve existir um balanceamento entre custo versus benefício, avaliando duas alternativas:

usar isolamento com ramos e conciliar os eventuais conflitos que podem surgir ao fazer a junção deste ramo com a mainline; ou

não usar isolamento e fazer todas as modificações na mainline. Esta última alternativa pode minimizar os problemas de integração, mas maximizar a concorrência e dificultar a entrega de uma release com um subconjunto de requisitos completo e correto em um marco do projeto.

Estratégias de ramificação podem ser utilizadas, ao longo do desenvolvimento, em situações específicas. Algumas podem ser combinadas, outras são mutuamente exclusivas. Dentre as estratégias de organização, vamos citar neste artigo:

[Manutenção caótica] Estratégia onde não existe isolamento e, portanto, a impossibilidade de separar o que é manutenção corretiva da evolutiva (ver Figura 9). Na mainline ocorre a evolução e a correção do software. Não existem ramificações.

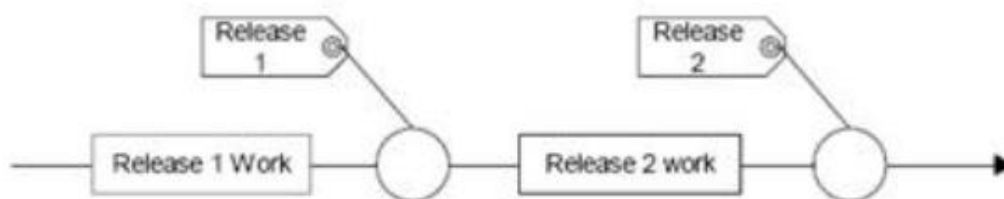


Figura 9. Manutenção caótica

### **Gerência de Configuração e Desenvolvimento de Software**

Por ser uma área fortemente calcada em controle, a Gerência de Configuração é referenciada em diversas normas, processos, procedimentos, políticas e padrões, como ISO 12207, CMMI e MPS.Br.

Do ponto de vista gerencial, o processo de Gerência de Configuração de Software é dividido em cinco funções (ver Nota 1): identificação da configuração, controle da configuração, acompanhamento da situação da configuração, auditoria da configuração e gerenciamento de entrega.

Nota 1:

A função de identificação da configuração tem por objetivo:

a seleção de quais artefatos serão itens de configuração;

a definição de uma nomenclatura, que possibilite a identificação inequívoca dos itens de configuração, baselines e releases; e

a descrição dos itens, tanto física quanto funcionalmente.

A seleção de itens de configuração é feita no início da fase de planejamento e leva em conta:

se o artefato é crítico para o projeto;

a dependência entre artefatos;

o impacto que uma modificação do item tem no produto;

se o artefato pode ser modificado por dois ou mais grupos;

se é frequentemente alterado devido a sua complexidade e

se é gerado manualmente, automaticamente ou ambos.

A função de controle da configuração é designada para controlar e acompanhar a evolução dos itens de configuração selecionados na função de identificação. Ferramentas como JIRA, Bugzilla, dentre outras, apoiam, em conjunto com as ferramentas de controle de versões, as atividades desta função.

A função de acompanhamento da situação da configuração visa:

armazenar as informações geradas pelas demais funções; e

permitir que essas informações possam ser acessadas em função de necessidades específicas, por exemplo, para a melhoria do processo, para a estimativa de custos futuros e para a geração de relatórios gerenciais.

Estas informações podem ser obtidas, no decorrer do projeto, a partir dos sistemas de controle de versões e modificações.

A função de auditoria da configuração ocorre geralmente quando uma release deve ser criada. Suas atividades compreendem:

auditoria funcional, que abrange a revisão dos planos, dados, metodologia e resultados dos testes, assegurando que a release cumpre corretamente o que foi especificado; e

auditoria física, com o objetivo de certificar que a release é completa em relação ao que foi acertado em cláusulas contratuais. A auditoria pode ser feita com base nos relatórios obtidos na função anterior.

Já a função de gerenciamento de liberação e entrega descreve o processo formal de:

construção e liberação de uma release do produto; e

entrega, com informações de como implantar o software no ambiente final de execução. Ferramentas, como Ant, permitem que roteiros de construção sejam escritos e executados no apoio a esta fase.

Para auxiliar e garantir a execução das atividades relacionadas com as funções da Gerência de Configuração de Software, uma organização pode definir uma equipe de Gerência de Configuração, normalmente única no contexto organizacional.

A Gerência de Configuração é um processo auxiliar de controle e acompanhamento das atividades do desenvolvimento de software (ler Nota 2).

Para exemplificar, vamos considerar um ciclo de vida iterativo e incremental que propõe inicialmente a identificação do escopo do projeto e a aprovação deste escopo pelo cliente. Posteriormente, nesta abordagem, o software é construído em ciclos sucessivos denominados iterações.

A cada iteração, os requisitos são priorizados, detalhados, aprovados e o software é modelado, construído e testado, como ilustra a Figura 11. Ao final, uma release do produto é entregue para homologação e aprovação do cliente.

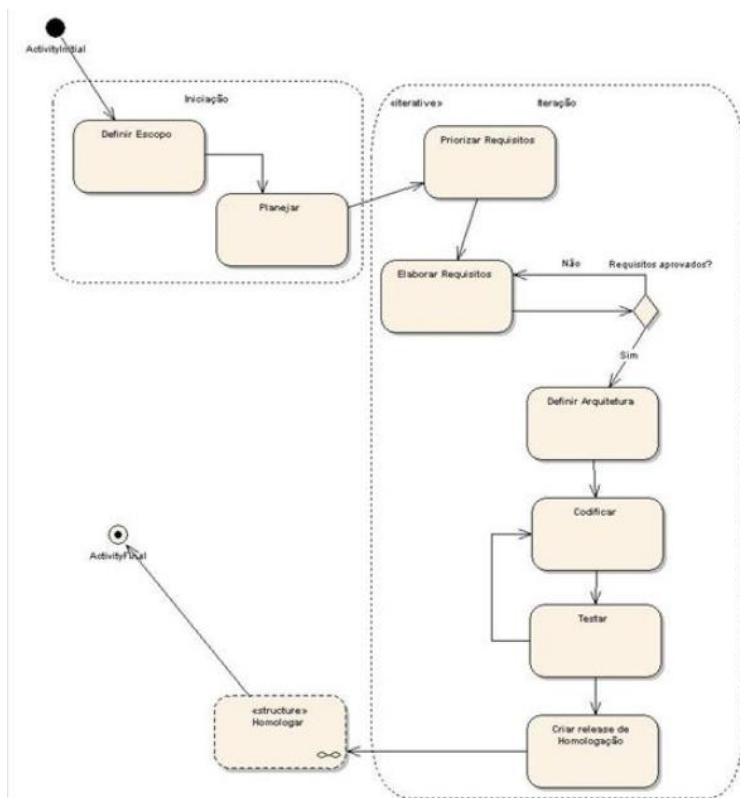


Figura 11. Processo de Desenvolvimento

Na atividade homologar é importante verificar a release em um ambiente isolado do desenvolvimento, realizando testes funcionais. Neste momento, à medida que erros são identificados no teste, solicitações de modificações são registradas na atividade modificar liberação. A Figura 12 ilustra a fase de homologação.

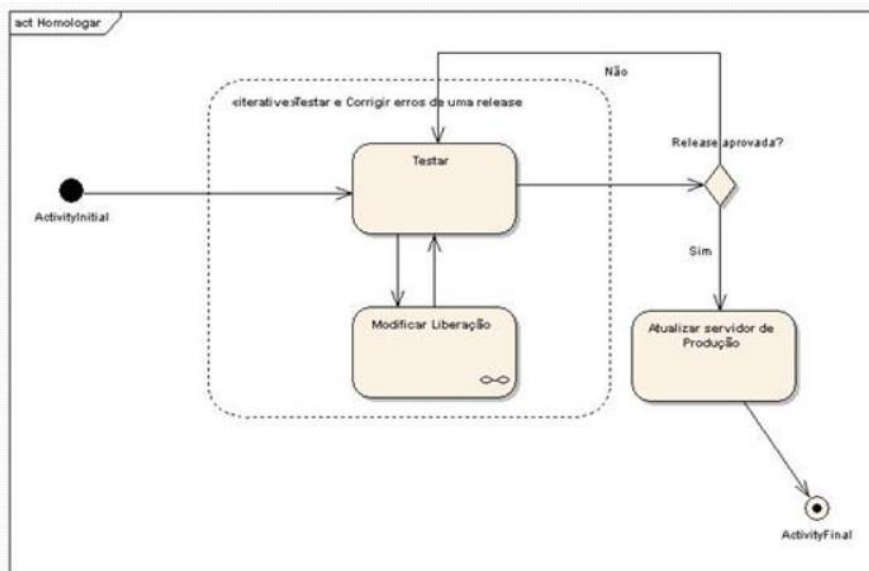


Figura 12. Fase de Homologação

Para que itens de configuração de uma baseline ou release possam evoluir de forma controlada, a função de controle da configuração estabelece as atividades:

solicitação de modificação, que pode ser corretiva, evolutiva, adaptativa ou preventiva;

classificação da modificação, que estabelece a prioridade da solicitação em relação às demais solicitações efetuadas anteriormente;

análise de impacto, que visa relatar os impactos em esforço, cronograma e custo;

avaliação da modificação, que estabelece se a modificação será implementada, rejeitada ou postergada, em função do laudo da análise de impacto da modificação;

implementação da modificação, caso a solicitação tenha sido aprovada pela avaliação da modificação;

verificação da modificação com relação à proposta de implementação levantada na análise de impacto;

gerência da baseline e release. Uma nova baseline ou release pode ser criada ao final, agrupando uma ou mais solicitações resolvidas. A Figura 13 ilustra esta função que está relacionada com a atividade modificar liberação executada na fase de homologação do produto.

Nota 2: Uso do processo de gerência de configuração

No caso de artefatos ainda em desenvolvimento, recomenda-se um processo mais ad-hoc, que não faça com que a dinâmica do desenvolvimento seja perdida. Neste caso, apenas as operações check-out e check-in são necessárias. Mas, recomenda-se que um resumo do que foi implementado seja exposto no comentário a cada operação de check-in realizada, juntamente com o item de configuração e sua respectiva versão no repositório.

No caso de baselines ou releases, isto é, versões estáveis do produto, é ideal ser mais formal, pois qualquer mudança nos itens de configuração pode causar impacto em custo e prazo de entrega do produto. O formalismo aplicado, no entanto, pode variar em função da flexibilidade que determinados processos de desenvolvimento de software necessitam.

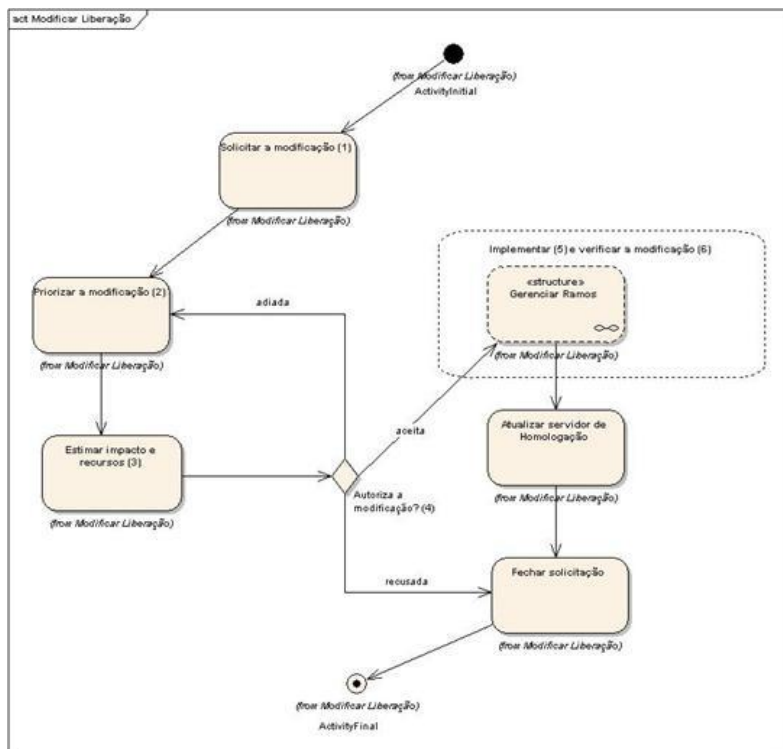


Figura 13. Modificar Liberação

Analisando as atividades da Figura 13, vemos que para implementar e verificar a modificação podemos criar um ramo separado. Portanto, podemos criar um ramo para corrigir os erros encontrados, adotando a estratégia de organização em série, como vimos na seção Estratégias de Organização. Desta forma, a implementação de novas funcionalidades do produto pode continuar na mainline, considerando as próximas iterações do desenvolvimento, enquanto a correção dos erros é feita em um ramo à parte.

Neste cenário, os erros são corrigidos no ambiente de desenvolvimento, e após todas as verificações, uma release é criada. Ao final, atualiza-se o ambiente de homologação com a nova release. Quando a release for aprovada pelo cliente, uma versão do produto pode ser posta em produção. A Figura 14 detalha este processo.

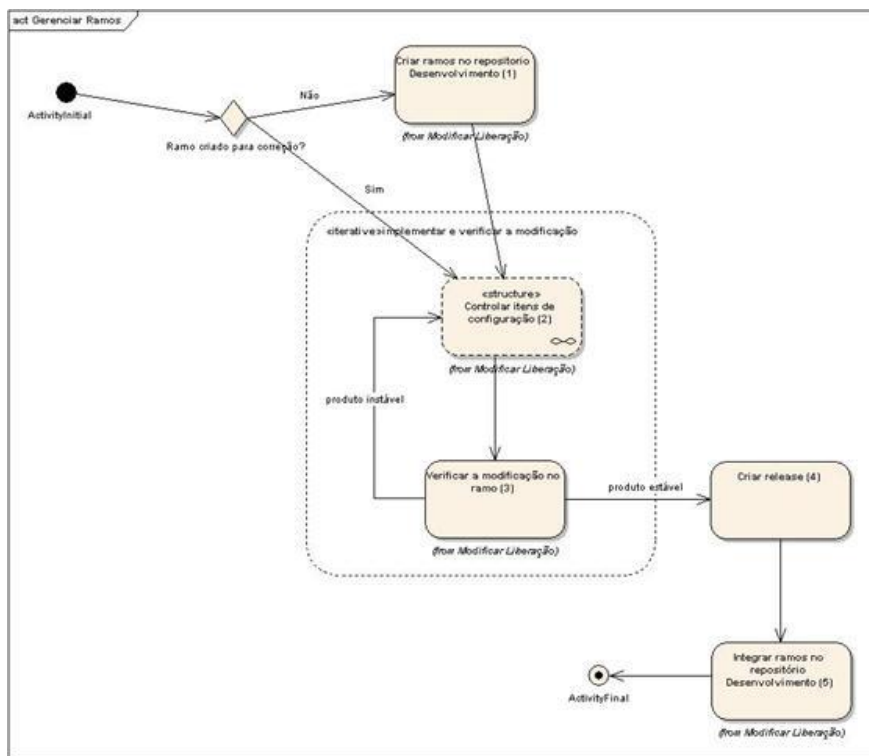


Figura 14. Gerenciar ramos

A atividade gerenciar ramos na Figura 14 mostra como proceder a implementação e verificação das correções dos erros encontrados na fase de homologação. Esta atividade apresenta os seguintes passos:

- criar ramo no repositório,
- controlar itens de configuração no ramo durante a atividade de implementação e teste,
- verificar se os erros foram corrigidos corretamente,
- criar release com um conjunto de solicitações atendidas,
- integrar na mainline as correções feitas no ramo em função das solicitações.

Neste contexto, controlar itens de configuração significa: efetuar check-out atualizando o espaço de trabalho com os itens contidos no ramo criado no repositório; alterar os itens e efetuar check-in, ao terminar a modificação, devolvendo os itens do espaço de trabalho para o repositório.

---



---



---



---



---



---

## Tipos Básicos De Dados

Em ciência da computação tipos de variáveis ou dados é uma combinação de valores e de operações que uma variável pode executar, o que pode variar conforme o sistema operacional e a linguagem de computador. São utilizados para indicar ao compilador ou interpretador as conversões necessárias para obter os valores em memória durante a construção do programa. O tipo de dado ajuda também o programador a detectar os eventuais erros envolvidos com semântica das instruções, erros esses detectados na análise semântica dos compiladores.

Dependendo da linguagem de programação, o tipo de um dado é verificado diferentemente, de acordo com a análise léxica, sintática e semântica do compilador ou interpretador da linguagem. Os tipos têm geralmente associações com valores na memória ou com objetos (para uma linguagem orientada a objeto) ou variáveis.

## Tipo Estático e Dinâmico

A verificação do tipo de um dado é feita de forma estática em tempo de compilação ou de forma dinâmica em tempo de execução. Em C, C++, Java e Haskell os tipos são estáticos, em Scheme, Lisp, Smalltalk, Perl, PHP, Visual Basic, Ruby e Python são dinâmicos.

Em C uma definição estática do tipo de uma variável ficaria assim:

```
printf ("O tipo char ocupa %lld bytes\n", sizeof(char));
```

### Tipo Forte E Fraco

Linguagens implementadas com tipificação forte (linguagem fortemente tipificada), tais como Java e Pascal, exigem que o tipo de dado de um valor seja do mesmo tipo da variável ao qual este valor será atribuído. Exemplo:

(Sintaxe genérica)

1. Declarar Variáveis
2. TEXTO nome
3. INTEIRO idade
5. Atribuições
6. nome = "Helisson"
7. idade = 13.1

Ocorrerá um erro ao compilar a linha 7, pois o valor "13.1", que é do tipo REAL, precisa ser convertido para o tipo de dado INTEIRO.

Em linguagens com tipos de dados fracos, tais como PHP e VBScript, a conversão não se faz necessária, sendo realizada implicitamente pelo compilador ou interpretador.

## Tipo Primitivo e Composto

Um tipo primitivo (também conhecido por nativo ou básico) é fornecido por uma linguagem de programação como um bloco de construção básico. Dependendo da implementação da linguagem, os tipos primitivos podem ou não possuir correspondência direta com objetos na memória.

Um tipo composto pode ser construído em uma linguagem de programação a partir de tipos primitivos e de outros tipos compostos, em um processo chamado composição.

Em C, cadeias de caracteres são tipos compostos, enquanto em dialetos modernos de Basic e em JavaScript esse tipo é nativo da linguagem.

Tipos primitivos típicos incluem caractere, inteiro (representa um subconjunto dos números inteiros, com largura dependente do sistema; pode possuir sinal ou não), ponto flutuante (representa o conjunto dos números racionais), booleano (lógica booleana, verdadeiro ou falso) e algum tipo de referência (como ponteiro ou handles).

Tipos primitivos mais sofisticados incluem tuplas, listas ligadas, números complexos, números racionais e tabela hash, presente sobretudo em linguagens funcionais.

Espera-se que operações envolvendo tipos primitivos sejam as construções mais rápidas da linguagem. Por exemplo, a adição de inteiros pode ser feita com somente uma instrução de máquina, e mesmo algumas CPUs oferecem instruções específicas para processar sequências de caracteres com uma única instrução. A maioria das linguagens não permite que o comportamento de um tipo nativo seja modificado por programas. Como exceção, Smalltalk permite que tipos nativos sejam entendidos, adicionando-se operações e também redefinindo operações nativas.

Uma estrutura em C e C++ é um tipo composto de um conjunto determinado de campos e membros. O tamanho total da estrutura para o tipo composto corresponde a soma dos requerimentos de cada campo da estrutura, além de um possível espaço para alinhamento de bits. Por exemplo:

```
struct Conta {  
  
    int numero;  
  
    char *nome;  
  
    char *sobrenome;  
  
    float balanco;};
```

Define um tipo composto chamado Conta. A partir de uma variável minhaConta do tipo acima, pode-se acessar o número da conta através de minhaConta.numero.

### **Algoritmos para Pesquisa e Ordenação**

Em ciência da computação, um algoritmo de busca, em termos gerais é um algoritmo que toma um problema como entrada e retorna a solução para o problema, geralmente após resolver um número possível de soluções.

Uma solução, no aspecto de função intermediária, é um método o qual um algoritmo externo, ou mais abrangente, utilizará para solucionar um determinado problema. Esta solução é representada por elementos de um espaço de busca, definido por uma fórmula matemática ou um procedimento, tal como as raízes de uma equação com números inteiros variáveis, ou uma combinação dos dois, como os circuitos hamiltonianos de um grafo.

Já pelo aspecto de uma estrutura de dados, sendo o modelo de explanação inicial do assunto, a busca é um algoritmo projetado para encontrar um item com propriedades especificadas em uma coleção de itens. Os itens podem ser armazenadas individualmente, como registros em um banco de dados.

A maioria dos algoritmos estudados por cientistas da computação que resolvem problemas são algoritmos de busca.

Os algoritmos de busca têm como base o método de procura de qualquer elemento dentro de um conjunto de elementos com determinadas propriedades. Que podiam ser livros nas bibliotecas, ou dados cifrados, usados principalmente durante as duas grandes guerras. Seus formatos em linguagem computacional vieram a se desenvolver juntamente com a construção dos primeiros computadores. Sendo que a maioria de suas publicações conhecidas começa a surgir a partir da década de 1970. Atualmente os algoritmos de busca são a base de motores de buscas da Internet

### **Classes de Algoritmos de Busca**

Algoritmos para a busca de espaços virtuais são usados em problema de satisfação de restrição, onde o objetivo é encontrar um conjunto de atribuições de valores para certas variáveis que irão satisfazer específicas equações e inequações matemáticas. Eles também são utilizados quando o objetivo



é encontrar uma atribuição de variável que irá maximizar ou minimizar uma determinada função dessas variáveis. Algoritmos para estes problemas incluem a base de busca por força bruta (também chamado de "ingênua" ou busca "desinformada"), e uma variedade de heurísticas que tentam explorar o conhecimento parcial sobre a estrutura do espaço, como relaxamento linear, geração de restrição, e propagação de restrições.

Algumas subclasses importantes são os métodos de busca local, que vêem os elementos do espaço de busca como os vértices de um grafo, com arestas definidas por um conjunto de heurísticas aplicáveis ao caso, e fazem a varredura do espaço, movendo-se de item para item ao longo das bordas, por exemplo de acordo com o declive máximo ou com o critério da melhor escolha, ou em uma busca estocástica. Esta categoria inclui uma grande variedade de métodos metaheurísticos gerais, como arrefecimento simulado, pesquisa tabu, times assíncronos, e programação genética, que combinam heurísticas arbitrárias de maneiras específicas.

Esta classe também inclui vários algoritmos de árvore de busca, que vêem os elementos como vértices de uma Árvore (teoria dos grafos) árvore, e atravessam-na em alguma ordem especial. Exemplos disso incluem os métodos exaustivos, como em busca em profundidade e em busca em largura, bem como vários métodos de busca por poda de árvore baseados em heurística como retrocesso e ramo e encadernado. Ao contrário das metaheurísticas gerais, que trabalham melhor apenas no sentido probabilístico, muitos destes métodos de árvore de busca têm a garantia de encontrar a solução exata ou ideal, se for dado tempo suficiente.

Outra importante sub-classe consiste de algoritmos para explorar a árvore de jogo de jogos para múltiplos participantes (multiplayer), como xadrez ou gamão, cujos nós consistem em todas as situações de jogo possíveis que poderiam resultar da situação atual. O objetivo desses problemas é encontrar o movimento que oferece a melhor chance de uma vitória, tendo em conta todos os movimentos possíveis do(s) adversário(s). Problemas similares ocorrem quando as pessoas, ou máquinas, têm de tomar decisões sucessivas cujos resultados não estão totalmente sob seu controle, como em um robô, ou na orientação de marketing, financeira ou de planejamento estratégico militar. Este tipo de problema - pesquisa combinatória - tem sido extensivamente estudado no contexto da inteligência artificial. Exemplos de algoritmos para esta classe são o algoritmo minimax, poda alfa-beta e o algoritmo A\*.

### **Para Subestruturas de Uma dada Estrutura**

O nome de pesquisa combinatória é geralmente usado para os algoritmos que procuram uma subestrutura específica de uma dada estrutura discreta, tais como um grafo, uma cadeia de caracteres, um grupo (matemática) finito, e assim por diante. O termo otimização combinatória é normalmente utilizado quando o objetivo é encontrar uma subestrutura com um valor máximo (ou mínimo) de algum parâmetro. (Uma vez que a subestrutura normalmente é representada no computador por um conjunto de variáveis de inteiros com restrições, estes problemas podem ser vistos como casos especiais de satisfação restrita ou otimização discreta, mas eles geralmente são formulados e resolvidos em um ambiente mais abstrato onde a representação interna não é explicitamente mencionada.)

Uma subclasse importante e extensivamente estudada são os algoritmos de grafos, em particular algoritmos de travessia de grafo, para encontrar determinadas subestruturas em um dado grafo - como subgrafos, caminhos, circuitos, e assim por diante. Exemplos incluem o algoritmo de Dijkstra, algoritmo de Kruskal, o algoritmo do vizinho mais próximo, e algoritmo de Prim.

Outra subclasse importante desta categoria são os algoritmos de busca de cadeia de caracteres, que busca de padrões dentro de expressões. Dois exemplos famosos são os algoritmos Boyer-Moore e Knuth-Morris-Pratt, e vários algoritmos baseados na estrutura de dados árvore de sufixo.

### **Busca Pelo Máximo de Uma Função**

Em 1953, Kiefer concebeu a busca de Fibonacci, que pode ser utilizada para encontrar o máximo de uma função unimodal e tem muitas outras aplicações em ciências da computação.

### **Para Computadores Quânticos**

Há também métodos de busca projetados para computadores quânticos, como o algoritmo de Grover, que são teoricamente mais rápidos do que a busca linear ou força bruta mesmo sem a ajuda de estruturas de dados ou heurísticas.

## Algoritmo De Ordenação

Algoritmo de ordenação em ciência da computação é um algoritmo que coloca os elementos de uma dada sequência em uma certa ordem -- em outras palavras, efetua sua ordenação completa ou parcial. As ordens mais usadas são a numérica e a lexicográfica.

Existem várias razões para se ordenar uma sequência. Uma delas é a possibilidade de acessar seus dados de modo mais eficiente.

### Métodos de Ordenação de Vetores

#### Métodos Simples

Insertion sort

Selection sort

Bubble sort

Comb sort

#### Métodos Sofisticados

Merge sort

Heapsort

Shell sort

Radix sort

Gnome sort

Counting sort

Bucket sort

Cocktail sort

Timsort

Quick sort

#### Métodos de Pesquisa

Pesquisa binária

Busca linear

BogoBusca

#### Listas Lineares e Suas Generalizações

Lista linear é uma estrutura de dados na qual elementos de um mesmo tipo de dado estão organizados de maneira sequencial. Não necessariamente, estes elementos estão fisicamente em sequência, mas a idéia é que exista uma ordem lógica entre eles. Um exemplo disto seria um consultório médico: as pessoas na sala de espera estão sentadas em qualquer lugar, porém sabe-se quem é o próximo a ser atendido, e o seguinte, e assim por diante. Assim, é importante ressaltar que uma lista linear permite representar um conjunto de dados afins (de um mesmo tipo) de forma a preservar a relação de ordem entre seus elementos. Cada elemento da lista é chamado de nó, ou nodo.

Definição:

Conjunto de  $N$  nós, onde  $N \geq 0$ ,  $x_1, x_2, \dots, x_n$ , organizados de forma a refletir a posição relativa dos

mesmos. Se  $N \geq 0$ , então  $x_1$  é o primeiro nó. Para  $1 < k < n$ , o nó  $x_k$  é precedido pelo nó  $x_{k-1}$  e seguido pelo nó  $x_{k+1}$  e  $x_n$  é o último nó. Quando  $N = 0$ , diz-se que a lista está vazia. Exemplos de listas lineares:

Pessoas na fila de um banco;

Letras em uma palavra;

Relação de notas dos alunos de uma turma;

Itens em estoque em uma empresa;

Dias da semana;

Vagões de um trem;

Pilha de pratos;

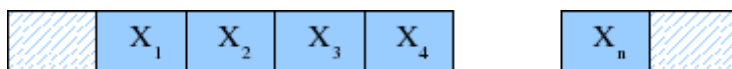
Cartas de baralho.

Alocação De Uma Lista

Quanto a forma de alocar memória para armazenamento de seus elementos, uma lista pode ser:

Sequencial Ou Contígua

Numa lista linear contígua, os nós além de estarem em uma sequência lógica, estão também fisicamente em sequência. A maneira mais simples de acomodar uma lista linear em um computador é através da utilização de um vetor.



A representação por vetor explora a sequencialidade da memória de tal forma que os nós de uma lista sejam armazenados em endereços contíguos.

Encadeada

Os elementos não estão necessariamente armazenados sequencialmente na memória, porém a ordem lógica entre os elementos que compõem a lista deve ser mantida. Listas lineares encadeadas são discutida na aula 11.

Operações Com Listas

As operações comumente realizadas com listas são:

Criação de uma lista

Remoção de uma lista

Inserção de um elemento da lista

Remoção de um elemento da lista

Acesso de um elemento da lista

Alteração de um elemento da lista

Combinação de duas ou mais listas

Classificação da lista

Cópia da lista

Localizar nodo através de info

Tipos de Listas Lineares

Os tipos mais comuns de listas lineares são as:

Pilhas

Uma pilha é uma lista linear do tipo LIFO - Last In First Out, o último elemento que entrou, é o primeiro a sair. Ela possui apenas uma entrada, chamada de topo, a partir da qual os dados entram e saem dela. Exemplos de pilhas são: pilha de pratos, pilha de livros, pilha de alocação de variáveis da memória, etc.

Filas

Uma fila é uma lista linear do tipo FIFO - First In First Out, o primeiro elemento a entrar será o primeiro a sair. Na fila os elementos entram por um lado ("por trás") e saem por outro ("pela frente"). Exemplos de filas são: a fila de caixa de banco, a fila do INSS, etc.

Deque

Um deque - Double-Ended QUEUE) é uma lista linear na qual os elementos entram e saem tanto pela "pela frente" quanto "por trás". Pode ser considerada uma generalização da fila.

Em ciência da computação, uma lista ou sequência é uma estrutura de dados abstrata que implementa uma coleção ordenada de valores, onde o mesmo valor pode ocorrer mais de uma vez. Uma instância de uma lista é uma representação computacional do conceito matemático de uma sequência finita, que é, uma tupla. Cada instância de um valor na lista normalmente é chamado de um item, entrada ou elemento da lista. Se o mesmo valor ocorrer várias vezes, cada ocorrência é considerada um item distinto.



Uma estrutura de lista encadeada isoladamente, implementando uma lista com 3 elementos inteiros.

O nome lista também é usado para várias estruturas de dados concretas que podem ser usadas para implementar listas abstratas, especialmente listas encadeadas.

As chamadas estruturas de lista estática' permitem apenas a verificação e enumeração dos valores. Uma lista mutável ou dinâmica pode permitir que itens sejam inseridos, substituídos ou excluídos durante a existência da lista.

Muitas linguagens de programação fornecem suporte para tipos de dados lista e possuem sintaxe e semântica especial para listas e operações com listas. Uma lista pode frequentemente ser construída escrevendo-se itens em sequência, separados por vírgulas, ponto e vírgulas ou espaços, dentro de um par de delimitadores como parênteses '()', colchetes '[]', chaves '{}' ou chevrons '<>'. Algumas linguagens podem permitir que tipos lista sejam indexados ou cortados como os tipos vetor.

Em linguagens de programação orientada a objetos, listas normalmente são fornecidas como instâncias ou subclasses de uma classe "lista" genérica. Tipos de dado lista são frequentemente implementados usando arrays ou listas encadeadas de algum tipo, mas outras estruturas de dados podem ser mais apropriadas para algumas aplicações. Em alguns contextos, como em programação Lisp, o termo lista pode se referir especificamente à lista encadeada em vez de um array.

É forma de organização através da enumeração de dados para melhor visualização da informação. Em informática, o conceito expande-se para uma estrutura de dados dinâmica, em oposição aos vetores, que são estruturas de dados estáticas. Assim, uma lista terá virtualmente infinitos elementos.

Numa lista encadeada existem dois campos. Um campo reservado para colocar o dado a ser armazenado e outro campo para apontar para o próximo elemento da lista. Normalmente a implementação é feita com ponteiros.

Existem vários tipos de implementação de listas como estruturas de dados:

Listas duplamente ligadas

Listas FIFO, ou filas (First In First Out - primeiro a entrar, primeiro a sair).

Listas LIFO, ou pilhas (Last In First Out - último a entrar, primeiro a sair).

### Características

Listas possuem as seguintes características:

Tamanho da lista significa o número de elementos presentes na lista. Listas encadeadas tem a vantagem de ter um tamanho variável, novos itens podem ser adicionados, o que aumentando seu tamanho.

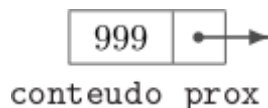
Cada elemento numa lista possui um índice, um número que identifica cada elemento da lista. Usando o índice de um elemento da lista é possível buscá-lo ou removê-lo.

Uma lista encadeada é uma representação de uma sequência de objetos, todos do mesmo tipo, na memória RAM (= random access memory) do computador. Cada elemento da sequência é armazenado em uma célula da lista: o primeiro elemento na primeira célula, o segundo na segunda e assim por diante.

### Estrutura de Uma Lista Encadeada

Uma lista encadeada (= linked list = lista ligada) é uma sequência de células; cada célula contém um objeto de algum tipo e o endereço da célula seguinte. Suporemos neste capítulo que os objetos armazenados nas células são do tipo int. Cada célula é um registro que pode ser definido assim:

```
struct reg {  
    int conteudo;  
    struct reg *prox;};
```



É conveniente tratar as células como um novo tipo-de-dados e atribuir um nome a esse novo tipo:

```
typedef struct reg celula; // célula
```

Uma célula c e um ponteiro p para uma célula podem ser declarados assim:

```
celula c;  
celula *p;
```

Se c é uma célula então c.conteudo é o conteúdo da célula e c.prox é o endereço da próxima célula. Se p é o endereço de uma célula, então p->conteudo é o conteúdo da célula e p->prox é o endereço da próxima célula. Se p é o endereço da última célula da lista então p->prox vale NULL .



(A figura pode dar a falsa impressão de que as células da lista ocupam posições consecutivas na memória. Na realidade, as células estão tipicamente espalhadas pela memória de maneira imprevisível.)

### Árvores e Suas Generalizações

Árvore binária é uma estrutura de dados caracterizada por:

Ou não tem elemento algum (árvore vazia).

Ou tem um elemento distinto, denominado raiz, com dois apontamentos para duas estruturas diferentes, denominadas sub-árvore esquerda e sub-árvore direita.

Perceba que a definição é recursiva e, devido a isso, muitas operações sobre árvores binárias utilizam recursão. É o tipo de árvore mais utilizado na computação. A principal utilização de árvores binárias são as árvores de busca.

Implementação de um Nó de uma árvore binária em C++

```
class No
```

```
private:
```

```
/* Aqui vão as informações do nó, no nosso caso usaremos apenas um inteiro como chave,
```

```
mas poderíamos usar qualquer outra informação junto com as chaves como strings, booleanos etc... */
```

```
int chave;
```

```
// Ponteiros do tipo Nó para as sub-arvores direitas e esquerdas respectivamente
```

```
No *esq;
```

```
No *dir;
```

```
public:
```

```
/* Aqui criaremos um construtor para o nó que seta o valor da chave e inicializa os ponteiros
```

```
das sub-arvores como null. OBS: Note que usarei nullptr em vez de NULL, se não utilizar o c++11  
essa keyword
```

```
não funcionará. */
```

```
No ( int chave )
```

```
{ this->chave = chave; // seta a chave
```

```
esq = nullptr; // inicializa a sub-arvore esquerda como null.
```

```
dir = nullptr; // inicializa a sub-arvore direita como null.}
```

```
// METODOS GETS E SETTERS.
```

```
int getChave()
```

```
{ return chave; }
```

```
No* getEsq()
```

```
{ return esq; }
```

```
No* getDir()
```

```
{ return dir; }
```

```
void setEsq( No* esq )
```

```
{this->esq = esq;}
```

```
void setDir( No* dir )
```

```
{ this->dir = dir; }
```

### Definições Para Árvores Binárias

Os nós de uma árvore binária possuem graus zero, um ou dois. Um nó de grau zero é denominado folha.

Uma árvore binária é considerada estritamente binária se cada nó da árvore possui grau zero ou dois.

A profundidade de um nó é a distância deste nó até a raiz. Um conjunto de nós com a mesma profundidade é denominado nível da árvore. A maior profundidade de um nó, é a altura da árvore.

Uma árvore é dita completa se todas as folhas da árvore estão no mesmo nível da árvore.

### Definições em Teoria dos Grafos

Em teoria dos grafos, uma árvore binária é definida como um grafo acíclico, conexo, dirigido e que cada nó não tem grau maior que 3. Assim sendo, só existe um caminho entre dois nós distintos.

E cada ramo da árvore é um vértice dirigido, sem peso, que parte do pai e vai o filho.

### Árvore Binária de Busca

Em Ciência da computação, uma árvore binária de busca (ou árvore binária de pesquisa) é uma estrutura de dados de árvore binária baseada em nós, onde todos os nós da subárvore esquerda possuem um valor numérico inferior ao nó raiz e todos os nós da subárvore direita possuem um valor superior ao nó raiz (esta é a forma padrão, podendo as subárvores serem invertidas, dependendo da aplicação).

O objetivo desta árvore é estruturar os dados de forma a permitir busca binária.

Seja  $S = \{s_1, s_2, \dots, s_n\}$  um conjunto de chaves tais que  $s_1 < s_2 \dots s_n$ . Seja  $k$  um valor dados. Deseja-

se verificar se  $k \in S$  e identificar o índice  $i$  tal que  $k = s_i$ .

A Árvore Binária de Busca (ABB) resolve os problemas propostos. A figura ilustra uma ABB.

Uma ABB é uma árvore binária rotulada  $T$  com as seguintes propriedades:

$T$  possui  $n$  nós. Cada nó  $u$  armazena uma chave distinta  $s_j \in S$  e tem como rótulo o valor  $r(u) = s_j$ .

Para cada nó  $v$  de  $T$   $r(v_1) < r(v)$  e  $r(v_2) > r(v)$ , onde  $v_1$  pertence à subárvore esquerda de  $v$  e  $v_2$  pertence à subárvore direita de  $v$ .

Dado o conjunto  $S$  com mais de um elemento, existem várias ABB que resolvem o problema.

#### Elementos

Nós - são todos os itens guardados na árvore

Raiz - é o nó do topo da árvore (no caso da figura acima, a raiz é o nó 8)

Filhos - são os nós que vem depois dos outros nós (no caso da figura acima, o nó 6 é filho do 3)

Pais - são os nós que vem antes dos outros nós (no caso da figura acima, o nó 10 é pai do 14)

Folhas - são os nós que não têm filhos; são os últimos nós da árvore (no caso da figura acima, as folhas são 1, 4, 7 e 13)

#### Complexidade

A complexidade das operações sobre ABB depende diretamente da altura da árvore.

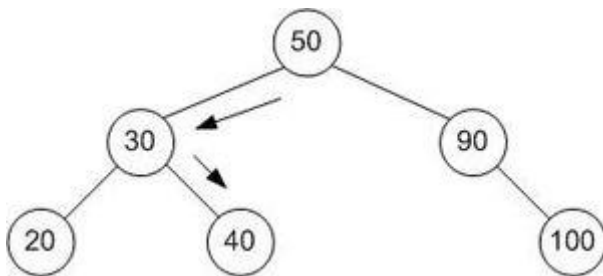
Uma árvore binária de busca com chaves aleatórias uniformemente distribuídas tem altura  $O(\log n)$ .

No pior caso, uma ABB poderá ter altura  $O(n)$ . Neste caso a árvore é chamada de árvore zig-zag e corresponde a uma degeneração da árvore em lista encadeada.

Em função da observação anterior, a árvore binária de busca é de pouca utilidade para ser aplicada em problemas de busca em geral. Daí o interesse em árvores balanceadas, cuja altura seja  $O(\log n)$  no pior caso

Operações

Busca



Buscando um valor na árvore binária.

A busca em uma árvore binária por um valor específico pode ser um processo recursivo ou iterativo. Será apresentado um método recursivo.

A busca começa examinando o nó raiz. Se a árvore está vazia, o valor procurado não pode existir na árvore. Caso contrário, se o valor é igual a raiz, a busca foi bem-sucedida. Se o valor é menor do que a raiz, a busca segue pela subárvore esquerda. Similarmente, se o valor é maior do que a raiz, a busca segue pela subárvore direita. Esse processo é repetido até o valor ser encontrado ou a subárvore ser nula (vazia). Se o valor não for encontrado até a busca chegar na subárvore nula, então o valor não deve estar presente na árvore.

Segue abaixo o algoritmo de busca implementado na linguagem Python:

```

# 'no' refere-se ao nó-pai, neste caso
def arvore_binaria_buscar (no, valor):
    if no is None:
        # valor não encontrado
        return None
    else:
        if valor == no.valor:
            # valor encontrado
            return no.valor
        elif valor < no.valor:
            # busca na subárvore esquerda
            return arvore_binaria_buscar(no.filho_esquerdo, valor)
        elif valor > no.valor:
  
```



# busca na subárvore direita

```
return arvore_binaria_buscar(no.filho_direito, valor)
```

Essa operação poderá ser  $O(\log n)$  em algumas situações, mas necessita  $O(n)$  de tempo no pior caso, quando a árvore assumir a forma de lista ligada (árvore zig-zag).

### Inserção

A inserção começa com uma busca, procurando pelo valor, mas se não for encontrado, procuram-se as subárvores da esquerda ou direita, como na busca. Eventualmente, alcança-se a folha, inserindo-se então o valor nesta posição. Ou seja, a raiz é examinada e introduz-se um nó novo na subárvore da esquerda se o valor novo for menor do que a raiz, ou na subárvore da direita se o valor novo for maior do que a raiz. Abaixo, um algoritmo de inserção em Python:

```
def arvore_binaria_inserir(no, chave, valor):
```

```
    if no is None:
```

```
        return TreeNode(None, chave, valor, None)
```

```
    if chave == no.chave:
```

```
        return TreeNode(no.filho_esquerdo, chave, valor, no.filho_direito)
```

```
    if chave < no.chave:
```

```
        return TreeNode(arvore_binaria_inserir(no.filho_esquerdo, chave, valor), no.chave, no.valor, no.filho_direito)
```

```
    else:
```

```
        return TreeNode(no.filho_esquerdo, no.chave, no.valor, arvore_binaria_inserir(no.filho_direito, chave, valor))
```

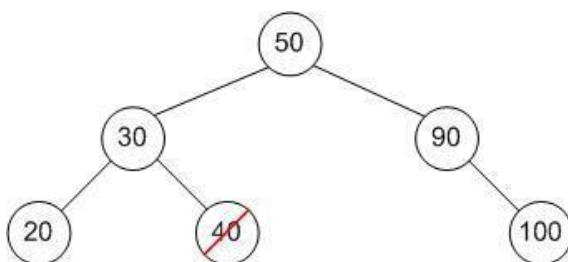
Esta operação requer  $O(\log n)$  vezes para o caso médio e necessita de  $O(n)$  no pior caso. A fim de introduzir um nó novo na árvore, seu valor é primeiro comparado com o valor da raiz. Se seu valor for menor que a raiz, é comparado então com o valor do filho da esquerda da raiz. Se seu valor for maior, está comparado com o filho da direita da raiz. Este processo continua até que o nó novo esteja comparado com um nó da folha, e então adiciona-se o filho da direita ou esquerda, dependendo de seu valor.

### Remoção

A exclusão de um nó é um processo mais complexo. Para excluir um nó de uma árvore binária de busca, há de se considerar três casos distintos para a exclusão:

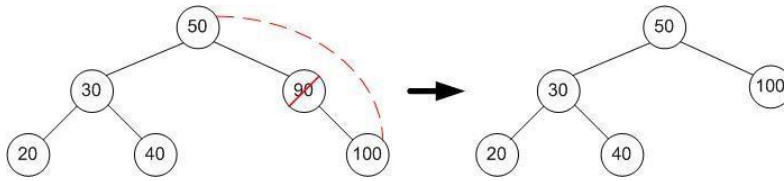
#### Remoção Na Folha

A exclusão na folha é a mais simples, basta removê-lo da árvore.



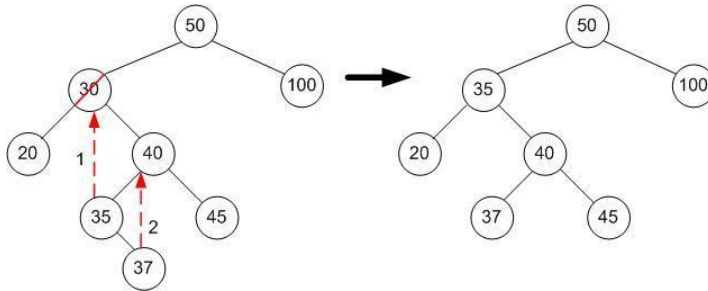
#### Remoção De Nó Com Um Filho

Excluindo-o, o filho sobe para a posição do pai.



### Remoção de Nó com Dois Filhos

Neste caso, pode-se operar de duas maneiras diferentes. Pode-se substituir o valor do nó a ser retirado pelo valor sucessor (o nó mais à esquerda da subárvore direita) ou pelo valor antecessor (o nó mais à direita da subárvore esquerda), removendo-se aí o nó sucessor (ou antecessor).



No exemplo acima, o nó de valor 30 está para ser removido, e possui como sucessor imediato o valor 35 (nó mais à esquerda da sua sub-árvore direita). Assim sendo, na exclusão, o valor 35 será promovido no lugar do nó a ser excluído, enquanto a sua sub-árvore (direita) será promovida para sub-árvore esquerda do 40, como pode ser visto na figura.

Exemplo de algoritmo de exclusão em Python:

```
def exclusao_em_arvore_binaria(nó_arvore, valor):
```

```
    if nó_arvore is None: return None # Valor não encontrado
```

```
    esquerda, nó_valor, direita = nó_arvore.esquerda, nó_arvore.valor, nó_arvore.direita
```

```
    if nó_valor == valor:
```

```
        if esquerda is None:
```

```
            return direita
```

```
        elif direita is None:
```

```
            return esquerda
```

```
        else:
```

```
            valor_max, novo_esquerda = busca_max(esquerda)
```

```
            return TreeNode(novo_esquerda, valor_max, direita)
```

```
        elif valor < nó_valor:
```

```
            return TreeNode(exclusao_em_arvore_binaria(esquerda, valor), nó_valor, direita)
```

```
        else:
```

```
            return TreeNode(esquerda, nó_valor, exclusao_em_arvore_binaria(direita, valor))
```

```
def busca_max(nó_arvore):
```

```
    esquerda, nó_valor, direita = nó_arvore.esquerda, nó_arvore.valor, nó_arvore.direita
```

if direita is None: return (nó\_valor, esquerda)

else:

(valor\_max, novo\_direita) = busca\_max(direita)

return (valor\_max, (esquerda, nó\_valor, novo\_direita))

Embora esta operação não percorra sempre a árvore até uma folha, esta é sempre uma possibilidade; assim, no pior caso, requer o tempo proporcional à altura da árvore, visitando-se cada nó somente uma única vez.

## Aplicações

### Percursos Em ABB

Em uma árvore binária de busca podem-se fazer os três percursos que se fazem para qualquer árvore binária (percursos em inordem, pré-ordem e pós-ordem). É interessante notar que, quando se faz um percurso em ordem em uma árvore binária de busca, os valores dos nós aparecem em ordem crescente. A operação "Percorre" tem como objetivo percorrer a árvore numa dada ordem, enumerando os seus nós. Quando um nó é enumerado, diz-se que ele foi "visitado".

Pré-ordem (ou profundidade):

Visita a raiz

Percorre a subárvore esquerda em pré-ordem

Percorre a subárvore direita em pré-ordem

Ordem Simétrica:

Percorre a subárvore esquerda em ordem simétrica

Visita a raiz

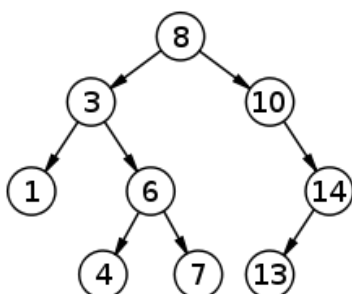
Percorre a subárvore direita em ordem simétrica

Pós-ordem:

Percorre a subárvore esquerda em pós-ordem

Percorre a subárvore direita em pós-ordem

Visita a raiz



Árvore Binária De Busca

Usando a ABB acima, os resultados serão os seguintes:

Pré-ordem => 8, 3, 1, 6, 4, 7, 10, 14, 13

Ordem simétrica => 1, 3, 4, 6, 7, 8, 10, 13, 14 (chaves ordenadas)

Pós-ordem => 1, 4, 7, 6, 3, 13, 14, 10, 8

**Ordenação**

Uma árvore binária de busca pode ser usada para ordenação de chaves. Para fazer isto, basta inserir todos os valores desejados na ABB e executar o percurso em ordem simétrica.

```
def criar_arvore_binaria(valor):
    arvore = None

    for v in valor:
        arvore = arvore_binaria_de_insercao(arvore, v)

    return arvore

def arvore_binaria_transversal(nó_arvore):
    if nó_arvore is None: return []

    else:
        esquerda, valor, direita = nó_arvore

    return (arvore_binaria_transversal(esquerda) + [valor] + arvore_binaria_transversal(direita))
```

Criar ABB tem complexidade  $O(n^2)$  no pior caso. A geração de um vetor de chaves ordenadas tem complexidade  $O(n)$ . O algoritmo de ordenação terá complexidade final  $O(n^2)$  no pior caso.

Cabe observar que há algoritmos de ordenação dedicados com complexidade  $O(n \cdot \log n)$ , com desempenho superior ao proposto neste tópico.

**Árvore AVL**

Árvore AVL			
Tipo	Árvore		
Ano	1962		
Inventado por	Georgy Adelson-Velsky e Yevgeniy Landis		
Complexidade de Tempo em Notação big O			
	Algoritmo	Caso Médio	Pior Caso
Espaço	$O(n)$		$O(n)$
Busca	$O(\log n)[1]$		$O(\log n)[1]$
Inserção	$O(\log n)[1]$		$O(\log n)[1]$
Remoção	$O(\log n)[1]$		$O(\log n)[1]$

Árvore AVL é uma árvore binária de busca balanceada, ou seja, uma árvore balanceada (árvore completa) são as árvores que minimizam o número de comparações efetuadas no pior caso para uma busca com chaves de probabilidades de ocorrências idênticas. Contudo, para garantir essa propriedade em aplicações dinâmicas, é preciso reconstruir a árvore para seu estado ideal a cada operação sobre seus nós (inclusão ou exclusão), para ser alcançado um custo de algoritmo com o tempo de pesquisa tendendo a  $O(\log N)$ .

As operações de busca, inserção e remoção de elementos possuem complexidade  $O(\log n)$  (no qual  $n$  é o número de elementos da árvore), que são aplicados a árvore de busca binária.

O nome AVL vem de seus criadores soviéticos Adelson Velsky e Landis, e sua primeira referência encontra-se no documento "Algoritmos para organização da informação" de 1962.

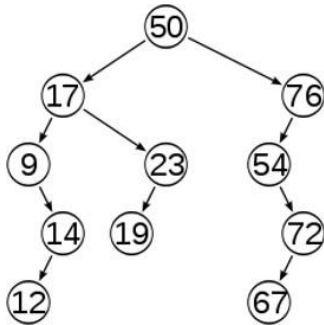
Nessa estrutura de dados cada elemento é chamado de nó. Cada nó armazena uma chave e dois ponteiros, uma para a subárvore esquerda e outro para a subárvore direita.

### História

Esta estrutura foi criada em 1962 pelos soviéticos Adelson Velsky e Landis que a criaram para que fosse possível inserir e buscar um elemento em tempo  $c \cdot \log(n)$  operações, onde  $n$  é o número de elementos contido na árvore. Tal estrutura foi a primeira árvore binária balanceada criada.

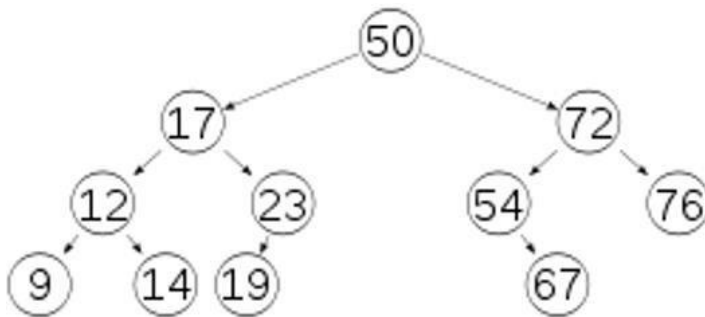
### Conceitos Básicos

#### Definição



#### Uma Árvore Não AVL

Uma árvore binária  $T$  é denominada AVL quando, para qualquer nó de  $T$ , as alturas de suas duas subárvores, esquerda e direita, diferem em módulo de até uma unidade.



#### Uma Árvore AVL

Pela definição fica estabelecido que todos os nós de uma árvore AVL devem respeitar a seguinte propriedade:

$|hd(u) - he(u)| \leq 1$ , onde  $hd(u)$  é a altura da subárvore direita do nó  $u$  e  $he(u)$  é a altura da subárvore esquerda do nó  $u$ .

O valor  $hd(u) - he(u)$  é denominado fator de balanço do nó. Quando um nó possui fator de balanço com valor  $-1$ ,  $0$  ou  $1$  então o mesmo é um nó regulado. Todos os nós de uma árvore AVL são regulados, caso contrário a árvore não é AVL.

### Estrutura

Proposta de estrutura dos nós de uma árvore AVL básica, com chave do tipo inteiro:

tipo No\_AVL = registro

chave: inteiro;

fb: inteiro; // Fator de Balanço

```
esq: ^No_AVL; // aponta subárvore esquerda
```

```
dir: ^No_AVL; // aponta subárvore direita
```

```
fim;
```

O campo chave armazena o valor da chave. Os campos esq e dir são ponteiros para as subárvores esquerda e direita, respectivamente. O campo fb armazena o fator de balanço.

Definição da estrutura da árvore:

```
tipo Arvore_AVL = registro
```

```
raiz: ^No_AVL;
```

```
// definição de outros campos de interesse
```

```
fim;
```

### **Balanceamento**

Toda árvore AVL é balanceada, isto é, sua altura é  $O(\log n)$ .

A vantagem do balanceamento é possibilitar que a busca seja de complexidade  $O(\log n)$ . Entretanto, as operações de inserção e remoção devem possuir custo similar. No caso da árvore AVL, a inserção e remoção têm custo  $O(\log n)$ .

Por definição, todos os nós da AVL devem ter  $fb = -1, 0$  ou  $1$ .

Para garantir essa propriedade, a cada inserção ou remoção o fator de balanço deve ser atualizado a partir do pai do nó inserido até a raiz da árvore. Na inserção basta encontrar o primeiro nó desregulado ( $fb = -2$  ou  $fb = 2$ ), aplicar a operação de rotação necessária, não havendo necessidade de verificar os demais nós. Na remoção a verificação deverá prosseguir até a raiz, podendo requerer mais de uma rotação.

### **Complexidade**

A árvore AVL tem complexidade  $O(\log n)$  para todas operações e ocupa espaço  $n$ , onde  $n$  é o número de nós da árvore.

	<b>Média</b>	<b>Pior Caso</b>
<b>Espaço</b>	$O(n)$	$O(n)$
<b>Busca</b>	$O(\log n)$	$O(\log n)$
<b>Inserção</b>	$O(\log n)$	$O(\log n)$
<b>Deleção</b>	$O(\log n)$	$O(\log n)$

Complexidade da árvore AVL em notação  $O$ .

### **Operações**

#### **Busca**

A busca é a mesma utilizada em árvore binária de busca.

A busca pela chave de valor  $K$  inicia sempre pelo nó raiz da árvore.

Seja  $pt\_u$  um ponteiro para o nó  $u$  sendo verificado. Caso o  $pt\_u$  seja nulo então a busca não foi bem sucedida ( $K$  não está na árvore ou árvore vazia). Verificar se a chave  $K$  igual  $pt\_u \rightarrow$  chave (valor chave armazenado no nó  $u$ ), então a busca foi bem sucedida. Caso contrário, se  $K < pt\_u \rightarrow$  chave então a busca segue pela subárvore esquerda; caso contrário, a busca segue pela subárvore direita.

## Algoritmo De Busca

```
busca_AVL(@pt_u:^no_AVL, K:inteiro):logico;
```

```
inicio
```

```
se pt_u é NULO então retornar Falso;
```

```
se K = pt_u->chave então retornar Verdadeiro;
```

```
senão se K < pt_u->chave então
```

```
    retornar busca_AVL(K, u->esq);
```

```
senão retornar busca_AVL(K, u->dir);
```

```
fim.
```

## Exemplo De Algoritmo De Busca Em Java.

// O método de procura numa AVL é semelhante ao busca binária de uma árvore binária de busca comum.

```
public BSTNode<T> search(T element) {  
    return search(element, this.root);  
}  
// Método auxiliar à recursão.  
private BSTNode<T> search(T element, BSTNode<T> node) {  
    if (element == null || node.isEmpty()) {  
        return new BSTNode<T>();  
    }  
    if (node.isEmpty() || node.getData().equals(element)) {  
        return node;  
    } else if (node.getData().compareTo(element) > 0) {  
        return search(element, node.getLeft());  
    } else {  
        return search(element, node.getRight());  
    }  
}
```

## Inserção

Para inserir um novo nó de valor K em uma árvore AVL é necessária uma busca por K nesta mesma árvore. Após a busca o local correto para a inserção do nó K será em uma subárvore vazia de uma folha da árvore. Depois de inserido o nó, a altura do nó pai e de todos os nós acima deve ser atualizada. Em seguida o algoritmo de rotação simples ou dupla deve ser acionado para o primeiro nó pai desregulado.

## Algoritmo Recursivo

```
inserir_AVL(@p:^no_AVL, K:inteiro, @mudou_h:logico):logico;
```

```
var aux: logico; // variável local auxiliar;
```

```
inicio
```

```
se p = NULO então
```

```
inicio // não achou chave, inserir neste local

p := novo(no_AVL); // aloca novo nó dinamicamente, diretamente na subárvore do nó pai

p^.chave := K;

p^.esq := NULO;

p^.dir := NULO;

mudou_h := Verdadeiro; // sinalizar que a altura da subárvore mudou 1 unidade

retornar Verdadeiro;

fim;

se K < p^.chave então // inserir recursivamente na subárvore esquerda

se inserir_AVL(p^.esq, K, mudou_h) então // ocorreu inserção na subárvore esquerda

inicio

se mudou_h então

inicio // mudou altura da subárvore esquerda de p

p^.fb := p^.fb - 1; // fator de balanço decrementa 1 unidade

caso p^.fb

-2: p := rotacao_direita(p); mudou_h := Falso;

0: mudou_h := Falso; // não mudou a altura da subárvore de raiz p

// -1: mudou_h := Verdadeiro; // desnecessário pois mudou_h já é Verdadeiro

fim

retornar Verdadeiro;

fim

senão

se K > p^.chave então // ocorreu inserção na subárvore direita

se inserir_AVL(p^.dir, K, mudou_h) então // ocorreu inserção

inicio

se mudou_h então

inicio // mudou altura da subárvore esquerda de p

p^.fb := p^.fb + 1; // fator de balanço incrementa 1 unidade

caso p^.fb

2: p := rotacao_esquerda(p); mudou_h := Falso;

0: mudou_h := Falso; // não mudou a altura da subárvore de raiz p

// 1: mudou_h := Verdadeiro; // desnecessário pois mudou_h já é Verdadeiro

fim
```



retornar Verdadeiro;

fim

retornar Falso;

fim.

Os parâmetros `p` e `mudou_h` são passados por referência. O ponteiro `p` aponta para o nó atual. O parâmetro `mudou_h` é do tipo lógico e informa ao chamador se a subárvore apontada por `p` mudou sua altura.

Como Identificar Mudança De Altura?

Considerar que o nó `p` é raiz da subárvore `Tp` e houve inserção em uma de suas subárvores.

Caso a subárvore `Tp` tenha mudado de altura, decrementar `fb` (inserção na subárvore esquerda) ou incrementar `fb` (inserção na subárvore direita).

Caso 1: Ao inserir um nó folha, a subárvore `Tp` passa de altura 0 para altura 1, então `Tp` mudou de altura.

Caso 2: `fb=0` antes da inserção foi alterado para 1 ou -1, então a subárvore `Tp` mudou de altura.

Caso 3: `fb=1` ou -1 antes da inserção, passou a ter valor 0, então a subárvore `Tp` não mudou de altura.

Caso 4: O `fb` passou a ter valor -2 ou 2 após a inserção, então há necessidade de aplicação de alguma operação de rotação. Após a rotação, a subárvore `Tp` terá a mesma altura anterior à inserção.

Exemplo De Algoritmo De Inserção Em Java

/\* Por definição, a árvore AVL é uma árvore binária de busca (BST).

\* Por este motivo utiliza-se aqui a mesma definição (classe) de Nós que uma BST simples.

```
public void insert(T element) {
    insertAux(element);
    BSTNode<T> node = search(element); // Pode-se utilizar o mesmo search exemplificado acima.
    rebalanceUp(node);
    private void insertAux(T element) {
        if (element == null) return;
        insert(element, this.root);
        private void insert(T element, BSTNode<T> node) {
            if (node.isEmpty()) {
                node.setData(element);
                node.setLeft(new BSTNode<T>());
                node.setRight(new BSTNode<T>());
                node.getLeft().setParent(node);
                node.getRight().setParent(node);
            }
        }
    }
}
```

```
} else {  
if (node.getData().compareTo(element) < 0) {  
insert(element, node.getRight());  
} else if (node.getData().compareTo(element) > 0) {  
insert(element, node.getLeft());
```

Algoritmos De Complemento À Inserção E/Ou Algoritmos Para Identificar Desbalanceamento Em Java

```
protected void rebalanceUp(BSTNode<T> node) {  
if (node == null || node.isEmpty()) return;  
rebalance(node);  
if (node.getParent() != null) {  
rebalanceUp(node.getParent());  
}  
protected int calculateBalance(BSTNode<T> node) {  
if (node == null || node.isEmpty()) return 0;  
return height(node.getRight()) - height(node.getLeft());  
}  
protected void rebalance(BSTNode<T> node) {  
int balanceOfNode = calculateBalance(node);  
if (balanceOfNode < -1) {  
if (calculateBalance(node.getLeft()) > 0) {  
leftRotation(node.getLeft());  
rightRotation(node);  
} else if (balanceOfNode > 1) {  
if (calculateBalance(node.getRight()) < 0) {  
rightRotation(node.getRight());  
leftRotation(node);
```

Rotação Para Direita E Para Esquerda Em Java

```
protected void leftRotation(BSTNode<T> no) {  
    BSTNode<T> noDireito = no.getRight();  
    no.setRight(noDireito.getLeft());  
    noDireito.getLeft().setParent(no);  
    noDireito.setLeft(no);  
    noDireito.setParent(no.getParent());  
    no.setParent(noDireito);
```

```

        if (no != this.getRoot()) {
            if (noDireito.getParent().getLeft() == no) {
                noDireito.getParent().setLeft(noDireito);
            } else {
                noDireito.getParent().setRight(noDireito);
            } else {
                this.root = (BSTNode<T>) noDireito;
            }
        }

protected void rightRotation(BSTNode<T> no) {
    BSTNode<T> noEsquerdo = no.getLeft();
    no.setLeft(noEsquerdo.getRight());
    noEsquerdo.getRight().setParent(no);
    noEsquerdo.setRight(no);
    noEsquerdo.setParent(no.getParent());
    no.setParent(noEsquerdo);
}

if (no != this.getRoot()) {
    if (noEsquerdo.getParent().getLeft() == no) {
        noEsquerdo.getParent().setLeft(noEsquerdo);
    } else {
        noEsquerdo.getParent().setRight(noEsquerdo);
    } else {
        this.root = (BSTNode<T>) noEsquerdo;
    }
}
    
```

### Remoção

O primeiro passo para remover uma chave  $K$  consiste em realizar uma busca binária a partir do nó raiz. Caso a busca encerre em uma subárvore vazia, então a chave não está na árvore e a remoção não pode ser realizada. Caso a busca encerre em um nó  $u$  ou o nó que contenha a chave então a remoção poderá ser realizada da seguinte forma:

Caso 1: O nó  $u$  é uma folha da árvore, apenas exclui-lo.

Caso 2: O nó  $u$  tem apenas uma subárvore, necessariamente composta de um nó folha, basta apontar o nó pai de  $u$  para a única subárvore e excluir o nó  $u$ .

Caso 3: O nó  $u$  tem duas subárvores: localizar o nó  $v$  predecessor ou sucessor de  $K$ , que sempre será um nó folha ou possuirá apenas uma subárvore; copiar a chave de  $v$  para o nó  $u$ ; excluir o nó  $v$  a partir da respectiva subárvore de  $u$ .

O último passo consiste em verificar a desregulagem de todos nós a partir do pai do nó excluído até o nó raiz da árvore. Aplicar rotação simples ou dupla em cada nó desregulado.

### Algoritmo Recursivo

```
remover_AVL(@p:^no_AVL, K:inteiro, @mudou_h:logico):logico;
```

```
var q: ^No_AVL; // ponteiro auxiliar para nó
inicio
se p = NULO então
    retornar Falso; // não achou a chave K a ser removida
se K < p^.chave então // remover recursivamente na subárvore esquerda
    se remover_AVL(p^.esq, K, mudou_h) então // ocorreu remoção na subárvore esquerda
        inicio
        se mudou_h então
            inicio // mudou altura da subárvore esquerda de p
            p^.fb := p^.fb + 1; // fator de balanço incrementa 1 unidade
            caso p^.fb
            2: p := rotacao_esquerda(p);
            se (p->fb=1) então mudou_h := Falso;
            1: mudou_h := Falso; // não mudou a altura da subárvore de raiz p
            // 0: mudou_h := Verdadeiro; // desnecessário pois mudou_h já é Verdadeiro
        fim
        retornar Verdadeiro;
    fim
senão
    se K > p^.chave então
        se remover_AVL(p^.dir, K, mudou_h) então // ocorreu remoção na s.a. direita
            inicio
            se mudou_h então
                inicio // mudou altura da subárvore direita de p
                p^.fb := p^.fb - 1; // fator de balanço decremente 1 unidade
                caso p^.fb
                -2: p := rotacao_direita(p);
                se (p->fb = -1) então mudou_h := Falso;
                -1: mudou_h := Falso; // não mudou a altura da subárvore de raiz p
                // 0: mudou_h := Verdadeiro; // desnecessário pois mudou_h já é Verdadeiro
            fim
            retornar Verdadeiro;
        fim
    fim
```

```
senão inicio
se K = p^.chave então // achou a chave K
inicio
se p^.esq=Nulo e p^.dir=Nulo então
inicio // nó folha
delete p;
p := Nulo; // Aterra a subárvore respectiva do nó pai
mudou_h := Verdadeiro;
fim
senão se p^.esq<>Nulo e p^.dir<>Nulo então
inicio // nó tem duas subárvores
q := Predecessor(p); // retorna nó com chave predecessora
p^.chave := q^.chave;
remover(p^.esq,p^.chave,mudou_h);
fim
senão inicio // tem apenas uma subárvore
se p^.esq<>Nulo então
inicio
p^.chave := p^.esq^.chave;
delete p^.esq;
p^.esq := Nulo;
fim
senão inicio
p^.chave := p^.dir^.chave;
delete(p^.dir);
p^.dir := Nulo;
fim;
mudou_h = Verdadeiro;
fim;
retornar Verdadeiro;
fim
fim
retornar Falso;
```

fim;

Predecessor(u:^No\_AVL):^No\_AVL // retorna nó contendo chave predecessora

inicio

u = u^.esq; // aponta para a raiz da subárvore esquerda

enquanto(u^.dir<>Nulo) faça // procura a maior chave da subárvore esquerda

u := u^.dir;

retornar u; // retorna o predecessor

fim;

Exemplo De Algoritmo De Remoção Em Java

```
public void remover(int valor) {
    removerAVL(this.raiz, valor);
}
private void removerAVL(No atual, int valor) {
    if (atual != null) {
        if (atual.getChave() > valor) {
            removerAVL(atual.getEsquerda(), valor);
        } else if (atual.getChave() < valor) {
            removerAVL(atual.getDireita(), valor);
        } else if (atual.getChave() == valor) {
            removerNoEncontrado(atual);
        }
    }
}
private void removerNoEncontrado(No noARemover) {
    No no;
    if (noARemover.getEsquerda() == null || noARemover.getDireita() == null) {
        if (noARemover.getPai() == null) {
            this.raiz = null;
            noARemover = null;
            return;
        }
        no = noARemover;
    } else {
        no = sucessor(noARemover);
        noARemover.setChave(no.getChave());
    }
    No no2;
    if (no.getEsquerda() != null) {
        no2 = no.getEsquerda();
    }
}
```

```
    } else {  
        no2 = no.getDireita();  
        if (no2 != null) {  
            no2.setPai(no.getPai());  
            if (no.getPai() == null) {  
                this.raiz = no2;  
            } else {  
                if (no == no.getPai().getEsquerda()) {  
                    no.getPai().setEsquerda(no2);  
                } else {  
                    no.getPai().setDireita(no2);  
                    verificarBalanceamento(no.getPai());  
                }  
            }  
        }  
        no = null;  
    }
```

#### Algoritmos Auxiliares Na Remoção

```
public No sucessor(No no) {  
    if (no.getDireita() != null) {  
        No noDireita = no.getDireita();  
        while (noDireita.getEsquerda() != null) {  
            noDireita = noDireita.getEsquerda();  
        }  
        return noDireita;  
    } else {  
        No noPai = no.getPai();  
        while (noPai != null && no == noPai.getDireita()) {  
            no = noPai;  
            noPai = no.getPai();  
        }  
        return noPai;  
    }  
}  
  
public void verificarBalanceamento(No atual) {  
    setBalanceamento(atual);  
    int balanceamento = atual.getBalanceamento();  
    if (balanceamento == -2) {  
        if (altura(atual.getEsquerda().getEsquerda()) >= altura(atual.getEsquerda().getDireita())) {  
            atual = rotacaoDireita(atual);  
        }  
    }  
}
```

```
        } else {  
            atual = duplaRotacaoEsquerdaDireita(atual);  
        }  
    } else if (balanceamento == 2) {  
        if (altura(atual.getDireita().getDireita()) >= altura(atual.getDireita().getEsquerda())) {  
            atual = rotacaoEsquerda(atual);  
        } else {  
            atual = duplaRotacaoDireitaEsquerda(atual);  
        }  
    }  
    if (atual.getPai() != null) {  
        verificarBalanceamento(atual.getPai());  
    } else {  
        this.raiz = atual;  
    }  
}
```

```
public No rotacaoEsquerda(No inicial) {  
    No direita = inicial.getDireita();  
    direita.setPai(inicial.getPai());  
    inicial.setDireita(direita.getEsquerda());  
    if (inicial.getDireita() != null) {  
        inicial.getDireita().setPai(inicial);  
    }  
    direita.setEsquerda(inicial);  
    inicial.setPai(direita);  
    if (direita.getPai() != null) {  
        if (direita.getPai().getDireita() == inicial) {  
            direita.getPai().setDireita(direita);  
        } else if (direita.getPai().getEsquerda() == inicial) {  
            direita.getPai().setEsquerda(direita);  
        }  
    }  
}
```



```
        }
    }
    setBalanceamento(inicial);
    setBalanceamento(direita);
    return direita;
}

public No rotacaoDireita(No inicial) {
    No esquerda = inicial.getEsquerda();
    esquerda.setPai(inicial.getPai());
    inicial.setEsquerda(esquerda.getDireita());
    if (inicial.getEsquerda() != null) {
        inicial.getEsquerda().setPai(inicial);
    }
    esquerda.setDireita(inicial);
    inicial.setPai(esquerda);
    if (esquerda.getPai() != null) {
        if (esquerda.getPai().getDireita() == inicial) {
            esquerda.getPai().setDireita(esquerda);
        } else if (esquerda.getPai().getEsquerda() == inicial) {
            esquerda.getPai().setEsquerda(esquerda);
        }
    }
    }
    setBalanceamento(inicial);
    setBalanceamento(esquerda);
    return esquerda;
}

public No duplaRotacaoEsquerdaDireita(No inicial) {
    inicial.setEsquerda(rotacaoEsquerda(inicial.getEsquerda()));
    return rotacaoDireita(inicial);
}

public No duplaRotacaoDireitaEsquerda(No inicial) {
    inicial.setDireita(rotacaoDireita(inicial.getDireita()));
```

```
        return rotacaoEsquerda(inicial);
    }
    private void setBalanceamento(No no) {
        no.setBalanceamento(altura(no.getDireita()) - altura(no.getEsquerda()));
    }

    private int altura(No atual) {
        if (atual == null) {
            return -1;
        }
        if (atual.getEsquerda() == null && atual.getDireita() == null) {
            return 0;
        }
        } else if (atual.getEsquerda() == null) {
            return 1 + altura(atual.getDireita());
        }
        } else if (atual.getDireita() == null) {
            return 1 + altura(atual.getEsquerda());
        }
        } else {
            return 1 + Math.max(altura(atual.getEsquerda()), altura(atual.getDireita()));
        }
    }
}
```

#### Como Identificar Mudança De Altura Na Remoção ?

Considerar que o nó p é raiz da subárvore Tp e houve remoção em uma de suas subárvores.

Caso a subárvore Tp tenha mudado de altura, incrementar fb (remoção na subárvore esquerda) ou decrementar fb (remoção na subárvore direita).

Caso 1: Ao remover um nó folha, a subárvore Tp passa de altura 1 para altura 0, então Tp mudou de altura.

Caso 2: fb=0 antes da remoção foi alterado para 1 (remoção à esquerda) ou -1 (remoção à direita), então a subárvore Tp não mudou de altura.

Caso 3: fb=1 ou -1 antes da remoção à direita e à esquerda, respectivamente, passando a ter valor 0, então a subárvore Tp mudou de altura.

Caso 4: fb passou a ter valor -2 ou 2 após a remoção, então houve necessidade de aplicação de rotação. Após a rotação dupla Tp muda de altura, exceto quando u=0 (antes da remoção- caso não relatado na literatura).

## Rotação

A operação básica em uma árvore AVL geralmente envolve os mesmos algoritmos de uma árvore de busca binária desbalanceada. A rotação na árvore AVL ocorre devido ao seu desbalanceamento, uma rotação simples ocorre quando um nó está desbalanceado e seu filho estiver no mesmo sentido da inclinação, formando uma linha reta. Uma rotação-dupla ocorre quando um nó estiver desbalanceado e seu filho estiver inclinado no sentido inverso ao pai, formando um "joelho".

Para garantirmos as propriedades da árvore AVL rotações devem ser feitas conforme necessário após operações de remoção ou inserção. Seja P o nó pai, FE o filho da esquerda de P e FD o filho da direita de P podemos definir 4 tipos diferentes de rotação:

Rotação simples à direita[editar | editar código-fonte]

Deve ser efetuada quando a diferença das alturas h dos filhos de P é igual a 2 e a diferença das alturas h dos filhos de FE é igual a 1. O nó FE deve tornar o novo pai e o nó P deve se tornar o filho da direita de FE. Segue pseudocódigo:

Seja Y o filho à esquerda de X

Torne o filho à direita de Y o filho à esquerda de X.

Torne X o filho à direita de Y



Caso 1.1 - Rotação Simples à Direita

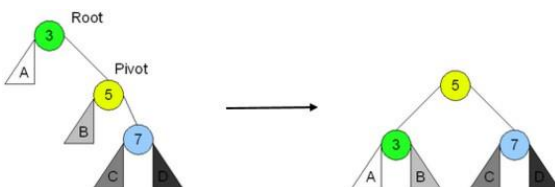
## Rotação à esquerda

Deve ser efetuada quando a diferença das alturas h dos filhos de P é igual a -2 e a diferença das alturas h dos filhos de FD é igual a -1. O nó FD deve tornar o novo pai e o nó P deve se tornar o filho da esquerda de FD. Segue pseudocódigo:

Seja Y o filho à direita de X

Torne o filho à esquerda de Y o filho à direita de X.

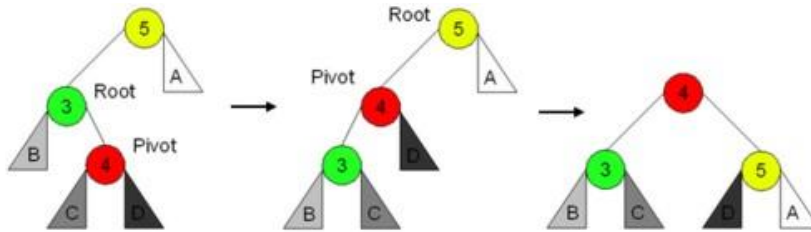
Torne X filho à esquerda de Y



## Rotação Simples A Esquerda

## Rotação Dupla À Direita

Deve ser efetuada quando a diferença das alturas h dos filhos de P é igual a 2 e a diferença das alturas h dos filhos de FE é igual a -1. Nesse caso devemos aplicar uma rotação à esquerda no nó FE e, em seguida, uma rotação à direita no nó P.

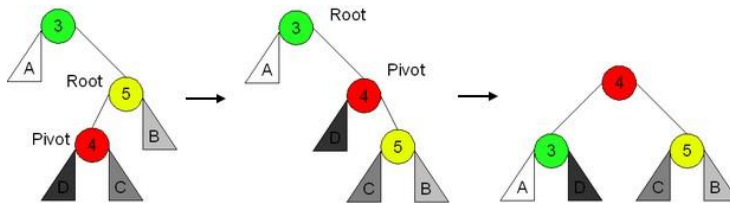


Caso 1.2 - Rotação dupla à direita

Deve ser observado que as 3 possíveis combinações de alturas da subárvores T2 e T3 constam da figura (h, h-1; h, h; e h-1, h), implicando em nos respectivos balanços do nó u (0/0/-1) e do nó p (1/0/0).

### Rotação Dupla À Esquerda

Deve ser efetuada quando a diferença das alturas h dos filhos de P é igual a -2 e a diferença das alturas h dos filhos de FD é igual a 1. Nesse caso devemos aplicar uma rotação à direita no nó FD e, em seguida, uma rotação à esquerda no nó P.



Rotação dupla à esquerda

Algoritmo de rotação à direita[editar | editar código-fonte]

```
rotaçaoDireita(p:NoAVL):@NoAVL;
```

```
inicio
```

```
var u,v:@NoAVL;
```

```
u := p^.esq;
```

```
se u^.fb > 0 então
```

```
inicio // rotação dupla, conforme figura Caso 1.2
```

```
v := u^.dir;
```

```
u^.dir := v^.esq;
```

```
p^.esq := v^.dir;
```

```
v^.esq := u;
```

```
v^.dir := p;
```

```
caso v->fb
```

```
-1: u^.fb := 0; p^.fb := 1;
```

```
0: u^.fb := 0; p^.fb := 0;
```

```
1: u^.fb := -1; p^.fb := 0;
```

```
v^.fb := 0;
```

```

retornar v;

fim;

// rotação simples, conforme figura Caso 1.1
p^.esq := u^.dir;
u^.dir := p;
se u^.fb < 0 então
  inicio
  u^.fb := 0;
  p^.fb := 0;
  fim;
senão inicio // ocorre apenas na remoção - não relatado na literatura
  u^.fb := 1;
  p^.fb := -1;
  fim;
retornar u;

fim;

```

Cabe ressaltar, apesar de não estar relatado na literatura, que na remoção poderá ocorrer a situação na qual  $u^.fb=0$ . Neste caso deverá ser aplicada rotação simples e o fator de balanço dos nós  $u$  e  $p$  serão respectivamente 1 e -1.

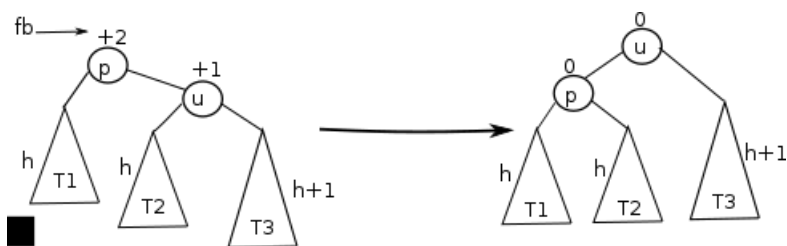
### Rotação À Esquerda

Caso o fator de balanço do nó  $p$  tenha valor +2, então haverá necessidade de aplicar rotação simples ou dupla à esquerda.

Para identificar se qual rotação aplicar, bastará analisar o fator de balanço do nó  $u$ , raiz da subárvore direita do nó  $p$ .

Rotação simples à esquerda[editar | editar código-fonte]

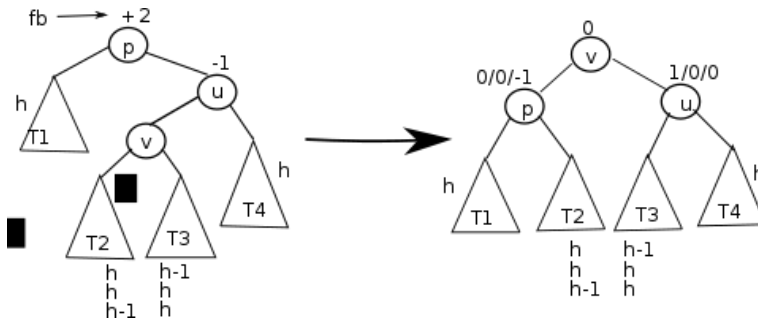
Caso o fator de balanço do nó  $u$  seja +1, então deverá ser aplicada uma rotação simples à esquerda. A figura a seguir mostra a configuração da árvore antes e depois da rotação.



### Caso 2.1 - Rotação Simples à Esquerda

Rotação dupla à esquerda[editar | editar código-fonte]

Caso o fator de balanço do nó u seja -1, então deverá ser aplicada uma rotação dupla à esquerda. A figura a seguir mostra a configuração da árvore antes e depois da rotação.



**Caso 2.2 - Rotação Dupla à Direita**

**Aplicações**

A árvore AVL é muito útil pois executa as operações de inserção, busca e remoção em tempo  $O(\log n)$  sendo inclusive mais rápida que a árvore rubro-negra para aplicações que fazem uma quantidade excessiva de buscas, porém esta estrutura é um pouco mais lenta para inserção e remoção. Isso se deve ao fato de as árvores AVL serem mais rigidamente balanceadas.

**Dicionários**

Árvore AVL pode ser usada para formar um dicionário de uma linguagem ou de programas, como os opcodes de um assembler ou um interpretador.

**Geometria Computacional**

Árvore AVL pode ser usada também na geometria computacional por ser uma estrutura muito rápida. Sem uma estrutura com complexidade  $O(\log n)$  alguns algoritmos da geometria computacional poderiam demorar dias para serem executados.

**Conjuntos**

Árvore AVL podem ser empregadas na implementação de conjuntos, principalmente aqueles cujas chave não são números inteiros.

A complexidade das principais operações de conjuntos usando árvore AVL:

Inserir -  $O(\log n)$ ;

Remover -  $O(\log n)$ ;

Pertence -  $O(\log n)$ ;

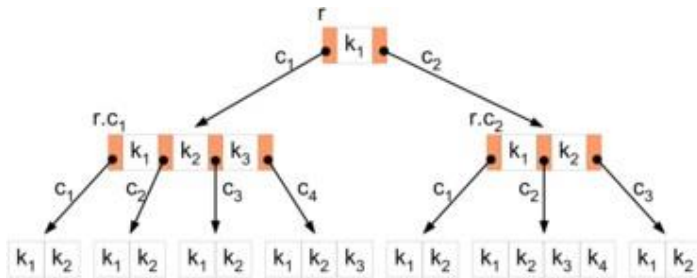
União -  $O(n \cdot \log n)$ ;

Interseção -  $O(n \cdot \log n)$ .

**Árvore B**

Árvore B		
Tipo	Árvore	
Ano	1971	
Inventado por	Rudolf Bayer, Edward Meyers McCreight	
Complexidade de Tempo em Notação big O		
	Algoritmo	Caso Médio
Espaço	$O(n)$	$O(n)$
Busca	$O(\log n)$	$O(\log n)$

Inserção	$O(\log n)$	$O(\log n)$
Remoção	$O(\log n)$	$O(\log n)$



**Exemplo de Árvore B**

Na ciência da computação uma árvore B é uma estrutura de dados projetada para funcionar especialmente em memória secundária como um disco magnético ou outros dispositivos de armazenamento secundário.

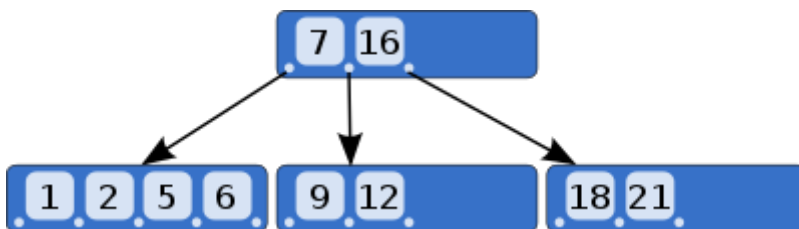
As árvores B são semelhantes as árvores preto e vermelho, mas são melhores para minimizar operações de E/S de disco. Muitos sistemas de bancos de dados usam árvores B ou variações da mesma para armazenar informações. Dentre suas propriedades ela permite a inserção, remoção e busca de chaves numa complexidade de tempo logarítmica e, por esse motivo, é muito empregada em aplicações que necessitam manipular grandes quantidades de informação tais como um banco de dados ou um sistema de arquivos.

Inventada por Rudolf Bayer e Edward Meyers McCreight em 1971 enquanto trabalhavam no Boeing Scientific Research Labs, a origem do nome (árvore B) não foi definida por estes.

Especula-se que o B venha da palavra balanceamento, do nome de um de seus inventores Bayer ou de Boeing, nome da empresa.

Árvores B são uma generalização das árvores binária de busca, pois cada nó de uma árvore binária armazena uma única chave de busca, enquanto as árvores B armazenam um número maior do que um de chaves de busca em cada nó, ou no termo mais usual para essa árvore, em cada página. Como a ideia principal das árvores B é trabalhar com dispositivos de memória secundária, quanto menos acessos a disco a estrutura de dados proporcionar, melhor será o desempenho do sistema na operação de busca sobre os dados manipulados.

**Visão Geral**



Árvore-B de ordem 2(Bayer & McCreight 1972) ou ordem 5 (Knuth 1998).

Os dispositivos de memória de um computador consistem na memória principal e secundária, sendo cada uma delas com suas características. A memória primária é mais conhecida como memória volátil de endereçamento direto (RAM), esta por sua vez apresenta baixo tempo de acesso, porém armazena um volume relativamente pequeno de informação e altos custos. Já a memória secundária, possui um endereçamento indireto, armazena um grande volume de informação e possui um acesso (seek) muito lento quando comparada com a memória primária. A árvore B é uma solução para cenários em que o volume de informação é alto (e este não pode ser armazenado diretamente em memó-

ria primária) e, portanto, apenas algumas páginas da árvore podem ser carregadas em memória primária.

As árvores B são organizadas por nós, tais como os das árvores binárias de busca, mas estes apresentam um conjunto de chaves maior do que um e são usualmente chamados de páginas. As chaves em cada página são, no momento da inserção, ordenadas de forma crescente e para cada chave há dois endereços para páginas filhas, sendo que, o endereço à esquerda é para uma página filha com um conjunto de chaves menor e o à direita para uma página filha com um conjunto de chaves maior. A figura acima demonstra essa organização de dados característica. Se um nó interno  $x$  contém  $n[x]$  chaves, então  $x$  tem  $n[x] + 1$  filhos. As chaves do nó  $x$  são usadas como pontos de divisão que separam o intervalo de chaves manipuladas por  $x$  em  $x[x]$  subintervalos, cada qual manipulado por um filho de  $x$ .

Vale lembrar que todo este endereçamento está gravado em arquivo (memória secundária) e que um acesso a uma posição do arquivo é uma operação muito lenta. Através da paginação é possível carregar em memória primária uma grande quantidade de registros contidos numa única página e assim decidir qual a próxima página que o algoritmo de busca irá carregar em memória primária caso esta chave buscada não esteja na primeira página carregada. Após carregada uma página em memória primária, a busca de chave pode ser realizada linearmente sobre o conjunto de chaves ou através de busca binária.

### Definição

#### Nó Ou Página

Um nó ou página, geralmente é representado por um conjunto de elementos apontando para seus filhos. Alguns autores consideram a ordem de uma árvore B como sendo a quantidade de registros que a página pode suportar. Outros consideram a ordem como a quantidade de campos apontadores. Todo nó da árvore tem um mínimo de registros definido pela metade da ordem, arredondando-se para baixo, caso a árvore seja de ordem ímpar, exceto a raiz da árvore, que pode ter um mínimo de um registro.

Por exemplo, os nós de uma árvore de ordem 5, podem ter, no mínimo  $\lfloor 5/2 \rfloor$  registros, ou seja, dois registros. A quantidade de filhos que um nó pode ter é sempre a quantidade de registros do nó mais 1 ( $V+1$ ). Por exemplo, se um nó tem 4 registros, este nó terá obrigatoriamente 5 apontamentos para os nós filhos.

Para definir uma árvore B devemos esclarecer os conceitos de ordem e página folha de acordo com cada autor Bayer e McCreight, com, dentre outros, definem a ordem como sendo o número mínimo de chaves que uma página pode conter, ou seja, com exceção da raiz todas devem conter esse número mínimo de chaves, mas essa definição pode causar ambiguidades quando se quer armazenar um número máximo ímpar de chaves. Por exemplo, se uma árvore B é de ordem 3, uma página estará cheia quando tiver 6 ou 7 chaves? Ou ainda, se quisermos armazenar no máximo 7 chaves em cada página qual será a ordem da árvore, uma vez que, o mínimo de chaves é  $k$  e o máximo  $2k$ ?

Knuth propôs que a ordem de uma árvore B fosse o número máximo de páginas filhas que toda página pode conter. Dessa forma, o número máximo de chaves por página ficou estabelecido como a ordem menos um.

O termo página folha também é inconsistente, pois é referenciado diferentemente por vários autores. Bayer e McCreight referem-se a estas como as páginas mais distantes da raiz, ou aquelas que contém chaves no nível mais baixo da árvore. Já Knuth define o termo como as páginas que estão abaixo do último nível da árvore, ou seja, páginas que não contém nenhuma chave.

De acordo com a definição de Knuth de ordem e página folha de Bayer e McCreight, uma árvore B de ordem  $d$  (número máximo de páginas filhas para uma página pai) deve satisfazer as seguintes propriedades:

Cada página contém no máximo  $d$  páginas filhas

Cada página, exceto a raiz e as folhas, tem pelo menos  $\lfloor d/2 \rfloor$  páginas filhas



A página raiz tem ao menos duas páginas filhas (ao menos que ela seja uma folha)

Toda página folha possui a mesma profundidade, na qual é equivalente à altura da árvore

Uma página não folha com k páginas filha contem k-1 chaves

Uma página folha contém pelo menos  $\lfloor d/2 \rfloor - 1$  chaves e no máximo d-1 chaves

#### Página Raiz

A página raiz das árvores B possuem o limite superior de d-1 chaves armazenadas, mas não apresentam um número mínimo de chaves, ou seja, elas podem ter um número inferior a  $\lfloor d/2 \rfloor - 1$  de chaves. Na figura acima, essa página é representada pelo nó que possui o registro 7 e 16.

#### Páginas Internas

As páginas internas são as páginas em que não são folhas e nem raiz, estas devem conter o número mínimo ( $\lfloor d/2 \rfloor - 1$ ) e máximo (d-1) de chaves.

#### Páginas Folha

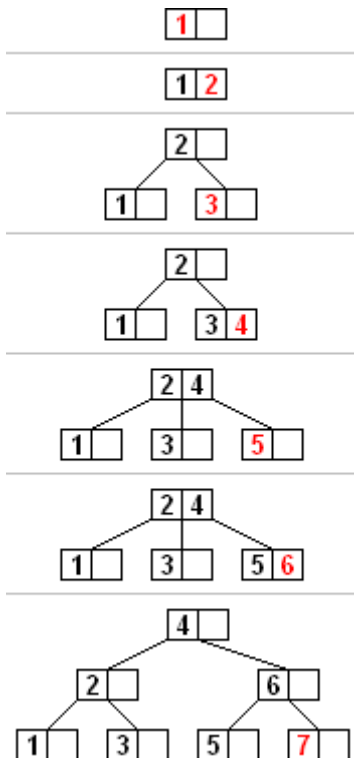
Estes são os nós que possuem a mesma restrição de máximo e mínimo de chaves das páginas internas, mas estes não possuem apontadores para páginas filhas. Na figura acima são todos os demais nós exceto a raiz.

#### Estrutura Da Página

Uma possível estrutura de dados para uma página de árvore B na linguagem C:

```
# define D 5 //árvore de ordem 5
typedef struct BTPage{
    //armazena numero de chaves na pagina
    short int totalChaves;
    //vetor de chaves
    int chaves[D-1];
    //Ponteiros das paginas filhas, -1 aponta para NULL
    struct BTPage filha[D];
}Page;
```

#### Operações Básicas



Exemplo de inserção em árvore B da sequência de 1 a 7. Os nós dessa árvore possuem no máximo 3 filhos

### Altura de uma Árvore B

O número de acessos ao disco exigidos para a maioria das operações em uma árvore B é proporcional a altura da árvore B.

### Como Criar uma Árvore Vazia

Para construir uma árvore B, primeiro criamos um nó raiz vazio, e depois inserimos novas chaves. Esses procedimentos alocam uma página de disco para ser usada como um novo nó no tempo  $O(1)$

As Operações Básicas Sobre Um Árvore B São A Busca, Inserção E Remoção De Chaves.

#### Busca

A busca de uma chave  $k$  em uma árvore B é muito parecido com uma busca em árvore binária, exceto pelo fato de que, em vez de tomar uma decisão de ramificação binária ou de "duas vias" em cada nó, tomamos uma decisão de ramificação de várias vias, de acordo com o número de filhos do nó. Em cada nó interno  $x$ , tomamos uma decisão de ramificação de  $(n[x] + 1)$  vias.

Esse método toma como entrada um ponteiro para o nó de raiz de uma subárvore e uma chave  $k$  a ser pesquisada.

#### Inserção

A operação de inserção, inicialmente com a árvore vazia, deve garantir que o nó raiz será criado. Criado o nó raiz, a inserção das próximas chaves seguem o mesmo procedimento: busca-se a posição correta da chave em um nó folha e insere a chave garantindo a ordenação destas. Após feito isso, considerando a abordagem de inserção de baixo para cima (Bottom-up) na árvore B, podem ocorrer duas situações:

Página folha está com um número menor de chaves do que o máximo permitido  $(d-1)$ : Nesse caso apenas inserimos a chave de maneira ordenada na página

Página folha completa ou com o número máximo de chaves permitido (d-1): Nesse caso ocorre o overflow da página em questão e é necessário a operação de split para manter o balanceamento da árvore.

Primeiramente escolhe-se um valor intermediário na sequência ordenada de chaves da página incluindo-se a nova chave que deveria ser inserida. Este valor é exatamente uma chave que ordenada com as chaves da página estará no meio da sequência.

Cria-se uma nova página e os valores maiores do que a chave intermediária são armazenados nessa nova página e os menores continuam na página anterior (operação de split).

Esta chave intermediária escolhida deverá ser inserido na página pai, na qual poderá também sofrer overflow ou deverá ser criada caso em que é criada uma nova página raiz. Esta série de overflows pode se propagar para toda a árvore B, o que garante o seu balanceamento na inserção de chaves.

Uma abordagem melhorada para a inserção é a de cima para baixo (Top-down) que utiliza uma estratégia parecida com a inserção de baixo para cima, a lógica para a inserção das próximas chaves (levando em consideração que a raiz já está criada) é a seguinte: busca-se a posição correta da chave em um nó, porém durante a busca da posição correta todo nó que estiver com o número máximo de chaves (d-1) é feita a operação de split, adicionando o elemento intermediário na sequência ordenada de chaves da página no pai e separando os elementos da página em outras duas novas páginas, onde uma vai conter os elementos menores que o elemento intermediário e a outra os elementos maiores que ele, a inserção será feita em um nó folha somente após todo o processo de split e insere a chave garantindo a ordenação destas. Esta abordagem melhorada previne de ter que ficar fazendo chamadas sucessivas ao pai do nó, o que pode ser caro se o pai estiver na memória secundária.

### Split

Trecho de uma árvore que tem ordem  $m = 3$ , sendo 10 o valor\_central no nó que sofre o split.

A função do split é dividir o nó em duas partes e "subir" o valor central do nó para um nó acima ou, caso o nó que sofreu o split seja a raiz, criar uma nova raiz com um novo nó. O que ocorre quando é feito um split:

Primeiramente calcula-se qual a mediana dos valores do nó, no caso o valor central do nó. Sendo tamanho = quantidade de elementos no nó, mediana = tamanho/2 e usamos a mediana para acessar o elemento que se encontra no centro do nó, no caso valor\_central = valores[mediana];

É testado se o nó que sofreu split tem pai, caso não, cria-se um novo nó apenas com o valor valor\_central e o seta como a nova raiz. São criados mais dois nós, cada um irá conter os valores do nó que estavam antes da mediana e depois da mediana. Um nó terá os valores menores que o valor\_central e ficará na primeira posição dos filhos da nova raiz, e o outro nó terá os valores maiores que o valor\_central e ficará na segunda posição dos filhos da nova raiz;

Caso o nó tenha pai, adicionamos o valor\_central ao nó pai. Caso o nó pai já esteja cheio, este também vai sofrer split após a inserção do valor nele. E da mesma forma que criamos dois nós para o caso do nó não ter pai, criaremos dois nós que conterão os valores menores e maiores que o valor\_central.

O nó com os menores valores ficará posicionado como filho do lado esquerdo do valor\_central e o nó com os maiores valores ficará posicionado como filho do lado direito do valor\_central. Por exemplo: Caso o valor\_central seja inserido na posição 0 do array de valores do nó pai, o nó filho com os menores valores ficará na posição 0 do array de filhos, e o nó com os maiores valores ficará na posição 1 do array de filhos.

### Remoção

A remoção é análoga a inserção, o algoritmo de remoção de uma árvore B deve garantir que as propriedades da árvore sejam mantidas, pois uma chave pode ser eliminada de qualquer página e não apenas de páginas folha. A remoção de um nó interno, exige que os filhos do nó sejam reorganizados. Como na inserção devemos nos resguardar contra a possibilidade da eliminação produzir uma árvore cuja estrutura viole as propriedades de árvores B. Da mesma maneira que tivemos de assegu-

rar que um nó não ficará pequeno demais durante a eliminação (a não ser pelo fato da raiz pode ter essa pequena quantidade de filhos).

Se o método para remover a chave  $k$  da subárvore com raiz em  $x$ . Este método tem que está estruturado para garantir que quando ele for chamado recursivamente em um nó  $x$ , o número de chaves em  $x$  seja pelo menos o grau mínimo  $t$ . Essa condição exige uma chave além do mínimo exigido pelas condições normais da árvore  $B$ , de forma que, quando necessário, uma chave seja movida para dentro do nó filho.

Descrição de como a eliminação funciona:

Se a chave  $k$  está no nó  $x$  e  $x$  é uma folha, elimine a chave  $k$  de  $x$ .

Se a chave  $k$  está no nó  $x$  e  $x$  é um nó interno:

Se o filho  $y$  que precede  $k$  no nó  $x$  tem pelo menos  $t$  chaves, então encontre o predecessor  $k'$  de  $k$  na subárvore com raiz em  $y$ . Elimine recursivamente  $k'$ , e substitua  $k$  por  $k'$  em  $x$ .

Simetricamente, se o filho  $z$  que segue  $k$  no nó  $x$  tem pelo menos  $t$  chaves, então encontre o sucessor  $k'$  de  $k$  na subárvore com raiz em  $z$ . Elimine recursivamente  $k'$  e substitua  $k$  por  $k'$  em  $x$ .

Caso contrário, se tanto  $y$  quanto  $z$  tem apenas  $t-1$  chaves, faça a intercalação de  $k$  e todos os seus itens  $z$  em  $y$ , de modo que  $x$  perca tanto  $k$  quanto o ponteiro para  $z$ , e  $y$  contenha agora  $2t-1$  chaves.

Se a chave  $k$  não estiver presente no nó interno  $x$ , determine a raiz  $c[x]$  da subárvore apropriada que deve conter  $k$ , se  $k$  estiver absolutamente na árvore. Se  $c[x]$  tiver somente  $t-1$  chaves:

Se  $c[x]$  tiver somente  $t-1$  chaves, mas tiver um irmão com  $t$  chaves, forneça a  $c[x]$  uma chave extra, movendo uma chave de  $x$  para baixo até  $c[x]$ , movendo uma chave do irmão esquerdo ou direito imediato de  $c[x]$  para dentro de  $x$ , e movendo o ponteiro do filho apropriado do irmão para  $c[x]$

Se  $c[x]$  e todos os irmão de  $c[x]$  têm  $t-1$  chaves, faça a intercalação de  $c[x]$  com um único irmão, o que envolve mover uma chave de  $x$  para baixo até o novo nó intercalado, a fim de se tornar a chave mediana para esse nó

Nessas operações podem ocorrer underflows nas páginas, ou seja, quando há um número abaixo do mínimo permitido ( $\lceil d/2 \rceil - 1$ ) de chaves em uma página.

Na remoção há vários casos a se analisar, as seguintes figuras apresentam alguns casos numa árvore de ordem 5:

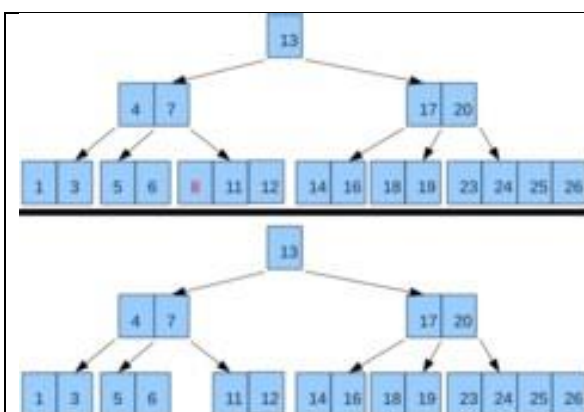


Figura 1: Remoção da chave 8 e posterior reorganização da estrutura

Caso da figura 1: Neste caso a remoção da chave 8 não causa o underflow na página folha em que ela está, portanto ela é simplesmente apagada e as outras chaves são reorganizadas mantendo sua ordenação.

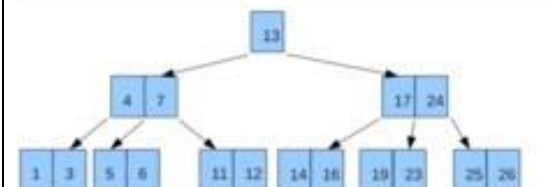
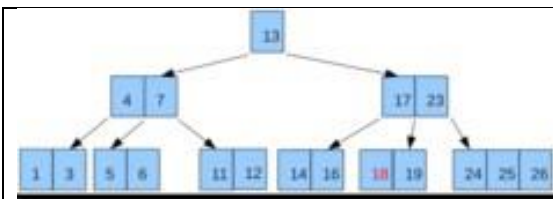


Figura 2: Remoção da chave 18 e posterior redistribuição das chaves

Caso da figura 2: O caso da figura 2 é apresentado a técnica de redistribuição de chaves. Na remoção da chave 18, a página que contém essa chave possui uma página irmã à direita com um número superior ao mínimo de chaves (página com chaves 24, 25 e 26) e, portanto, estas podem ser redistribuídas entre elas de maneira que no final nenhuma delas tenha um número inferior ao mínimo permitido.

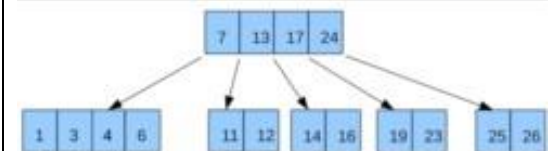
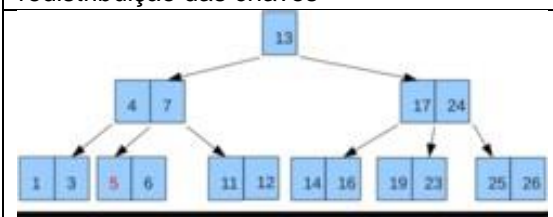


Figura 3: Remoção da chave 5 e posterior concatenação com página irmã à esquerda

Caso da figura 3: Nesta figura foi removido a chave 5, como não foi possível utilizar a técnica de redistribuição, pois as páginas irmãs possuem o número mínimo de chaves, então foi necessário concatenar o conteúdo da página que continha a chave 5 com sua página irmã à esquerda e a chave separadora pai. Ao final do processo a página pai fica com uma única chave (underflow) e é necessário diminuir a altura da árvore de maneira que o conteúdo da página pai e sua irmã, juntamente com a raiz, sejam concatenados para formar uma página única.

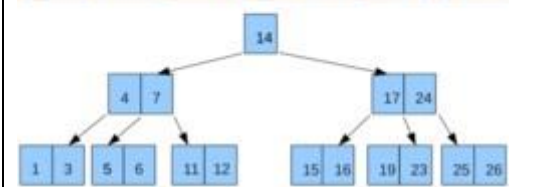
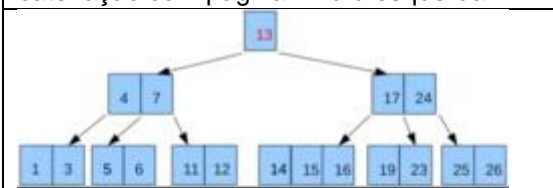


Figura 4: Remoção da chave 13 e promoção da menor chave da subárvore à direita de 13

Caso da figura 4: A remoção da chave 13 nesse caso foi realizado com a substituição do 13 pelo menor número da subárvore à direita de 13 que era o 14. Essa troca não causou o underflow da página em que estava o 14 e, portanto não gerou grandes alterações na árvore.

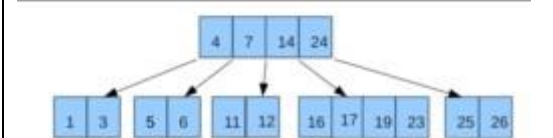
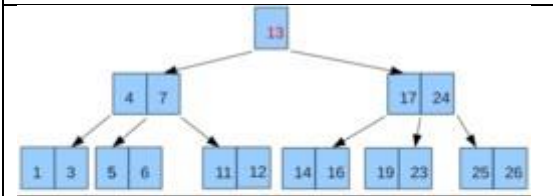


Figura 5: Chave 14 é promovida para a raiz o que causa underflow em sua página

Caso da figura 5: Caso semelhante ao anterior, mas esse ocorre o underflow da página que contém a menor chave da subárvore à direita de 13. Com isso, como não é possível a redistribuição, concatena-se o conteúdo dessa página com sua irmã à direita o que gera também underflow da página pai. O underflow da página pai também é resolvido com a concatenação com sua irmã e a raiz, resultando na diminuição da altura da árvore.

Algoritmos

Busca

Neste algoritmo recursivo os parâmetros recebidos inicialmente devem ser a chave buscada e um ponteiro para a página raiz da árvore B.

Busca(k, ponteiroRaiz)

```
{
  se(ponteiroRaiz == -1)
  {
    return (chave nao encontrada)
  }
  senao
  {
    carrega em memoria primaria pagina apontado por ponteiroRaiz
    procura k na pagina carregada
    se(k foi encontrada)
    {
      return (chave encontrada)
    }
    senao
    {
      ponteiro = ponteiro para a próxima página da possível ocorrência de k
      return (Busca (k, ponteiro))
    }
  }
}
```

Algoritmo De Busca Em Java

```
public BNodePosition<T> search(T element) {
    return searchAux(root, element);
}

private BNodePosition<T> searchAux(BNode<T> node, T element) {
    int i = 0;
    BNodePosition<T> nodePosition = new BNodePosition<T>();
    while (i <= node.elements.size() && element.compareTo(node.elements.get(i)) > 0) {
        i++;
    }
}
```

```
        if (i <= node.elements.size() && element.equals(node.elements.get(i))) {  
            nodePosition.position = i;  
            nodePosition.node = node;  
            return nodePosition;  
        }  
        if (node.isLeaf()) {  
            return new BNodePosition<T>();  
        }  
        return searchAux(node.children.get(i), element);  
    }  
}
```

### Inserção

O algoritmo de inserção em árvore B é um procedimento recursivo que inicialmente ponteiroRaiz aponta para a raiz da árvore em arquivo, key é a chave a ser inserida e chavePromovida representa a chave promovida após um split de uma página qualquer.

Insercao(ponteiroRaiz, key, chavePromovida)

```
{  
    se(ponteiroRaiz == -1)//se ponteiroRaiz nao aponta para nenhuma pagina  
    {  
        chavePromovida = key  
        return(flag que indica que houve promoção de chave)  
    }  
    senao  
    {  
        carregue a página P apontada por ponteiroRaiz em memoria primária  
        busque por key nessa página P  
        posicao = página no qual key poderia estar  
    }  
    se(key foi encontrada)  
    {  
        //chave ja esta na arvore, retorne uma flag de erro  
        return(flag de erro)  
    }  
    flagRetorno = Insercao(posicao, key, chavePromovida)//procedimento recursivo
```

se(flagRetorno indica que nao houve promoçao de chave ou que ocorreu um erro)

```
{
return(conteudo de flagRetorno)
}
senao se(há espaço na página P para chavePromovida)
{
insere chavePromovida na página P
escreve página P em arquivo
return(flag que indica que nao houve promocao de chave)
}
senao //nao ha espaço em P para key
{
realize operação de split em P
escreva em arquivo a nova página e a página P
return(flag que indica que houve promoçao de chave)
}
}
```

Inserção Recursiva

```
public void insertRec(BNode<T> node, T element) {
    if (node.isLeaf()) {
        node.addElement(element);
        if (node.elements.size() > node.getMaxKeys()) {
            node.split();
        }
    } else {
        int position = searchPositionInParent(node.getElements(), element);
        insertRec(node.getChildren().get(position), element);
    }
}

// insert abordagem top-down
public void insert(BNode<T> node, T element) {
    if(node.isFull()) {
        node = split(node); // Troque a referencia para o novo node.
```



```
        // split retorna a referencia do no que contem a mediana do no anterior.
    }
    if(node.isLeaf()) {
        node.addElement(element);
        this.size ++;
    } else {
        int i = 0;
        while (i < node.size() && node.getElementAt(i).compareTo(element) < 0) {
            i ++;
        }
        insert(node.getChildren().get(i), element);
    }
}
```

#### Remoção

1 Busque a chave k

2 Busque a menor chave M na página folha da sub-árvore à direita de k

3 Se a chave k não está numa folha

4 {

5 Substitua k por M

6 }

7 Apague a chave k ou M da página folha

8 Se a página folha não sofrer underflow

9 {

10 fim do algoritmo

11 }

12 Se a página folha sofrer underflow, verifique as páginas irmãs da página folha

13 {

14 Se uma das páginas tiver um número maior do que o mínimo redistribua as chaves

15 Senão concatene as páginas com uma de suas irmãs e a chave pai separadora

16 }

17 Se ocorrer concatenação de páginas aplique o trecho das linhas 8 até 17 para a página pai da folha

Algoritmo split em Java

```
protected void split() {
int mediana = (size()) / 2;
BNode<T> leftChildren = this.copyLeftChildren(mediana);
BNode<T> rightChildren = this.copyRightChildren(mediana);
if (parent == null) {
parent = new BNode<T>(maxChildren);
parent.children.addFirst(this);
}
BNode<T> parent = this.parent;
int index = parent.indexOfChild(this);
parent.removeChild(this);
parent.addChild(index, leftChildren);
parent.addChild(index + 1, rightChildren);
leftChildren.setParent(parent);
rightChildren.setParent(parent);
this.promote(mediana);
if (parent.size() >= maxChildren) {
parent.split();
}
}

protected void promote(int mid) {
T element = elements.get(mid);
this.parent.addElement(element);
}

Imprimir em Ordem em C
void emOrdem (tpaginaB raiz) {
if(raiz==NULL)
return;
for(int i=0;i<raiz.n,i++)
emOrdem(raiz->pont[i]);
printf("%i",raiz->chv[i]);
}
emOrdem(raiz->pont[raiz.n]);
```

}

Algoritmo Split Dentro Da Classe Node Em Java

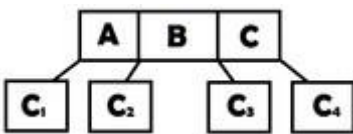
```
protected void split() {  
    T mediana = this.getElementAt(elements.size() / 2);  
    int posicao, esquerda, direita;  
    BNode<T> maior = new BNode<>(this.getMaxChildren());  
    BNode<T> menor = new BNode<>(this.getMaxChildren());  
    LinkedList<BNode<T>> criancas = new LinkedList<BNode<T>>();  
    this.armazenaElementos(mediana, maior, menor);  
    if (this.getParent() == null && this.isLeaf()) {  
        this.setElements(new LinkedList<T>());  
        this.addElement(mediana);  
        this.addChild(0, menor);  
        this.addChild(1, maior);  
    }  
    else if (this.getParent() == null && !isLeaf()) {  
        criancas = this.getChildren();  
        this.setElements(new LinkedList<T>());  
        this.addElement(mediana);  
        this.setChildren(new LinkedList<BNode<T>>());  
        this.addChild(0, menor);  
        this.addChild(1, maior);  
        this.reajustaFilhos(criancas, menor, 0, menor.size() + 1);  
        this.reajustaFilhos(criancas, maior, maior.size() + 1, criancas.size());  
    }  
    else if (this.isLeaf()) {  
        BNode<T> promote = new BNode<>(this.getMaxChildren());  
        promote.getElements().add(mediana);  
        promote.parent = this.getParent();  
        menor.parent = this.getParent();  
        maior.parent = this.getParent();  
        posicao = buscaPosicaoNoPai(promote.getParent().getElements(), mediana);  
        esquerda = posicao;
```

```

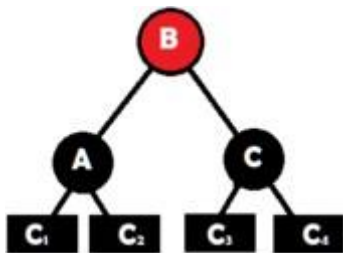
    direita = posicao + 1;
    this.getParent().getChildren().set(esquerda, menor);
    this.getParent().getChildren().add(direita, maior);
    promote.promote();
}
else {
    criancas = this.getChildren();
    BNode<T> paraPromote = new BNode<>(this.getMaxChildren());
    paraPromote.getElements().add(media);
    paraPromote.parent = this.getParent();
    menor.parent = this.getParent();
    maior.parent = this.getParent();
    posicao = buscaPosicaoNoPai(paraPromote.getElements(), media);
    esquerda = posicao;
    direita = posicao + 1;
    this.getParent().getChildren().add(esquerda, menor);
    this.getParent().getChildren().add(direita, maior);
}
}
}

```

Árvores 2-3-4



Representação genérica de árvore B com três chaves e, conseqüentemente, quatro filhos.



Árvore preta e vermelha resultante de uma transformação de uma árvore B.

Árvores 2-3-4 são um tipo de árvore B que possuem uma, duas ou três chaves. E, conseqüentemente, dois, três ou quatro filhos. São utilizadas na implementação de dicionários. Além disso, servem como base para o desenvolvimento do código de árvores preto e vermelho.

Existem três situações na mudança de árvore B para árvore preto-vermelho:

Caso o nó só possua uma chave, basta transformá-lo num nó de cor preta e ligá-lo aos seus filhos correspondentes.

Caso o nó possua duas chaves, a chave mais à esquerda será transformada num nó preto e a mais à direita, num nó vermelho. O nó preto terá como filho da esquerda o primeiro nó filho da antiga árvore e como filho da direita o novo nó vermelho. Este, por sua vez, terá como filhos os dois filhos restantes da lista de filhos da árvore original.

Caso o nó possua três chaves, a chave do meio será transformada num nó vermelho e terá como filhos as antigas chaves adjacentes que serão nós pretos com os antigos filhos da árvore B.

Aplicando essas situações, deve-se checar se as propriedades de árvores preto-vermelho são mantidas como o valor do nó da esquerda ser menor que o nó atual.

#### Variações

As árvores B não são as únicas estruturas de dados usadas em aplicações que demandam a manipulação de grande volume de dados, também existem variações desta que proporcionam determinadas características como as árvores B+ e B\*. Estas, por sua vez, se assemelham muito com as árvores B, mas possuem propriedades diferentes.

As árvores B+ possuem seus dados armazenados somente em seus nós folha e, seus nós internos e raiz, são apenas referências para as chaves que estão em nós folha. Assim é possível manter ponteiros em seus nós folha para um acesso sequencial ordenado das chaves contidas no arquivo.

Árvores B\* diferem das árvores B em relação ao particionamento de suas páginas. A estratégia dessa variação é realizar o particionamento de duas páginas irmãs somente quando estas estiverem completamente cheias e, claro, isso somente é possível através da redistribuição de chaves entre estas páginas filhas. Estando completamente cheias, as chaves são redistribuídas entre três páginas diferentes que são as duas irmãs anteriores e uma nova criada.

#### Comparação Com as Variações

Se compararmos as árvores B com suas variações podemos enumerar algumas características importantes para a escolha de implementação destas:

Árvores B+:

A principal característica proporcionada por esta variação é o fato de permitir o acesso sequencial das chaves por meio de seu sequence set de maneira mais eficiente do que o realizado em árvores B.

Além do mais, as páginas utilizadas em seu index set podem conter mais apontadores para páginas filha permitindo reduzir a altura da árvore.

Árvores B\*:

A principal vantagem decorrente dessa variação é o fato desta apresentar suas páginas com no mínimo 2/3 do número máximo de chaves, ou seja, esta variação apresenta no pior caso um índice de utilização do arquivo de 66%, enquanto em árvores B esse índice de pior caso cai para 50%.

#### Em Sistema de Arquivos

Além de sua utilização em bancos de dados, uma Árvore B (ou uma variante) também é usada em sistemas de arquivos para permitir acesso aleatório rápido a um bloco arbitrário em um arquivo particular. O problema básico consiste em transformar o bloco de arquivos  $i$  em um endereço de bloco de disco (ou talvez para um cilindro-cabeça-sector).



## Gerenciamento de Memória

A base do funcionamento da Memória Virtual é o Princípio da Localidade que estabelece que há uma tendência que os futuros endereços de memória de instruções e dados sejam próximos a endereços de memória recentemente acessados. Esse comportamento se deve as características peculiares aos programas, que frequentemente fazem uso de endereços em sequência (vetores), localizados em blocos de código bem definidos e frequentemente invocados (funções), ou de códigos repetitivos (laços de repetição).

A ideia básica da memória virtual é que o tamanho combinado do programa, dos seus dados e da pilha pode exceder a quantidade de memória física disponível para ele, ou seja, neste caso, a simples troca, vista anteriormente, não resolveria o problema. O Sistema Operacional, então, mantém partes do programa atualmente em uso, em forma de páginas ou segmentos, na memória principal e o restante em disco. Essas páginas/segmentos são "trocados" entre memória principal e secundária conforme o SO as solicita, conforme a demanda do programa.

A memória virtual também pode trabalhar em um sistema de multiprogramação, com pedaços de vários programas na memória simultaneamente. Enquanto um programa está esperando parte dele próprio ser trazido para a memória (ele fica esperando a E/S e não pode executar) a CPU pode ser dada a outro processo, assim como em qualquer sistema de multiprogramação.

Para a implementação desta técnica, alguns recursos mínimos são necessários: localização da página através do hardware MMU, carga de página, substituição de página e área de troca, partição ou arquivo especial de troca (swap ou página) destinada a armazenar páginas.

Muitos sistemas de Memória Virtual utilizam uma técnica denominada paginação, vista mais adiante.

### Troca (Swapping)

Em algumas situações não é possível manter todos os processos na memória e uma solução para essas situações é o mecanismo conhecido como swapping (troca). A gerência de memória reserva uma área do disco para esse mecanismo, que é utilizada para receber processos da memória. A execução desse processo é suspensa, com isso é dito que o mesmo sofreu uma swap-out. Mais tarde, esse mesmo processo será copiado do disco para a memória, mecanismo conhecido como swap-in. Esse mecanismo de trocas de processos no disco tem como objetivo permitir que o sistema operacional consiga executar mais processos do que caberia na memória.

Esse processo gera um grande custo de tempo de execução para os programas. Fazer a cópia do processo da memória para o disco e depois fazer o inverso é demorado.

### Paginação

O espaço de endereço virtual é dividido em unidades chamadas páginas. As unidades correspondentes na memória física são chamadas molduras de página (ou quadros). As páginas e as molduras (quadros) têm sempre exatamente o mesmo tamanho.

No espaço físico (memória) tem-se várias molduras de página. Por exemplo, podem existir 05 páginas situadas no espaço de endereço virtual que são mapeadas na memória física. No entanto, o espaço de endereço virtual é maior que o físico. As outras páginas não são mapeadas. No hardware real, um bit presente/ausente em cada entrada monitora se a página é mapeada ou não.

Quando um programa tenta utilizar uma página não mapeada em uma moldura, a MMU detecta o acontecimento (que a página não está mapeada) e gera uma interrupção, passando a CPU para o Sistema Operacional. Tal interrupção é chamada falha de página. O S.O., então, seleciona uma moldura de página pouco utilizada e grava o seu conteúdo de volta ao disco, substituindo-a pela página requisitada.

Quanto à forma como a paginação pode ser implementada, podemos considerar a paginação simples e a paginação por demanda. Na primeira, todas as páginas lógicas do processo são mapeadas e carregadas para a memória física, isso supondo-se que o espaço de endereçamento de memória para um processo tenha o tamanho máximo igual à capacidade da memória física alocada para processos. No caso da paginação por demanda, apenas as páginas lógicas efetivamente acessadas pelos processo

são carregadas. Nesse caso, uma página marcada como inválida na tabela de páginas de um processo pode tanto significar que a página está fora do espaço lógico de endereçamento do processo ou que simplesmente a página ainda não foi carregada. Para descobrir qual das situações é a verdadeira basta conferir o descritor de processo, que contém os limites de endereçamento lógico do processo em questão.

### Tabelas de Página

O propósito de tabelas de página é mapear páginas virtuais em molduras de página. No entanto, existem duas questões que devem ser consideradas:

A tabela de páginas pode ser extremamente grande, sendo que cada processo necessita de sua própria tabela de páginas;

O mapeamento deve ser rápido: o mapeamento do virtual para o físico deve ser feito em cada referência da memória, o que pode ocorrer diversas vezes em uma única instrução, não devendo tomar muito tempo para não se tornar um gargalo na execução da instrução.

Neste caso, há a necessidade de um mapeamento de páginas rápido e grande. Existem duas formas básicas de projetar tabelas de páginas:

ter uma única tabela de páginas, através de uma matriz de rápidos registradores de hardware, com uma entrada para cada página virtual, indexada pelo número da página. Esta é uma solução cara se a tabela de páginas é grande;

manter a tabela de páginas inteiramente na memória principal, sendo que o hardware precisa de apenas um registrador que aponta para o início da tabela de páginas.

Esta última solução possui algumas variações que têm desempenho melhor. Uma das propostas que busca evitar o problema de ter enormes tabelas de páginas na memória todo tempo é a de Tabelas de Páginas Multinível. Neste caso, existe uma tabela de primeiro nível e diversas tabelas de segundo nível. O importante aqui é que somente as tabelas mais utilizadas estão presentemente na memória. As demais se encontram em disco. Apesar de permitir um espaço de endereçamento muito grande, o espaço ocupado de memória principal é muito pequeno.

O sistema de tabela de páginas de dois níveis pode ser expandido para três, quatro ou mais níveis, sendo que níveis adicionais dão mais flexibilidade, mas a complexidade da implementação acima de três níveis dificilmente vale a pena. Em relação aos detalhes de uma única entrada de uma tabela de páginas, seu arranjo pode depender da máquina, mas os tipos de informação usualmente são os mesmos.

O campo mais importante é o número da moldura de página, pois o objetivo do mapeamento de páginas é localizar este valor. Ao lado dele, tem-se o bit de presente/ausente. Se este bit for 1, significa que a entrada é válida e pode ser utilizada. Se for 0, a página virtual a que esta entrada pertence não está atualmente na memória. Ao acessar uma entrada da tabela de páginas com este bit configurado como zero ocorrerá uma falha de página.

Os bits "proteção" informam que tipos de acesso são permitidos. Em sua forma simples, este campo contém apenas um bit, com 0 para leitura/gravação e 1 para leitura somente. Um arranjo mais sofisticado é manter 3 bits, cada um para habilitar leitura, gravação e execução da página. Os bits "modificada" e "referenciada" monitoram a utilização da página. Ambos os bits tem como objetivo auxiliar no momento da substituição de páginas. O último bit permite que o cache seja desativado para a página, recurso importante para páginas que são mapeadas em registradores de dispositivo em vez da memória.

Alguns bits auxiliares são, normalmente, adicionados na tabela de páginas para facilitar na substituição de páginas quando a memória física estiver cheia e for necessário retirar uma página lógica da memória física para alocar outra página lógica. Tem-se o bit de modificação (dirty bit) assim que a página for carregada tem valor zero se a página for alterada, na memória física, altera-se o valor para 1, portanto se a página for a página vítima e o bit de modificação for zero não será necessário fazer a cópia da página lógica em memória para a página lógica em disco pois as duas são iguais. Bit de referência: é zero assim que a página lógica é alocada na página física, se a página for acessada altera o valor para



um (a MMU altera o valor). Bit de trava: usado em páginas que não podem sair da memória física, o Sistema Operacional "tranca" uma página lógica na memória física ativando esse bit.

### TLBs – Translation Lookside Buffers – Memória Associativa

A TLB, também conhecida como memória associativa, é um dispositivo de hardware cujo propósito é mapear endereços virtuais em endereços físicos sem passar pela tabela de páginas. Usualmente, ela faz parte da MMU.

Ela constitui-se de um banco de registradores que armazenam um pequeno número de entradas, muito rápidas, contendo as tabelas de páginas mais utilizadas. Quando um endereço virtual é enviado a MMU, ela primeiramente verifica se o seu número de página virtual está presente na TLB. Se o resultado for positivo (hit), a moldura de página é tomada diretamente da TLB sem a necessidade de passar pela tabela de páginas na memória (mais lento). Caso contrário (miss), a pesquisa é feita normalmente na tabela de páginas presente na memória. Então, uma das entradas é removida da TLB e a entrada da tabela de páginas pesquisa é colocada em seu lugar.

A TLB melhora bastante o desempenho no acesso à tabela de páginas, visto que registradores são muito mais rápidos que a memória RAM. Suas desvantagens estão em seu custo (registradores são caros), seu tamanho limitado e o fato de existir uma única TLB na MMU, sendo esta compartilhada por todos os processos.

Para calcularmos o desempenho da TLB, podemos tomar  $h$  como taxa de acerto (taxa em que a página necessária estará na TLB). Então o erro seria  $1-h$ . Sendo assim, a tempo de acesso hit (tempo necessário para pegar a página da TLB) é dada pelo tempo de acesso à TLB mais o tempo de acesso à memória uma única vez. Enquanto o tempo de acesso miss (quando falta a página na TLB) é dada pelo tempo de acesso à TLB mais o tempo de dois acessos à memória. O tempo de acesso médio é dado pelo tempo de acesso hit multiplicado por  $h$  somado do tempo de acesso miss multiplicado por  $(1-h)$ . Resumindo em fórmulas, temos:

$$T_{\text{acessoHit}} = T_{\text{acessoTLB}} + T_{\text{acessoMemoria}}$$

$$T_{\text{acessoMiss}} = T_{\text{acessoTLB}} + T_{\text{acessoMemoria}} + T_{\text{acessoMemoria}}$$

$$T_{\text{acessoMedio}} = h \times T_{\text{acessoHit}} + (1-h) \times T_{\text{acessoMiss}}$$

A taxa de acerto ( $h$ ) depende do tamanho da TLB e do algoritmo que a controla, mantendo as páginas mais utilizadas.

### Tabelas de Páginas Invertidas

Outra abordagem para trabalhar com páginas é manter uma tabela onde cada entrada representa uma moldura de página ao invés de um endereço virtual. Desta forma, a entrada deve monitorar qual página virtual está associada àquela moldura de página. Embora as tabelas de páginas invertidas economizem quantidade significativas de espaço (pelo menos nas situações em que o espaço de endereço virtual é muito maior que a memória física), elas tem a desvantagem de que o mapeamento (tradução) do endereço virtual para o físico é mais complexo e potencialmente mais lento. Uma forma de facilitar a tradução do virtual para o físico é a utilização da TLB pesquisada por software. A pesquisa pode ser feita a partir de um encadeamento de páginas virtuais que possuam um mesmo endereço hash.

### Tamanho de Página

Um ponto do qual o projetista deve se preocupar é com o tamanho da página. Conforme visto anteriormente, se esse tamanho de página for grande, pode ocorrer de o processo utilizador não ocupar todo o espaço a ele destinado. Se a página tiver um tamanho demasiadamente pequeno, a tabela de páginas será muito grande.

É possível uma modelagem matemática. Considere que cada processo tenha tamanho de  $s$  bytes, e cada página tenha tamanho de  $p$  bytes. A tabela de páginas terá um espaço de  $e$  bytes por entrada. Assim, o número de páginas que um processo precisará é de  $s/p$ . O espaço que esse processo ocupa na tabela de páginas é  $s \cdot e/p$ . O tamanho perdido na última página devido a fragmentação interna será de  $p/2$ .

Assim, haverá um custo de  $s \cdot e/p + p/2$  da tabela de páginas. Para que o tamanho da página seja ideal, o custo será zero. Dessa forma, derivando a expressão anterior e igualando a zero obtemos que o tamanho ideal de página é de  $s \cdot e \cdot \text{sqrt}$ .

### Thrashing

Estado no qual o sistema operacional ao invés de executar instruções efetivas "gasta" tempo efetuando a troca de páginas entre memória física e memória lógica, em outras palavras desperdiça um tempo significativo trazendo ou removendo páginas da memória. Para o usuário a sensação é que o sistema está travado ou congelado e para o hardware há um significativo acesso ao disco ao invés de processamento

A memória como um recurso importante deve ser gerenciado com cuidado, sendo o gerenciador de memória a parte do sistema operacional que gerencia a sua hierarquia. Seu trabalho é controlar que partes da memória que estão em uso e que partes não estão, alocar memória para os processos quando eles necessitam e desalocarem quando eles terminarem, gerenciar a troca entre a memória principal e o disco quando a memória principal é muito pequena para armazenar todos os processos; Esses são alguns papéis do gerenciador de memória.

### Unidade de Gerenciamento de Memória (Memory Management Unit - MMU)

É um módulo de hardware que faz o mapeamento entre os endereços lógicos (end. da memória virtual) e os endereços físicos da memória (RAM), ou seja, é um dispositivo que transforma endereços virtuais em endereços físicos. A MMU normalmente traduz número de páginas virtuais para número de página físicas utilizando a cache chamada TLB-Translation LookKaside Buffer.

### Gerenciamento Básico de Memória

São divididos em duas classes:

Monoprogramação sem Troca ou Paginação: esquema mais simples de gerenciamento, executando apenas um programa por vez, compartilhando a memória entre o programa e o sistema operacional. Apenas um programa é carregado por vez.

Multiprogramação com partições fixas: aumentando a utilização da CPU, a multiprogramação permite que enquanto um processo é bloqueado esperando a E/S acabar, outro pode utilizar a CPU, melhorando a utilização da CPU e assim evitando desperdícios de ciclos de processamento. Na multiprogramação a memória é dividida em N partes, normalmente quando o sistema é iniciado, Os jobs são colocados em filas de entrada associadas à menor partição capaz de armazená-lo, por usar partições de tamanho fixo, todos o restante de espaço de memória não utilizado pelo Job será perdido, tal desperdício é chamado de Fragmentação Interna.

Outro tipo de fragmentação existente é a Fragmentação Externa que acontece na seguinte ocasião: imagine que exista duas partições livres, uma de 50 e outra de 80 kbytes, nesse instante é criado um processo de 90 Kbytes que não poderá ser carregado em memória pela forma como ela é gerenciada, sendo assim acontece uma fragmentação externa para que processo possa "interagir" com a memória. Na Multiprogramação com partições fixas os processos são colocados em filas únicas ou não, esperando alguma partição que possa armazená-los.

Multiprogramação com partições Variáveis: o tamanho dos processos na memória podem variar dinamicamente com o passar do tempo, o tamanho das partições é ajustado dinamicamente às necessidades exatas dos processos. A imagem abaixo ilustra o funcionamento deste algoritmo, considerando a ocorrência de swapping (trazer um processo do disco para a memória [swap in] executá-lo durante um intervalo de tempo e depois devolvê-lo ao disco [swap out]).

Diferentemente do esquema de partições fixa, na multiprogramação com partições variáveis o tamanho e a localização dos processos varia, a medida que o mesmo deixa e retorna à memória. A gerencia dos espaços vazios é mais complicada, bem como a alocação e liberação das partições. O sistema operacional mantém uma lista de espaços livres na memória física. Sempre que um novo processo é criado esta lista é percorrida e será usada uma lacuna maior ou igual ao tamanho do processo em questão. O espaço que ultrapassar o tamanho do processo pode dar origem a uma nova partição. As forma de percorrer esta lista é :

**Fisrt-fit:** Inicia a procura a partir da primeira página de memória e vai varrendo até encontrar a primeira lacuna suficientemente grande para armazenar o processo.

**Best-fit:** varre toda a memória e escolhe a página mais ajustada ao tamanho do processo.

**Worst-fit:** varre toda a memória e escolhe a página menos ajustada ao tamanho do processo.

**Next-fit:** segue a mesma idéia do firt-fit, mas somente a primeira busca é iniciada na parte da memória, as outras iniciam onde terminou a última. Usa-se uma lista circular para permitir que, eventualmente, toda a memória seja percorrida.

Como já mencionado a troca cria lacunas na memória é possível juntar todas elas em um grande espaço, movendo-os para a mesma direção. Técnica chamada de compactação de memória.

Como já mencionado, existem processos que tentem a necessitar mais tamanho de memória conforme são executados, para isso, nada melhor que alocar uma pequena memória extra para esses processos, essa memória extra somente deve constar no momento em que o processo esta na memória, quando ele retorna para o disco ele deve conter o tamanho real.

**Gerenciamento de memória com mapas de Bits:** a cada unidade de alocação da memória é atribuído um bit para dizer se a posição está livre ou ocupada. Assim, o conjunto de todos os bits é representado em uma tabela, denominada mapa de bits, que mapeia todas as posições de memória dizendo o estado de cada uma. Devemos ressaltar que o tamanho da unidade de alocação é muito importante e quanto menor as unidades, maior será o mapa de bits. Como o mapa de bits também é armazenado em memória seu tamanho ocupará espaço útil e, conseqüentemente, uma parte da memória será desperdiçada. Quando um processo de k bits necessitar ser armazenado em memória a MMU deverá procurar no mapa k bits consecutivos indicando que a posição está vazia (pode ser o bit 0 ou 1). Como varrer o mapa de bits é lento este método quase não é usado.

**Gerência de memória com lista ligada:** As representações dos espaços livres e ocupados são feitos através de uma lista ligada, onde P indica uma região ocupada por um processo e H um espaço livre de memória (imagem abaixo). A lista pode estar ordenada por endereços de memória, conforme ilustrado na imagem. Assim como no mapa de bits, qualquer alteração nas posições de memória deve gerar uma alteração no mapeamento promovido pela lista ligada. Se a lista estiver ordenada por endereço uma atualização mais rápida é permitida sempre que um processo terminar de executar suas instruções ou for retirado da memória. A utilização de uma lista duplamente encadeada facilita no processo de atualização da mesma.

Existem alguns algoritmos que podem ser utilizados para alocar as informações na memória:

**algoritmo da primeira alocação (first fit):** procura-se pelo primeiro espaço na lista o suficientemente grande para armazenar o processo. É um algoritmo rápido pois ele gasta o tempo mínimo em procura. Se o processo não ocupa todo o espaço o restante é disponibilizado como buraco na lista. A pesquisa por espaço sempre inicia na parte baixa de memória, independentemente dos locais escolhidos para alocar os dados.

**algoritmo da melhor alocação (best fit):** busca em toda a lista o espaço cujo o tamanho seja o mais próximo possível do tamanho do processo. Este algoritmo é mais lento que o anterior pois precisa pesquisar em toda a lista para descobrir qual a melhor opção.

**algoritmo da próxima alocação (next fit):** semelhante ao first-fit, só que a próxima alocação inicia com uma busca a partir da página onde terminou a alocação anterior e não da parte baixa da memória.

**algoritmo da pior alocação (worst fit):** procura pelo maior espaço capaz de armazenar o processo, de tal forma que o espaço restante seja grande o suficiente para armazenar outro processo.

### **Memória Virtual**

Técnica utilizada para executar um programa que não cabe na memória existente, consiste em manter partes do programa, dos dados e da pilha no disco, sendo que existe uma forma de decisão de quais processos devem permanecer no disco e quais na memória. Técnica realizada de forma automática

pelo computador, podemos alocar diversos processos na memória virtual, de forma que cada um pensa ter uma quantidade de memória que somadas ultrapassam a quantidade real de memória.

Então:

É um espaço variável e reservado no disco onde o Sistema Operacional continua armazenado os dados que não couberam na memória RAM. Ou seja, na memória RAM ficam os dados temporários usados enquanto o computador está ligado, se ela enche, os dados vão sendo gravados no HD.

Devido aos Sistemas Operacionais apresentarem características variadas, determinados procedimentos tornam-se por vezes semelhantes ou próprios de cada Sistema. O gerenciamento de memória é um procedimento fundamental na objetividade de todos os Sistemas Operacionais.

A maioria dos computadores trabalha com o conceito de hierarquia de memória, possuindo uma pequena quantidade de memória cachê, muito rápida, uma quantidade de memória principal (RAM) e uma quantidade muito grande de memória de armazenamento em disco (HD), considerada lenta.

A memória é um dos itens mais importantes dentro de um computador, por isso a importância de um bom gerenciamento.

Gerenciamento (ou gestão) de memória é um complexo campo da ciência da computação e são constantemente desenvolvidas várias técnicas para torná-la mais eficiente. Em sua forma mais simples, está relacionado em duas tarefas essenciais:

**Alocação:** Quando o programa requisita um bloco de memória, o gerenciador o disponibiliza para a alocação;

**Reciclagem:** Quando um bloco de memória foi alocado, mas os dados não foram requisitados por um determinado número de ciclos ou não há nenhum tipo de referência a este bloco pelo programa, esse bloco é liberado e pode ser reutilizado para outra requisição.

ou melhor

Os que não realizam paginação ou troca entre disco e memória

O que não realizam este processo

O ideal seria, uma memória que não precisasse deste artifício, uma memória rápida, de grande capacidade e a um custo pequena. o que não acontece.

O que se vê são vários tipos de memória trabalhando paralelamente com os recursos computacionais.

Assim, devido a suas similaridade e características distintas, as melhorias precisam de um meio para que possam ser gerenciadas da melhor forma possível.

É uma estrutura que possui uma hierarquia devido suas características e atividades desenvolvidas.

Cache (rápida e pequena, volátil, custo alto)

RAM (velocidade média, tamanho médio e volátil, custo médio)

Memórias Secundárias (velocidade lenta, tamanho grande e não volátil, custo baixo)

Também quando discutimos sobre gerenciamento de memória(GM), vimos que, apesar de seu crescimento exponencial com o tempo, cada vez temos mais memórias nos computadores modernos, temos do outro lado os programas cada vez mais exigentes com espaço, ou seja, cada vez temos programas maiores que a memória disponível.

A tendência é que, com as interfaces gráficas, os programas ocupem ainda mais as memórias. É um problema para os projetistas de Sistemas Operacionais que devem prever mecanismo de Gerenciamento de memória mais eficientes possíveis.

Outra questão está relacionada as modalidades de programação distinta, tais como:

Monoprogramação sem troca de processos ou paginação

Um esquema mais simples de gerenciamento de memória, onde a memória é compartilhada entre o SO e os programas.

Podemos neste caso, possui três configurações:

A 1ª configuração esteve presente nos Mainframes ou computadores de grande porte

A 2ª esta presente nos palmtops e outros computadores de mão

E a 3ª configuração esteve presente nos sistemas MS-DOS

### **Multiprogramação com Partições Fixas**

Atualmente, a monoprogramação não é mais utilizada, hoje, se um processo fica bloqueado esperando um recurso de E/S, outro processo poderá está utilizando a CPU naquele instante. Em sistemas Operacionais de rede, com vários usuários startando processos, isto bem evidente.

Esta habilidade presente somente neste sistemas operacionais rede, agora também faz parte dos sistemas dos usuários.

Como isso acontece, é realizado a divisão da memória em n partições que pode ser feita manualmente. Quando um job chega, há duas possibilidades: ele é colocado em uma fila de entrada da menor partição capaz de armazená-lo ou ele é colocado em uma fila de entrada única.

Como o tamanho das partições são fixas, se um job não ocupar todo o espaço da partição, este é pedido

### **Troca de Processos**

Em sistemas computacionais modernos, usualmente, não há memória principal suficiente para armazenar os processos atualmente ativos. Desta forma, os processos excedentes são mantidos em disco e trazidos para a RAM quando necessário. Há duas formas de resolver este problema: troca e memória virtual, (sendo a última explicada posteriormente).

A operação de troca consiste em manter na memória alguns processos por um determinado período e, a seguir, trocar estes processos pelos demais que estão esperando em disco. A partição utilizada para cada processo pode ser fixa ou definida dinamicamente (de acordo com o tamanho do processo). Durante a operação de troca, é possível surgirem lacunas de memória que podem ser reunidas em um espaço maior através da técnica de compactação de memória.

No entanto, a atribuição dinâmica de memória conduz a um problema adicional: o gerenciamento de memória em relação a quais parcelas estão em uso e/ou quais estão livres. Existem duas formas de resolver este problema.

### **Gerenciamento de Memória com Mapas de Bits**

Com mapas de bits, a memória é dividida em unidades de alocação. Cada bit do mapa representa uma unidade de alocação, sendo que se o bit for 0, a unidade está livre; caso contrário, a unidade está ocupada. Quanto menor for a unidade de alocação, maior será o mapa de bits e vice-versa. O maior problema com os mapas de bits é que procurar uma lacuna (sequência de 0s) suficientemente grande para um determinado processo pode ser uma operação muito lenta.

### **Gerenciamento de Memória com Listas Encadeadas**

Neste caso, é mantida uma lista encadeada com os segmentos de memória livres e encadeados. Uma possível configuração seria manter, em cada entrada, o endereço em que inicia, o seu comprimento e, evidentemente, o ponteiro para a próxima entrada.

### **Memória Virtual**

Quando os programas excedem o espaço de memória física disponível para eles, é possível dividir o programa em pedaços chamados overlays. No entanto, isto exige que o programador seja responsável por esta divisão. De forma a facilitar o desenvolvimento de programas, passando esta tarefa para o Sistema Operacional, foi criado o método conhecido como Memória Virtual.

A ideia básica da memória virtual é que o tamanho combinado do programa, dos seus dados e da pilha pode exceder a quantidade de memória física disponível para ele, ou seja, neste caso, a simples troca, vista anteriormente, não resolveria o problema. O Sistema Operacional, então, mantém partes do programa atualmente em uso na memória principal e o restante em disco.

### Paginação

Em qualquer computador, existe um conjunto de endereços que um programa pode produzir. Estes endereços gerados por programas são chamados endereços virtuais e formam o espaço de endereço virtual.

A MMU (Memory Management Unit) é responsável por mapear os endereços virtuais para endereços físicos de memória, sendo que consiste de um chip ou uma coleção de chips.

O espaço de endereço virtual é dividido em unidades chamadas páginas. As unidades correspondentes na memória física são chamadas molduras de página. As páginas e as molduras têm sempre exatamente o mesmo tamanho.

No espaço físico (memória) tem-se várias molduras de página. Por exemplo, podem existir 05 páginas situadas no espaço de endereço virtual que são mapeadas na memória física. No entanto, o espaço de endereço virtual é maior que o físico. As outras páginas não são mapeadas. No hardware real, um bit presente/ausente em cada entrada monitora se a página é mapeada ou não.

Quando um programa tenta utilizar uma página não mapeada em uma moldura, a MMU detecta o acontecimento (que a página não está mapeada) e gera uma interrupção, passando a CPU para o Sistema Operacional. Tal interrupção é chamada falha de página. O S.O., então, seleciona uma moldura de página pouco utilizada e grava o seu conteúdo de volta ao disco, substituindo-a pela página requisitada.

raduzir não é mais “apenas” transferir, com o mínimo de perdas e distorções, ideias de um idioma para outro. Em função das necessidades dos consumidores de hoje (e, por consequência, dos fornecedores), traduzir agora inclui os conceitos de “traduzir com preço competitivo” e “traduzir com homogeneidade de vocabulário” – para não falar de qualidade do atendimento, rapidez na entrega e tantas outras coisas que o tradutor está aprendendo a entregar com seu trabalho.

As questões do “preço competitivo” e da “homogeneidade de vocabulário”, por si, já justificam, na área técnica, o uso das memórias de tradução e dos gerenciadores de terminologia.

Acreditar, porém, que o simples fato de comprar o software e saber usá-lo já resolveria tais questões – isso seria ingenuidade. Depois que aprendemos a “pilotar” o software, aparecem novas necessidades.

E a principal é: pensar mais. Pensar, decidir, organizar: ora, se passamos a ter um banco de dados de unidades de tradução e outro de terminologia, então precisamos saber utilizar esse patrimônio. Se aproveitamos itens de outros trabalhos, precisamos ter controle absoluto sobre o modo como os aproveitamos. A gestão das memórias de tradução e dos glossários aparece como uma nova tarefa, porém, mais que um trabalho adicional, ela otimiza o uso do nosso patrimônio e evita erros – às vezes caros – durante o uso

O presente trabalho irá tratar sobre gerenciamento de memória nos sistemas operacionais, que é um recurso que deve ser manuseado com cuidado. Serão apresentadas a evolução da memória ao longo dos anos, sua estrutura, como funciona e os diversos tipos de procedimentos inclusos no gerenciamento básico de memória, como a definição de mono e multiprogramação. Serão apresentados, também, os conceitos de realocação e proteção, troca e uma rápida introdução à memória virtual.

A memória é um recurso muito importante que requer atenção. Ela vem crescendo cada vez mais ao longo dos anos, além de melhorar seu desempenho (velocidade), o que é ideal para os programadores,

visando também a característica de ser não-volátil, ou seja, o conteúdo não é perdido quando a memória perde contato com a energia. A maioria dos computadores possui uma hierarquia de memória, com uma pequena parte (cache) volátil rápida e cara, e outra parte, a maior, lenta, barata e não-volátil.

Em Sistemas Operacionais, a parte que cuida da memória é chamada de gerenciador de memória, que controla o que está e o que não está em uso, alocando a mesma para recursos que necessitem.

Os sistemas de gerenciamento podem ser divididos em duas classes: os que fazem troca e paginação, movendo processos entre memória principal e disco, e os que não fazem.

Monoprogramação sem Troca ou Paginação - esquema mais simples possível, executando somente um programa por vez, compartilhando memória entre esse programa e o sistema operacional. Quando organizado dessa maneira, somente um processo por vez pode ser executado. O usuário digita um comando, o sistema operacional copia do disco para a memória e executa, sobrepondo o primeiro quando um novo comando é adicionado e executado.

Multiprogramação com Partições Fixas - permite vários processos simultâneos, aumentando a utilização da CPU. A maneira mais fácil de implementá-la é dividir a memória em  $n$  partições. Quando um job chega, é 'alocado' para a menor partição capaz de armazená-lo (para que não haja perda), e assim sucessivamente, formando uma espécie de fila. Há também o modelo que utiliza partições fixas, o MFT (Multiprogramação com um número Fixo de Tarefas), onde cada job que chega aguarda uma partição livre adequada, onde são carregados e aguardam pela conclusão. Atualmente poucos sistemas operacionais suportam esse modelo.

A multiprogramação introduz problemas essenciais que devem ser resolvidos - realocação e proteção. A realocação procura assegurar que cada aplicação tenha seu próprio espaço de endereçamento, começando em zero. Faz uso do endereço absoluto, 100, por exemplo, somado com a localização da partição (partição 2, por exemplo,  $100 + 200K$ ). Para que haja uma realocação como essa durante o carregamento, o linkeditor inclui no programa binário uma lista ou mapa de bits, informando quais palavras do programa são endereços a serem realocados e quais são códigos de instruções, constantes ou outros itens.

No condizente à proteção, um programa malicioso poderia criar uma instrução que lê e grava qualquer palavra na memória e em seguida saltar para ela, já que trabalha com endereços absolutos. Em sistemas multiusuários, é indesejável que um processo leia ou grave memória pertencente a outros usuários. Para isso, a IBM dividiu a memória em blocos de 2KB e atribuiu um código de proteção de 4 bits para cada bloco. Apenas o SO poderia alterar os códigos e a chave de proteção.

Uma outra solução para ambos os problemas de realocação e proteção é equipar a máquina com registradores especiais de hardware, os chamados registrador base e registrador limite. Quando um processo é agendado, o base é carregado com o endereço do início da partição, e o de limite com o comprimento da partição.

Todo endereço gerado tem o registrador base automaticamente adicionado a si mesmo antes de ser enviado para a memória, assim uma instrução CALL 100 se transforma em CALL  $100K + 100$ , sem que a instrução seja modificada. O registrador limite certifica-se de que os endereços não tentarão endereçar memória fora da partição atual.

Um processo pode ser removido da memória temporariamente para um armazenamento auxiliar em seguida retornado para continuar sua execução. muitas vezes não há memória suficiente para armazenar todos os processos ativos. Os processos em excesso são mantidos no disco e trazidos de lá para execução dinamicamente.

Porém, um processo não pode invadir a memória de outro processo. No caso das partições variáveis a alocação da memória é dinâmica, ou seja, é alterada à medida que os processos entram e saem da memória. A vantagem é a melhor utilização da memória em relação às partições fixas, e a desvantagem é que as trocas de processos deixam muitos espaços vazios na memória fragmentando-a. Para este problema é possível a reorganização da memória, contudo esse processo é muito demorado.

A quantidade de memória que deve ser alocada a um processo quando ele for criado é um ponto muito importante a ser considerado. Se o processo não puder crescer então a alocação é simples, ou seja, o SO aloca somente o necessário. Uma solução alternativa é prevenir ou alocar uma memória extra para

o seu possível crescimento. Pode-se criar uma memória para variáveis dinâmicas que são alocadas e desalocadas (Heap), e outra (Stack) para variáveis locais comuns e endereços de retorno. Para que isso seja possível, o SO precisa gerenciar esse processo, usando duas práticas para gerenciamento de memória, dependendo do hardware disponível.

Uma delas é a troca, que é a mais simples, que consiste em trazer o processo inteiro, executá-lo temporariamente e então devolvê-lo ao disco. A outra estratégia é a memória virtual, que permite que os programas executem mesmo quando estão apenas parcialmente na memória principal. A compactação de memória é quando múltiplas lacunas na memória são juntas em um grande espaço, movendo o mais baixo possível. Um ponto importante é quanta memória deve ser alocada para um processo quando é criado ou recuperado. Se processos são criados com tamanhos fixos, a alocação é simples, alocando-se somente o necessário.

Em relação à troca, ela pode ser dividida em dois processos: gerenciamento por mapas de bits e gerenciamento com listas encadeadas, que serão vistas a seguir.

### **Gerenciamento de Memória com Mapas de Bits**

Com o mapa de bits há uma divisão da memória em unidades de alocação, tanto muito pequenas como muito grandes. Correspondente a cada unidade de alocação há um bit no mapa de bits, que é 0 se a unidade está livre ou 1 se estiver ocupada. Quanto menor a unidade de alocação, maior o mapa de bits. Um mapa de bits oferece uma maneira simples de monitorar palavras de memória em uma quantidade fixa de memória, já que o tamanho do mapa de bits depende do tamanho da memória e da unidade de alocação. O problema é que quando um processo de  $k$  unidades é enviado para a memória, o gerenciador deve fazer uma busca no mapa por uma lacuna de  $k$  unidades disponível. Essa pesquisa é uma operação lenta, sendo este um argumento contra os mapas de bits.

### **Gerenciamento de Memória com Listas Encadeadas**

Uma outra forma de monitorar a memória é manter uma lista encadeada dos segmentos de memória alocados e livres, onde um segmento é um processo ou lacuna entre dois processos. Quando os processos e as lacunas são mantidos em uma lista classificada por endereço, vários algoritmos podem ser utilizados para alocar memória para um processo criado recentemente ou trocado a memória. O algoritmo mais simples é chamado de algoritmo do primeiro ajuste. O algoritmo do primeiro ajuste é rápido, pois faz o mínimo de pesquisa de lacunas na lista. Uma variação secundária do algoritmo do primeiro ajuste é a do próximo ajuste, que funciona da mesma maneira que o primeiro, exceto pela monitoração da posição em que ele está sempre encontra uma lacuna adequada.

Outro algoritmo conhecido é o do melhor ajuste, o qual pesquisa na lista inteira e pega a menor lacuna adequada. Antes de dividir uma lacuna grande (que pode ser necessária posteriormente), o melhor ajuste tenta localizar uma lacuna que é próxima do tamanho real necessário. Esse algoritmo é mais lento do que o de primeiro ajuste, pois deve pesquisar na lista inteira uma vez que é chamado, além de ter um maior desperdício, já que tente a encher a memória de lacunas minúsculas e inúteis. Para que seja evitado o problema de dividir lacunas quase exatas entre um processo e uma lacuna minúscula, pode-se pensar no pior ajuste, que resume-se a sempre pegar a maior lacuna disponível, de modo que a lacuna resultante seja suficientemente grande para ser útil.

Todos esses quatro algoritmos podem ser acelerados ao se manter listas separadas para processos e lacunas, onde todos eles concentram-se em localizar as lacunas, não processos. O preço a ser pago por essa aceleração é a complexidade adicional e a queda de velocidade ao se deslocar memória. Ainda um outro algoritmo de alocação é o ajuste rápido, que mantém listas separadas para alguns dos tamanhos exigidos mais comuns. Com o ajuste rápido, localizar uma lacuna do tamanho exigido é extremamente rápido, mas tem a mesma desvantagem dos outros que classificam por tamanho de lacuna.

Antigamente, era feita a divisão da memória em pedaços chamados overlays, onde o overlay 0 começava primeiro e, quando pronto, chamava outro overlay. Alguns desses sistemas eram altamente complexos, permitindo múltiplos overlays na memória simultaneamente. Embora o trabalho de comutação overlay fosse feito pelo sistema, o trabalho de dividir o programa em pedaços era feito pelo programador, o que gerava muito trabalho ao dividir grandes programas em pequenos pedaços, até que uma solução foi encontrada. Essa solução veio a ser conhecida por memória virtual, cuja ideia básica é que o tamanho combinado do programa, dos dados e da pilha pode exceder a quantidade de memória física



disponível para ele. O sistema operacional mantém essas partes do programa atualmente em uso na memória principal e o restante no disco.

A memória virtual pode também trabalhar em sistemas de multiprogramação, com pedaços de vários programas diferentes na memória ao mesmo tempo. Enquanto um programa está esperando parte dele mesmo serem trazidas para a memória, ele espera a E/S e não pode executar, então a CPU pode ser dada a outro processo, assim como em qualquer sistema de multiprogramação.

No Linux a memória funciona da seguinte maneira, processos que estão em execução têm prioridade na memória, quando termina um processo e se tiver espaço na memória, ficam resíduos desse processo na memória para uma futura volta desse processo ser mais rápida. Caso essa memória RAM esteja lotada com processos que estão em execução, aí começa a utilização da memória SWAP (troca)". (LIMA, 2007)

A memória virtual do Linux é paginada, ou seja, é possível que sejam executados programas que tenham o tamanho maior que a própria memória física do computador. É como se o sistema operacional assumisse a responsabilidade de manter em sua memória e sustentasse os programas em execução, deixando o resto no HD. Só que a utilização da memória virtual torna o computador mais lento, mas faz com ele que aparente ter mais memória RAM do que tem de fato.

Ou seja, quanto mais memória é requisitada, o Linux passa a transferir arquivos não usados há algum tempo, da memória RAM à memória virtual, também chamada de swap, liberando, assim, memória física para os aplicativos.

Em suma, o que acontece é que o Linux utiliza a energia ociosa (que não está sendo aproveitada) para cache, a fim de agilizar os processos solicitados pelo usuário naquele determinado momento. Durante esse período de ociosidade da energia, o sistema toma para si essa memória, mas logo que o usuário necessite do recurso, o Linux libera imediatamente.

É possível afirmar, então, que o Linux gerencia a energia e a memória de forma muito inteligente, pois ao invés de deixá-la "sem fazer nada" utiliza para agilizar outros processos. Esse é o gerenciamento de memória do Linux!

### **Gerenciamento de Memória**

Gerenciador de Memória é a parte do SO que é responsável por cuidar de quais partes da memória estão em uso, quais estão livres, alocar memória a processos quando eles precisam, desalocar quando eles não necessitarem mais e gerenciar a troca dos processos entre a memória principal e o disco (quando a memória principal não é suficiente para manter todos os processos)

Maneiras de Gerenciar a Memória:

Gerenciamento sem Troca ou Paginação: troca e paginação são métodos utilizados de movimentação da memória para o disco e vice-versa durante a execução dos processos. Sem troca ou paginação é o caso mais simples.

Monoprogramação sem Troca ou Paginação: temos um único processo sendo executado por vez, de forma que o mesmo possa utilizar toda a memória disponível, com exceção da parte reservada ao SO (que permanece constante em local pré-determinado). O SO carrega um programa do disco para a memória executa-o e em seguida aguarda comandos do usuário para carregar um novo programa, que irá se sobrepor ao anterior.

Multiprogramação: a monoprogramação não é mais utilizada em sistemas grandes, pois:

Muitas aplicações são mais facilmente programáveis, quando as dividimos em dois ou mais processo;

Os grandes computadores em geral oferecem serviços interativos simultaneamente para diversos usuários (seria impossível trabalhar com um único processo por vez, pois representaria sobrecarga devido à constante necessidade de chavear de um processo para outro – constantemente lendo e escrevendo no disco);

É necessário que diversos processos estejam “simultaneamente” em execução devido as operações de E/S, que implica em grandes esperas nas quais por questão de eficiência a UCP deve ser entregue a outro processo.

Multiprogramação com Partições Fixas: consiste em dividir a memória existente em n partições fixas, podendo ser de tamanhos diferentes. Essas partições poderiam ser criadas ao inicializar o sistema pelo operador.

Uma maneira de se fazer isso seria: criar uma fila para cada partição existente e cada vez que um processo é iniciado, ele é colocado na fila de menor partição capaz de o executar. Os processos em cada partição são escolhidos de acordo com alguma forma de política, por exemplo, o primeiro a chegar é atendido antes. O problema é que pode ocorrer que uma partição grande esteja sem utilização, enquanto que diversos processos estão aguardando para utilizar uma partição menor. Para resolver isso podemos fazer o seguinte: estabelecer apenas uma fila para todas as partições e quando uma partição fica livre, um novo processo que caiba na partição livre é escolhido e colocado na mesma. A melhor forma de fazer a escolha seria percorrer a fila procurando o maior processo aguardando que caiba na partição livre, pois se a partição livre for entregue para o primeiro processo da fila, pode ocorrer que uma partição grande seja entregue a um processo pequeno.

Realocação e Proteção: há a necessidade de realocações, pois processos diferentes executam em posições diferentes de memória e com endereços diferentes. Uma possível solução é modificar as instruções conforme o programa é carregado na memória (quando o SO carrega o programa, adiciona a todas as instruções que se referenciam a endereços, o valor do ponto inicial de carga do programa). Esta solução exige que o linker coloque no início do código do programa, uma tabela que apresente as indicações das posições no programa que devem ser modificadas no carregamento. Mas isso não resolve a proteção, pois um programa malicioso ou errado pode ler ou alterar posições na memória de outros usuários, já que as referências são sempre as posições absolutas de memória.

Uma solução adotada para isso foi dividir a memória em unidades de 2 KB e associar um código de proteção de 4 bits a cada uma dessas regiões. Durante a execução de um processo, o PSW contém um código de 4 bits que é testado com todos os acessos à memória realizados pelo processo, e gera uma interrupção se tentar acessar uma região de código diferente.

Uma solução alternativa para o problema da realocação e da proteção é a utilização de registradores de base e limite. Sempre que um processo é carregado na memória, o SO ajusta o valor do registrador de base de acordo com a disponibilidade de memória. Toda vez que um acesso é realizado na memória pelo processo, o valor do registrado é automaticamente somado, assim não há necessidade de que o código do programa seja modificado durante o carregamento. O registrador de limite indica o espaço de memória que o processo pode executar, então todo acesso realizado pelo processo à memória é testado com o valor do registrador limite para a validação do seu acesso. O método dos registradores permite que um programa seja movido na memória, mesmo após já estar em execução, o que antes não era possível sem antes alterar os endereços novamente.

Troca (swapping): num sistema de batch, desde que se mantenha a UCP ocupada o máximo de tempo possível, não há necessidade de se complicar o método de gerenciamento de memória. Mas num sistema de time-sharing, onde muitas vezes existe menos memória do que o necessário para manter todos os processos de usuário, então é preciso que uma parte dos processos seja temporariamente mantida em disco. Para executar processos que estão no disco, eles devem ser enviados para a memória, o que significa retirar algum que lá estava. Este processo é denominado troca.

Multiprogramação com Partições Variáveis: partições que variam conforme as necessidades dos processos, em tamanho, em localização e em número de partições, melhorando a utilização da memória (mas complica a alocação e desalocação da memória). Compactação de memória que é a combinação de todos os buracos formados na memória em um único é raramente utilizada devido a grande utilização de UCP requerida.

Para determinarmos quanta memória deve ser alocada a um processo quando ele é iniciado, temos duas situações: se os processos necessitarem de uma quantidade pré-fixada e invariante de memória basta alocar a quantidade necessária a cada processo ativo. E o outro caso é quando os processos necessitam de mais memória durante o processamento (alocação dinâmica de memória). Neste caso pode existir um buraco de memória próximo ao processo bastando alocar a memória desse buraco ou

o processo pode estar cercado por outros processos, ou o buraco que existe não é suficiente. Para os dois últimos casos temo que tomar algumas das seguintes ações: mover o processo para um buraco de memória maior e se não houver tal espaço, alguns processos devem ser retirados da memória para deixar espaço para esse processo e se não houver espaço no disco para outros processos, o processo que pediu mais espaço na memória deve ser morto. Quando se espera que diversos processos cresçam durante a execução, o melhor seria reservar espaço extra para esses processos quando eles são criados para eliminar a sobrecarga de lidar com movimentação ou troca de processos.

Gerenciamento de Espaço: as duas principais formas de cuidar da utilização de memória são:

**Gerenciamento com Mapa de Bits:** A memória é subdividida em unidades de um certo tamanho. A cada unidade é associada um bit que se for 0 indica que essa parte da memória está livre e se for 1 indica que está ocupada. O tamanho deve ser cuidadosamente escolhido. A desvantagem é que quando um novo processo que ocupa  $k$  unidades de memória deve ser carregado na memória, o gerenciador deve percorrer o mapa de bits para encontrar  $k$  bits iguais a zero consecutivos, o que não é um processo simples.

**Gerenciamento com Listas Encadeadas:** mantemos uma lista encadeada de segmentos alocados e livres, sendo que cada segmento é um processo ou um buraco entre dois processos. A lista apresenta-se em ordem de endereços, e quando um processo termina ou é enviado para o disco, e a atualização da lista ocorre da seguinte maneira: cada processo, desde que não seja nem o primeiro nem o último da lista, apresenta-se cercado por dois segmentos, que podem ser buracos ou outros processos. Os buracos adjacentes devem ser combinados num único. Para escolher o ponto em que deve ser carregado um processo recém criado ou que veio do disco por uma troca, vamos utilizar alguns algoritmos assumindo que o gerenciador de memória sabe quanto espaço alocar no processo:

**First Fit (primeiro encaixe):** percorrer a fila até encontrar o primeiro espaço em que caiba o processo. É um algoritmo rápido.

**Next Fit (próximo encaixe):** o mesmo que o algoritmo anterior, só que ao invés de procurar sempre a partir do início da lista, procura a partir do último ponto em que encontrou. Desempenho próximo ao anterior.

**Best Fit (melhor encaixe):** consiste em verificar toda a lista e procurar o buraco que tiver espaço mais próximo das necessidades do processo. É mais lento, e desempenho pior que o First Fit

**Worst Fit (pior ajuste):** pega sempre o maior buraco disponível. Desempenho ruim.

Esses algoritmos podem ter sua velocidade aumentada pela manutenção de duas listas separadas, uma para processos e outra para buracos. Quando temos duas listas separadas, outro algoritmo é possível. É o Quick Fit (ajuste rápido), que consiste em manter listas separadas para alguns dos tamanhos mais comuns especificados (ex. uma fila para 2k, outra para 4k, outra para 8k etc). Neste caso, a busca de um buraco com o tamanho requerido, é extremamente rápido, entretanto, quando um processo termina, a liberação de seu espaço é complicada, devido à necessidade de reagrupar os buracos e modificá-los de fila.

**Alocação de espaço de troca (swap):** espaço de troca é o espaço ocupado no disco pelos processos que aí estão guardados, pois foram retirados da memória devido a uma troca. Os algoritmos para gerenciar o espaço alocado em disco para swap são os mesmos apresentados para o gerenciamento de memória. A diferença é que em alguns sistemas, cada processo tem no disco um espaço reservado para o mesmo e na memória ele é constantemente mudado de lugar. Além disso, como os discos são dispositivos de bloco, a quantidade de espaço reservado para os processos no disco deverá ser múltipla do tamanho do bloco.

**Memória Virtual:** A primeira solução adotada para programas grandes demais para a quantidade de memória foi a utilização de overlays. Nesta técnica o programa era subdividido em partes menores (overlays), que podiam ser rodadas separadamente e quando um overlay terminava a execução um outro poderia ser carregado na mesma posição de memória utilizada pelo anterior. O problema é a divisão do programa em overlays não é simples e deve ser realizada pelo programador.

A técnica de memória virtual para executar um programa que não cabe na memória existente consiste em manter partes do programa, dos dados e da pilha no disco, sendo que existe uma forma de decisão

de quais processos devem permanecer no disco e quais na memória. Esta técnica é realizada de forma automática pelo computador. Podemos alocar diversos processos na memória virtual, de forma que cada um pensa ter uma quantidade de memória que somadas ultrapassam a quantidade real de memória.

Técnicas Utilizadas em Sistemas com Memória Virtual:

**Paginação:** espaço virtual é o espaço de memória que pode ser referenciado por um programa qualquer em dado processador. Ex: um processador com endereçamento de 16 bits, possui um espaço virtual de 64 kbytes. Quando uma instrução como: LD A,(1000h) o 1000h corresponde a um endereço virtual, de um espaço de endereçamento virtual de 64k bytes. Em um computador sem memória virtual, o endereço virtual corresponde ao endereço efetivamente colocado no duto de endereçamento da memória. Quando o computador possui memória virtual, esse endereço virtual é enviado para uma unidade de gerenciamento de memória (MMU), que corresponde a um chip ou um conjunto de chips que translada esse endereço virtual em um endereço físico, de acordo com uma tabela.

O espaço de endereços virtuais é dividido em unidades chamadas páginas e o espaço de memória física é dividido em unidades chamadas quadros de página, de mesmo tamanho das páginas. A MMU tem uma tabela que indica para cada página, qual o quadro de página que corresponde à mesma. Se o processador tenta acessar o endereço 0, a MMU verifica que isto corresponde ao primeiro endereço da primeira página, verifica então que essa primeira página está alocada no terceiro quadro de página. Converte então esse endereço para 8192 (decimal) e envia o endereço convertido para a memória (nem a memória e nem o processador precisam ficar sabendo da existência de paginação).

Como nem todas as páginas do espaço virtual podem estar residentes na memória simultaneamente, ocorrer o caso de que um acesso seja realizado para um página que não está na memória. Para saber isso a MMU mantém na tabela de translação um bit para cada página que indica se a mesma está presente ou não na memória. Se um acesso for realizado a uma página ausente, é gerada uma falta de página, que chama uma rotina de tratamento de interrupção específica para o SO, que então se encarrega do carregamento da página faltante e o ajuste correspondente na tabela de translação.

A forma mais comum de implementação da MMU, é escolher alguns dos bits mais significativos do endereço virtual como indicadores do número de página e o restante dos bits como um deslocamento dentro dessa página. Ex: na figura acima, de 16 bits do endereço virtual, 12 serão utilizados para o deslocamento e 4 serão utilizados como um índice para qual das 16 páginas está sendo referenciada. A MMU pega os 4 bits do índice da página, acessa a posição correspondente da tabela de translação, verifica se a página está presente na memória, se não estiver, gera uma interrupção para carregamento e depois verifica o valor colocado nessa entrada da tabela de translação e os junto aos 12 bits de deslocamento dentro da página.

A paginação fornece uma forma de se conseguir grande espaços de endereçamento lineares em uma quantidade finita de memória física.

**Segmentação:** Na segmentação temos um espaço bidimensional no sentido de que é dividido em uma um certo número de segmentos, cada um com dado número de bytes. Dessa forma, um endereçamento é sempre expresso da forma (segmento, deslocamento). Os diferentes segmentos são associados a diversos programas ou mesmo arquivos, de forma que neste caso, os arquivos podem ser acessados como se fossem posições de memória. Na segmentação os programadores tentam colocar entidades diferentes em segmentos diferentes para facilitar o compartilhamento de objetos entre processos diferentes.

A maioria dos computadores trabalha com o conceito de hierarquia de memória, possuindo uma pequena quantidade de memória cache, muito rápida, uma quantidade de memória principal (RAM) e uma quantidade muito grande de memória de armazenamento em disco (HD), considerada lenta. O problema básico para o gerenciamento de memória é que os programas atuais são muito grandes para rodarem, completamente, na memória cache. O gerenciador de memória deve ser capaz de controlar parte da memória que está em uso (e quais não estão), alocar memória para processos quando eles necessitam e desalocar quando eles terminam e, principalmente, gerenciar a troca entre a memória principal e o disco, quando a memória principal é muito pequena para armazenar todos os processos.

Existem dois tipos de memória principal: a memória lógica e a memória física. A memória lógica é aquela manipulada pelos programas, ela é visível para os programas; sempre que um programa necessita alocar um espaço na memória esse espaço é alocado em memória lógica. A memória física é a memória implementada pelos circuitos integrados é nela que os espaços alocados em memória lógica vão realmente residir, portanto a memória física tem tamanho menor que a memória lógica, geralmente. Para isso é necessário realizar uma tradução de endereços lógicos para endereços físicos, pois assim um programa que aloca uma memória lógica possa ter de fato uma memória física alocada para si. Esse processo de tradução de endereços lógicos em endereços físicos é realizado por uma unidade de gerência de memória chamada MMU

Os processos não enxergam a memória física, hardware usado para endereçar os circuitos integrados de memória, e sim a memória lógica, que é a memória capaz de ser endereçada e acessada pelo conjunto de instruções do processador, sendo que cada processo possui a sua memória lógica que é independente da memória lógica dos outros processos. A memória física é implementada pelos circuitos integrados de memória, pela eletrônica do computador. Ela possui espaço de endereçamento físico que é o conjunto formado por todos os endereços dos circuitos integrados que formam a memória, normalmente esses endereços são em formato hexadecimal. A memória lógica possui um espaço de endereçamento lógico, maior que o espaço de endereçamento físico, é formado por todos os endereços lógicos gerado pelo processo, sendo gerado pela CPU e único por processo.

Como o processo "enxerga" endereço de memória lógico e o hardware "enxerga" endereço de memória físico é necessário ter a conversão de endereço de memória lógico para endereço de memória físico. Esse mapeamento de endereço lógico em endereço físico é feito pela MMU, unidade de gerência de memória. Basicamente a MMU que vai mapear os endereços lógicos gerados pelos processos nos correspondentes endereços físicos que serão enviados para a memória.

Existem duas formas bem simples de transformação do endereço lógico para o endereço físico:

A MMU verifica se o endereço lógico é maior que o registrador limite inferior e menor que o registrador limite superior, se sim encaminha o acesso com esse endereço válido, se não, gera uma interrupção de endereçamento inválido. Nesse caso o endereço lógico é igual ao endereço físico.

A MMU verifica se o endereço lógico é menor que o registrador limite superior, se sim adiciona o registrador base ao endereço lógico e encaminha o acesso com esse endereço resultante, se não gera interrupção de endereçamento inválido. Nesse caso o endereço lógico é diferente do endereço físico.

### **A MMU Consiste de um Chip ou uma Coleção de Chips.**

Para que um programa seja executado ele precisa ser transformado em processo(s), assim é necessário alocar o descritor de processos, alocar espaço na memória para o código (área conhecida como TEXT, onde se localiza o programa principal, as funções e as bibliotecas estáticas), os dados (Data, área onde as variáveis são alocadas - globais, locais estáticas, buffers internos) e a pilha (que possui o HEAP, área onde se localiza as variáveis dinâmicas, e o STACK, endereços de retorno de chamadas e parâmetros de funções).

A atribuição de endereço físico para as áreas de código e áreas de dados pode ser feita de três formas: em tempo de compilação, em tempo de carga e em tempo de execução. Em tempo de compilação o programador já faz a conversão de endereço lógico em endereço físico, pois ele tem conhecimento de qual área da memória ira utilizar.

Em tempo de carga o código precisa ser relocável de forma que todas as referências a memória sejam corrigidas para que o endereço de carga corresponda (carregador relocador), em outras palavras no momento da carga o programa executável é interpretado e os endereços corrigidos, dispensando a MMU. Em tempo de execução tem-se o código absoluto e é realizada uma relocação dinâmica usando a MMU, não sendo necessário corrigir os endereços no momento da carga do programa em memória.

Além de gerenciar quais espaços em memória estão em uso, é também necessário controlar os espaços de memória livres. Existem duas técnicas mais utilizadas para resolver este problema, estas serão vistas nas próximas seções.

### **Gerenciamento de Memória Livre com Mapas de Bits**

Com mapas de bits, a memória é dividida em unidades de alocação. Cada bit do mapa representa uma unidade de alocação, sendo que se o bit for 0, a unidade está livre; caso contrário, a unidade está ocupada. Quanto menor for a unidade de alocação, maior será o mapa de bits e vice-versa. O maior problema com os mapas de bits é que procurar uma lacuna (seqüência de 0s) suficientemente grande para um determinado processo pode ser uma operação muito lenta.

O SO mantém um 1 bit para indicar se cada bloco da memória (ou unidade de alocação) está ocupado (1) ou livre (0). A Memória é dividida em unidades de alocação. Considerações sobre o tamanho do bloco de memória:

Quanto menor a unidade de alocação, maior será o mapa de bits.

- Pequeno: necessidade de muitos bits  $\Rightarrow$  uso ineficiente da memória. Exemplo: se tamanho do bloco = 1 byte, 1/9 da memória serão utilizados para o mapa de bits.

- Grande: memória sub-utilizada, pois se o tamanho do processo não for múltiplo do tamanho da unidade de alocação, uma quantidade de memória considerável será desperdiçada no último bloco.

Vantagens do uso de Mapa de Bits:

Simplicidade: o tamanho do mapa depende apenas do tamanho da memória e das unidades de alocação.

Desvantagens:

Quando um processo necessita de k unidades de alocação, o gerenciador de memória deve encontrar uma seqüência de k bits 0, o que se constitui um processo lento.

### **Gerenciamento de Memória Livre com Listas Encadeadas**

Neste caso, é mantida uma lista encadeada com os segmentos de memória livres e encadeados. Uma possível configuração seria manter, em cada entrada, o endereço em que inicia, o seu comprimento e, evidentemente, o ponteiro para a próxima entrada.

A principal vantagem de utilizar uma lista encadeada classificada por endereço é que sua atualização é simples e direta. Vários algoritmos podem ser utilizados para encontrar uma lacuna de memória para alocação de um processo:

Primeiro ajuste: varre a lista desde o início e aloca no primeiro espaço (lacuna) suficientemente grande;

Próximo ajuste: varre a lista da posição atual e aloca no primeiro espaço suficientemente grande;

Melhor ajuste: varre a lista completamente e aloca no espaço que gerar a menor lacuna de memória;

Pior ajuste: varre a lista completamente e aloca no espaço que gerar a maior lacuna de memória disponível, de modo que a lacuna resultante possa ser suficientemente grande para ser útil;

Ajuste rápido: mantém diversas listas separadas para os tamanhos de processos mais comuns.

A base do funcionamento da Memória Virtual é o Princípio da Localidade que estabelece que há uma tendência que os futuros endereços de memória de instruções e dados sejam próximos a endereços de memória recentemente acessados. Esse comportamento se deve as características peculiares aos programas, que frequentemente fazem uso de endereços em seqüência (vetores), localizados em blocos de código bem definidos e frequentemente invocados (funções), ou de códigos repetitivos (laços de repetição).

A ideia básica da memória virtual é que o tamanho combinado do programa, dos seus dados e da pilha pode exceder a quantidade de memória física disponível para ele, ou seja, neste caso, a simples troca, vista anteriormente, não resolveria o problema. O Sistema Operacional, então, mantém partes do programa atualmente em uso, em forma de páginas ou segmentos, na memória principal e o restante em disco. Essas páginas/segmentos são "trocados" entre memória principal e secundária conforme o SO as solicita, conforme a demanda do programa.

A memória virtual também pode trabalhar em um sistema de multiprogramação, com pedaços de vários programas na memória simultaneamente. Enquanto um programa está esperando parte dele próprio ser trazido para a memória (ele fica esperando a E/S e não pode executar) a CPU pode ser dada a outro processo, assim como em qualquer sistema de multiprogramação.

Para a implementação desta técnica, alguns recursos mínimos são necessários: localização da página através do hardware MMU, carga de página, substituição de página e área de troca, partição ou arquivo especial de troca (swap ou página) destinada a armazenar páginas.

Muitos sistemas de Memória Virtual utilizam uma técnica denominada paginação, vista mais adiante.

### **Troca (Swapping)**

Em algumas situações não é possível manter todos os processos na memória e uma solução para essas situações é o mecanismo conhecido como swapping (troca). A gerência de memória reserva uma área do disco para esse mecanismo, que é utilizada para receber processos da memória. A execução desse processo é suspensa, com isso é dito que o mesmo sofreu uma swap-out. Mais tarde, esse mesmo processo será copiado do disco para a memória, mecanismo conhecido como swap-in. Esse mecanismo de trocas de processos no disco tem como objetivo permitir que o sistema operacional consiga executar mais processos do que caberia na memória.

Esse processo gera um grande custo de tempo de execução para os programas. Fazer a cópia do processo da memória para o disco e depois fazer o inverso é demorado.

### **Paginação**

O espaço de endereço virtual é dividido em unidades chamadas páginas. As unidades correspondentes na memória física são chamadas molduras de página (ou quadros). As páginas e as molduras (quadros) têm sempre exatamente o mesmo tamanho.

No espaço físico (memória) tem-se várias molduras de página. Por exemplo, podem existir 05 páginas situadas no espaço de endereço virtual que são mapeadas na memória física. No entanto, o espaço de endereço virtual é maior que o físico. As outras páginas não são mapeadas. No hardware real, um bit presente/ausente em cada entrada monitora se a página é mapeada ou não.

Quando um programa tenta utilizar uma página não mapeada em uma moldura, a MMU detecta o acontecimento (que a página não está mapeada) e gera uma interrupção, passando a CPU para o Sistema Operacional. Tal interrupção é chamada falha de página. O S.O., então, seleciona uma moldura de página pouco utilizada e grava o seu conteúdo de volta ao disco, substituindo-a pela página requisitada.

Quanto à forma como a paginação pode ser implementada, podemos considerar a paginação simples e a paginação por demanda. Na primeira, todas as páginas lógicas do processo são mapeadas e carregadas para a memória física, isso supondo-se que o espaço de endereçamento de memória para um processo tenha o tamanho máximo igual à capacidade da memória física alocada para processos.

No caso da paginação por demanda, apenas as páginas lógicas efetivamente acessadas pelos processos são carregadas. Nesse caso, uma página marcada como inválida na tabela de páginas de um processo pode tanto significar que a página está fora do espaço lógico de endereçamento do processo ou que simplesmente a página ainda não foi carregada. Para descobrir qual das situações é a verdadeira basta conferir o descritor de processo, que contém os limites de endereçamento lógico do processo em questão.

### **Tabelas de Página**

O propósito de tabelas de página é mapear páginas virtuais em molduras de página. No entanto, existem duas questões que devem ser consideradas:

A tabela de páginas pode ser extremamente grande, sendo que cada processo necessita de sua própria tabela de páginas;

O mapeamento deve ser rápido: o mapeamento do virtual para o físico deve ser feito em cada referência da memória, o que pode ocorrer diversas vezes em uma única instrução, não devendo tomar muito tempo para não se tornar um gargalo na execução da instrução.

Neste caso, há a necessidade de um mapeamento de páginas rápido e grande. Existem duas formas básicas de projetar tabelas de páginas:

ter uma única tabela de páginas, através de uma matriz de rápidos registradores de hardware, com uma entrada para cada página virtual, indexada pelo número da página. Esta é uma solução cara se a tabela de páginas é grande;

manter a tabela de páginas inteiramente na memória principal, sendo que o hardware precisa de apenas um registrador que aponta para o início da tabela de páginas.

Esta última solução possui algumas variações que têm desempenho melhor. Uma das propostas que busca evitar o problema de ter enormes tabelas de páginas na memória todo tempo é a de Tabelas de Páginas Multinível. Neste caso, existe uma tabela de primeiro nível e diversas tabelas de segundo nível. O importante aqui é que somente as tabelas mais utilizadas estão presentemente na memória. As demais se encontram em disco. Apesar de permitir um espaço de endereçamento muito grande, o espaço ocupado de memória principal é muito pequeno.

O sistema de tabela de páginas de dois níveis pode ser expandido para três, quatro ou mais níveis, sendo que níveis adicionais dão mais flexibilidade, mas a complexidade da implementação acima de três níveis dificilmente vale a pena. Em relação aos detalhes de uma única entrada de uma tabela de páginas, seu arranjo pode depender da máquina, mas os tipos de informação usualmente são os mesmos.

O campo mais importante é o número da moldura de página, pois o objetivo do mapeamento de páginas é localizar este valor. Ao lado dele, tem-se o bit de presente/ausente. Se este bit for 1, significa que a entrada é válida e pode ser utilizada. Se for 0, a página virtual a que esta entrada pertence não está atualmente na memória. Ao acessar uma entrada da tabela de páginas com este bit configurado como zero ocorrerá uma falha de página.

Os bits "proteção" informam que tipos de acesso são permitidos. Em sua forma simples, este campo contém apenas um bit, com 0 para leitura/gravação e 1 para leitura somente. Um arranjo mais sofisticado é manter 3 bits, cada um para habilitar leitura, gravação e execução da página. Os bits "modificada" e "referenciada" monitoram a utilização da página. Ambos os bits tem como objetivo auxiliar no momento da substituição de páginas. O último bit permite que o cache seja desativado para a página, recurso importante para páginas que são mapeadas em registradores de dispositivo em vez da memória.

Alguns bits auxiliares são, normalmente, adicionados na tabela de páginas para facilitar na substituição de páginas quando a memória física estiver cheia e for necessário retirar uma página lógica da memória física para alocar outra página lógica. Tem-se o bit de modificação (dirty bit) assim que a página for carregada tem valor zero se a página for alterada, na memória física, altera-se o valor para 1, portanto se a página for a página vítima e o bit de modificação for zero não será necessário fazer a cópia da página lógica em memória para a página lógica em disco pois as duas são iguais. Bit de referência: é zero assim que a página lógica é alocada na página física, se a página for acessada altera o valor para um (a MMU altera o valor). Bit de trava: usado em páginas que não podem sair da memória física, o Sistema Operacional "tranca" uma página lógica na memória física ativando esse bit.

### **TLBs – Translation Lookside Buffers – Memória Associativa**

A TLB, também conhecida como memória associativa, é um dispositivo de hardware cujo propósito é mapear endereços virtuais em endereços físicos sem passar pela tabela de páginas. Usualmente, ela faz parte da MMU.

Ela constitui-se de um banco de registradores que armazenam um pequeno número de entradas, muito rápidas, contendo as tabelas de páginas mais utilizadas. Quando um endereço virtual é enviado a MMU, ela primeiramente verifica se o seu número de página virtual está presente na TLB. Se o resultado for positivo (hit), a moldura de página é tomada diretamente da TLB sem a necessidade de passar pela tabela de páginas na memória (mais lento). Caso contrário (miss), a pesquisa é feita normalmente na



tabela de páginas presente na memória. Então, uma das entradas é removida da TLB e a entrada da tabela de páginas pesquisada é colocada em seu lugar.

A TLB melhora bastante o desempenho no acesso à tabela de páginas, visto que registradores são muito mais rápidos que a memória RAM. Suas desvantagens estão em seu custo (registradores são caros), seu tamanho limitado e o fato de existir uma única TLB na MMU, sendo esta compartilhada por todos os processos.

Para calcularmos o desempenho da TLB, podemos tomar  $h$  como taxa de acerto (taxa em que a página necessária estará na TLB). Então o erro seria  $1-h$ . Sendo assim, a tempo de acesso hit (tempo necessário para pegar a página da TLB) é dada pelo tempo de acesso à TLB mais o tempo de acesso à memória uma única vez. Enquanto o tempo de acesso miss (quando falta a página na TLB) é dada pelo tempo de acesso à TLB mais o tempo de dois acessos à memória. O tempo de acesso médio é dado pelo tempo de acesso hit multiplicado por  $h$  somado do tempo de acesso miss multiplicado por  $(1-h)$ . Resumindo em fórmulas, temos:

$$T_{\text{acessoHit}} = T_{\text{acessoTLB}} + T_{\text{acessoMemoria}}$$

$$T_{\text{acessoMiss}} = T_{\text{acessoTLB}} + T_{\text{acessoMemoria}} + T_{\text{acessoMemoria}}$$

$$T_{\text{acessoMedio}} = h \times T_{\text{acessoHit}} + (1-h) \times T_{\text{acessoMiss}}$$

A taxa de acerto ( $h$ ) depende do tamanho da TLB e do algoritmo que a controla, mantendo as páginas mais utilizadas.

### Tabelas de Páginas Invertidas

Outra abordagem para trabalhar com páginas é manter uma tabela onde cada entrada representa uma moldura de página ao invés de um endereço virtual. Desta forma, a entrada deve monitorar qual página virtual está associada àquela moldura de página. Embora as tabelas de páginas invertidas economizem quantidade significativas de espaço (pelo menos nas situações em que o espaço de endereço virtual é muito maior que a memória física), elas tem a desvantagem de que o mapeamento (tradução) do endereço virtual para o físico é mais complexo e potencialmente mais lento. Uma forma de facilitar a tradução do virtual para o físico é a utilização da TLB pesquisada por software. A pesquisa pode ser feita a partir de um encadeamento de páginas virtuais que possuam um mesmo endereço hash.

### Tamanho de Página

Um ponto do qual o projetista deve se preocupar é com o tamanho da página. Conforme visto anteriormente, se esse tamanho de página for grande, pode ocorrer de o processo utilizador não ocupar todo o espaço a ele destinado. Se a página tiver um tamanho demasiadamente pequeno, a tabela de páginas será muito grande.

É possível uma modelagem matemática. Considere que cada processo tenha tamanho de  $s$  bytes, e cada página tenha tamanho de  $p$  bytes. A tabela de páginas terá um espaço de  $e$  bytes por entrada. Assim, o numero de páginas que um processo precisará é de  $s/p$ . O espaço que esse processo ocupa na tabela de páginas é  $s \cdot e/p$ . O tamanho perdido na última página devido a fragmentação interna será de  $p/2$ .

Assim, haverá um custo de  $s \cdot e/p + p/2$  da tabela de páginas. Para que o tamanho da página seja ideal, o custo será zero. Dessa forma, derivando a expressão anterior e igualando a zero obtemos que o tamanho ideal de página é de  $s \cdot e \cdot \sqrt{2}$ .

### Thrashing

Estado no qual o sistema operacional ao invés de executar instruções efetivas "gasta" tempo efetuando a troca de páginas entre memória física e memória lógica, em outras palavras desperdiça um tempo significativo trazendo ou removendo páginas da memória. Para o usuário a sensação é que o sistema está travado ou congelado e para o hardware há um significativo acesso ao disco ao invés de processamento

## **Informática Sistemas Operacionais Windows E Linux**

Os Sistemas Operacionais (SO) têm evoluído com o tempo, tornando-se mais fáceis, bonitos e agradáveis ao usuário. Mas antigamente a história era outra, sua estrutura e complexidade não permitiam que qualquer usuário comum operasse em SO.

Para adquirir noções sobre esse tema, especialmente com relação a Windows e Linux é necessário entender o que é um software. Eles foram criados para que um computador funcionasse corretamente, pois o hardware não executa tarefas sozinho, mas por meio de um sistema que gerencia as atividades.

Softwares são todos os elementos que fazem parte da programação e que funcionam dentro da estrutura física do computador (hardware). Assim, eles são classificados em dois tipos:

**Softwares Básicos:** programas básicos e indispensáveis para o funcionamento do computador. Ex.: Sistema Operacional, utilitários, tradutores, linguagens de programação e ambiente operacional.

**Softwares Aplicativos:** são todos os programas que se preocupam em atender as necessidades de um usuário comum. Podem ser programas de uso geral, como planilhas, editores de texto, criação de gráficos, gerenciamento de dados, etc. E, também, programas de uso específico, construídos apenas para um determinado objetivo, como realização do imposto de renda, folha de pagamento, crediário, etc.

O Que É Sistema Operacional?

O Sistema Operacional é um dispositivo lógico-físico que realiza trocas entre o usuário e o computador. Nele são inseridos alguns softwares que administram todas as partes do sistema e apresentam de forma amigável ao usuário.

Ele também tem a função de fazer o gerenciamento dos vários usuários da máquina e é sobre esse sistema que os programas são inseridos e os recursos do computador são gerenciados, como a memória principal, as interrupções, a memória secundária e os dispositivos de entrada e saída do computador.

Um sistema operacional possui duas camadas, a primeira é chamada de Kernel, é o seu núcleo principal, uma das partes essenciais e básicas que dá suporte a conversa entre software e hardware. O segundo são os utilitários, programas utilizados para 'rodar' dentro do Kernel, ou seja, os softwares aplicativos já citados.

Importante

O Sistema Operacional deverá ser projetado de acordo com as características do hardware, as linguagens de programação e suas ferramentas.

### **Tipos de Sistemas Operacionais**

Com o avanço dos computadores foram surgindo alguns tipos de sistemas operacionais que contribuíram para o desenvolvimento do software. Os tipos de sistema operacional existentes são:

**Monotarefa (Monoprogramável)** - quando há apenas um programa em execução e todos os recursos são feitos em prol desse programa, tendo ele uma estrutura básica. Ex.: MS-DOS.

**Multitarefa (Multiprogramável)** - sistema que permite o funcionamento de vários programas, além de compartilhamento e gerenciamento de recursos, apresentando uma estrutura complexa. Ex.: Windows.

**Sistema com Múltiplos Processadores** - sistema em que existem duas ou mais CPUs conectadas e trabalhando em conjunto. Existem os fortemente acoplados, quando compartilham apenas uma memória e são controlados por um Sistema Operacional; E, os fracamente acoplados, em que cada sistema interconectados possui o seu Sistema Operacional.

**Conheça Alguns Sistemas Operacionais**

### **UNIX: Sistema Operacional Para Grandes Corporações**

É um sistema multiusuário (vários usuários em única vez) e multiprogramável, com uma estrutura mais complexa, organização de arquivos por meio de subdiretórios, garantindo a proteção das informações e redirecionamento de entrada e saída de dados.

Ele foi criado na década de 1970, por desenvolvedores da AT&T, sendo distribuído comercialmente em linguagem 'C' após 1980 e considerado um dos primeiros sistemas operacionais modernos. A partir dele foram criados conceitos importantes no mundo da computação. O Unix foi projetado para grandes universidades e corporações e após ele, foram lançados outros sistemas inspirados em sua interface gráfica e linguagem, como o BSD (Berkeley Software Distribution).

O Unix está dividido internamente em Kernel (núcleo do sistema operacional) e Interpretador de comandos SHELL (reúne a interface do sistema, executa os comandos digitados pelo usuário).

Na época, programadores pensavam em inovar, não somente na produção de sistemas operacionais utilizados em grandes corporações, mas no desenvolvimento de sistemas para usuários comuns que seriam utilizados futuramente nos computadores pessoais.

### **Mac OS: Sistema Operacional para PCs**

Uma das primeiras empresas a pensar em computadores pessoais foi a Apple, empresa fundada em 1970 por Steve Jobs. Ele lançou, inicialmente, o computador Apple I, com um sistema operacional próprio chamado de Mac OS (Macintosh Operating System) que era conhecido como System. Posteriormente lançou o Apple II, III, Macintosh e Lisa.

A cada versão nova dos computadores da linha Macintosh, o sistema System sofria modificações e melhorias. Na década de 90, foi lançado o System 7, um sistema mais avançado que permitia o uso de cores, com a vantagem de ser multitarefa, possuir a linguagem Apple Script, dentre outras características. Após isso, houve a inserção do processador PowerPC, da empresa IBM, e a possibilidade de criação de cópias por outros fabricantes. Apenas, depois da versão 7.6 o nome MAC OS foi considerado.

Com o aparecimento de problemas que atingiram drasticamente esse sistema operacional, ocasionadas pela diminuição de seu uso e domínio do sistema operacional da Microsoft, a Apple decidiu reescrever todo o código com base no Unix, sendo chamado de MAC OSX.

Esse sistema, tem como características: qualidade na interface gráfica do computador, com o lançamento do Aqua (interface gráfica que permite a produção de relevos, sombreamentos, reflexos e outros elementos de design), além de comandos diferenciados em suas últimas versões, como permissão de múltiplos toques e uma navegação baseada na intuição do usuário.

### **Outras Versões do Sistema Operacional Mac OS X**

As versões do sistema operacional Mac OS X recebem o nome de felinos, sendo algumas desenvolvidas para funcionar em tablets da Apple, Iphone e Ipod Touch, veja:

Mac OS X versão 10.0 – Cheetah;

Mac OS X versão 10.1 – Puma;

Mac OS X versão 10.2 – Jaguar;

Mac OS X versão 10.3 – Panther;

Mac OS X versão 10.4 – Tiger;

Mac OS X versão 10.5 – Leopard;

Mac OS X versão 10.6 – Snow Leopard;

Mac OS X versão 10.7 – Lion;

Mac OS X versão 10.8 – Mountain Lion.

### **Windows: Sistema Operacional em Janelas**

A palavra Windows traduzida do inglês quer dizer 'janelas', um gerenciador de interfaces que permite o usuário ver informações e se comunicar com o computador. Ele foi desenvolvido, na década de 1980, por Bill Gates, mas somente se tornou um sistema operacional a partir do Windows NT, lançado na década de 90. A partir da primeira interface, foram surgindo outras versões para Windows, como 1.01, 2.03, 2.1, 3.0, etc.

O Windows NT (New Technology) foi desenvolvido para o ambiente corporativo. Ele é multiusuário, multitarefa e multiplataforma, rodando não somente em plataformas como INTEL, mas em DEC Alpha, MIPS, etc. Uma das características dos NT é a de se transformar em servidor na internet, sendo dividido em Windows NT Server e Windows NT Workstation.

Anteriormente, não havia ainda o Windows, mas softwares que 'rodavam' no computador e eram sistemas gráficos com versões compatíveis ao sistema DOS (MS-DOS, DR-DOS, PC-DOS), sendo utilizado e criado pela Microsoft, o MS-DOS (sistema orientado por meio de linhas de comando digitadas através do teclado pelo o utilizador).

### **Outras Versões do Sistema Operacional Windows**

Cada versão foi sendo melhorada e adaptada para os usuários, trazendo uma convergência de tecnologias, além de maior desempenho e rapidez com a tecnologia de 64 bits. As versões do Windows possuem preços diferenciados, por se tratar de um software proprietário:

Windows 35;

Windowa 98;

Windows Me (Millennium Edition);

Windows 2000;

Windows XP (Experience);

Windows Server 2003;

Windows Vista;

Windows 7;

Windows 8.

### **Linux: Sistema Operacional de Código Aberto**

O sistema operacional GNU/Linux foi desenvolvido por Linus Torvalds, na Finlândia, em 1991. Ele é uma versão do SO Unix que possui código aberto e pode ser escrito e distribuído por qualquer tipo de usuário na internet, por ser um software gratuito (free software), sendo proibido a comercialização do sistema.

Qualquer pessoa poderá ver o código fonte de um sistema Linux, resolver problemas através de uma lista de discussão online, em que consultores e usuários que trabalham na manutenção do código poderão solucionar, fazer atualizações, etc. Além disso, ele dá suporte a placas, cd-rom e outros dispositivos mais ultrapassados e/ou avançados.

Das características desse sistema estão a multitarefa, multiusuário, conexão com outros tipos de sistemas operacionais, segurança quanto a proteção de processos executados na memória RAM, não há licença para seu uso, etc.

O SO Linux é composto pelo kernel e vários programas, que podem ser criados de acordo com as suas distribuições. Cada distribuição linux tem características diferentes e foram criadas para usuários específicos.

**Outras Distribuições do Sistema Operacional Linux**

Slawckaware;

Debian;

Fedora;

Red Hat;

Conectiva;

Monkey;

Ubuntu;

Mandriva;

Mint;

Opensuse;

Puppy;

Sabayon, etc.

**Segurança da Informação**

Após o crescimento do uso de sistemas de informação, comércio eletrônico e tecnologia digital as empresas se viram obrigadas a pensar na segurança de suas informações para evitar ameaças e golpes. Assim, a segurança da informação surgiu para reduzir possíveis ataques aos sistemas empresariais e domésticos. Resumindo, a segurança da informação é uma maneira de proteger os sistemas de informação contra diversos ataques, ou seja, mantendo documentações e arquivos.

**Princípios Básicos da Segurança da Informação****Disponibilidade**

É a garantia de que os sistemas e as informações de um computador estarão disponíveis quando necessário.

**Confidenciabilidade**

É a capacidade de controlar quem vê as informações e sob quais condições. Assegurar que a informação só será acessível por pessoas explicitamente autorizadas.

**Autenticidade**

Permite a verificação da identidade de uma pessoa ou agente externo de um sistema. É a confirmação exata de uma informação.

**Integridade**

Princípio em que as informações e dados serão guardados em sua forma original evitando possíveis alterações realizadas por terceiros.

**Auditoria**

É a possibilidade de rastrear os diversos passos que o processo realizou ou que uma informação foi submetida, identificando os participantes, locais e horários de cada etapa. Exame do histórico dos eventos dentro de um sistema para determinar quando e onde ocorreu violação de segurança.

**Privacidade**

Capacidade de controlar quem viu certas informações e quem realizou determinado processo para saber quem participou, o local e o horário.

### **Legalidade**

É a garantia de legalidade de uma informação de acordo com a legislação vigente.

Não Repúdio

Não há como "dizer não" sobre um sistema que foi alterado ou sobre um dado recebido.

### **Ameaças**

Uma ameaça acontece quando há uma ação sobre uma pessoa ou sobre um processo utilizando uma determinada fraqueza e causa um problema ou consequência. Sendo assim, são caracterizados como divulgação ruim, usurpação, decepção e rompimento. As ameaças podem ter origem natural, quando surgem de eventos da natureza, como terremotos ou enchentes; podem ser involuntárias, como falta de energia ou erros causados por pessoas desconhecidas; ou se tratam de ameaças voluntárias em que hackers e bandidos acessam os computadores no intuito de disseminar vírus e causar danos.

### **Tipos de Ameaça**

Ameaça Inteligente: Situação em que seu adversário possui capacidade técnica e operacional para fazer uso de algo vulnerável no sistema;

Ameaça de Análise: Após uma análise poderão descobrir as possíveis consequências da ameaça a um sistema.

Principais Ameaças ao Sistema de Informação: incêndio, problemas na eletricidade, erros no hardware e software, alterações em programas, furto de dados, invasão ao terminal de acesso, dificuldades de telecomunicação, etc.

### **Ataques**

Um ataque pode ser decorrente de um furto a um sistema de segurança no intuito de invadir sistemas e serviços. Ele pode ser dividido em ativo, passivo e destrutivo; o ativo muda os dados, o passivo libera os dados e o destrutivo proíbe qualquer acesso aos dados. Para que um ataque seja considerado bem sucedido o sistema atacado deve estar vulnerável.

### **Tipos de Ataque**

#### **Cavalo de Troia**

O cavalo de troia ou trojan horse, é um programa disfarçado que executa alguma tarefa maligna. Um exemplo: o usuário roda um jogo que conseguiu na Internet. O jogo secretamente instala o cavalo de troia, que abre uma porta TCP do micro para invasão. Alguns trojans populares são NetBus, Back Orifice e SubSeven. Há também cavalo de troia dedicado a roubar senhas e outros dados sigilosos.

#### **Quebra de Senha**

O quebrador, ou cracker, de senha é um programa usado pelo hacker para descobrir uma senha do sistema. O método mais comum consiste em testar sucessivamente as palavras de um dicionário até encontrar a senha correta.

#### **Denial Of Service (DOS)**

Ataque que consiste em sobrecarregar um servidor com uma quantidade excessiva de solicitações de serviços. Há muitas variantes, como os ataques distribuídos de negação de serviço (DDoS). Nessa variante, o agressor invade muitos computadores e instala neles um software zumbi, como o Tribal Flood Network ou o Trinoo. Quando recebem a ordem para iniciar o ataque, os zumbis bombardeiam o servidor-alvo, tirando-o do ar.

**Mail Bomb**

É a técnica de inundar um computador com mensagens eletrônicas. Em geral, o agressor usa um script para gerar um fluxo contínuo de mensagens e abarrotar a caixa postal de alguém. A sobrecarga tende a provocar negação de serviço no servidor de e-mail.

**Phreaking**

É o uso indevido de linhas telefônicas, fixas ou celulares. Conforme as companhias telefônicas foram reforçando a segurança, as técnicas tornaram-se mais complexas. Hoje, o phreaking é uma atividade elaborada, que poucos hackers dominam.

**Scanners de Portas**

Os scanners de portas são programas que buscam portas TCP abertas por onde pode ser feita uma invasão. Para que a varredura não seja percebida pela vítima, alguns scanners testam as portas de um computador durante muitos dias, em horários aleatórios.

**Smurf**

O Smurf é outro tipo de ataque de negação de serviço. O agressor envia uma rápida seqüência de solicitações de Ping (um teste para verificar se um servidor da Internet está acessível) para um endereço de broadcast. Usando spoofing, o cracker faz com que o servidor de broadcast encaminhe as respostas não para o seu endereço, mas para o da vítima. Assim, o computador-alvo é inundado pelo Ping.

**Sniffing**

O sniffer é um programa ou dispositivo que analisa o tráfego da rede. Sniffers são úteis para gerenciamento de redes. Mas nas mãos de hackers, permitem roubar senhas e outras informações sigilosas.

**Spoofing**

É a técnica de se fazer passar por outro computador da rede para conseguir acesso a um sistema. Há muitas variantes, como o spoofing de IP. Para executá-lo, o invasor usa um programa que altera o cabeçalho dos pacotes IP de modo que pareçam estar vindo de outra máquina.

**Scamming**

Técnica que visa roubar senhas e números de contas de clientes bancários enviando um e-mail falso oferecendo um serviço na página do banco.

Controles de Segurança

**Autenticar e Autorizar**

Autorizar um usuário é conceder ou negar acesso ao sistema utilizando controles de acesso no intuito de criar perfis de acesso. Com esses perfis é possível definir que tarefa será realizada por determinada pessoa. Autenticar é a comprovação de que uma pessoa que está acessando o sistema é quem ela diz ser. Ela é importante, pois limita o controle de acesso e autoriza somente determinadas pessoas o acesso a uma informação.

**Processo de Autenticação**

Identificação positiva: quando o usuário possui alguma informação em relação ao processo, como acontece quando ele possui uma senha de acesso.

Identificação proprietária: o usuário tem algum material para utilizar durante a etapa de identificação como um cartão.

Identificação Biométrica: casos em que o usuário se identifica utilizando alguma parte do corpo como a mão ou impressão digital.

## Prevenção de Riscos e Códigos Maliciosos (Malware)

### Contas e Senhas

Elabore sempre uma senha que contenha pelo menos oito caracteres, compostos de letras, números e símbolos e jamais utilize como senha seu nome, sobrenomes, números de documentos, placas de carros, números de telefones, datas que possam ser relacionadas com você ou palavras que façam parte de dicionários.

Utilize uma senha diferente para cada serviço e altere com frequência. Além disso, crie tantos usuários com privilégios normais, quantas forem as pessoas que utilizam seu computador.

### Vírus

Instale e mantenha atualizado um bom programa antivírus e atualize as assinaturas do antivírus, de preferência diariamente;

Configure o antivírus para verificar os arquivos obtidos pela Internet, discos rígidos (HDs) e unidades removíveis, como CDs, DVDs e pen drives;

Desabilite no seu programa leitor de e-mails auto-execução de arquivos anexados às mensagens;

Não execute ou abra arquivos recebidos por e-mail ou por outras fontes, mesmo que venham de pessoas conhecidas. Caso seja necessário abrir o arquivo, certifique-se que ele foi analisado pelo programa antivírus;

Utilize na elaboração de documentos formatos menos suscetíveis à propagação de vírus, tais como RTF, PDF ou PostScript;

Não utilize, no caso de arquivos comprimidos, o formato executável. Utilize o próprio formato compactado, como por exemplo Zip ou Gzip.

Worms, Bots e Botnets

Siga todas as recomendações para prevenção contra vírus;

Mantenha o sistema operacional e demais softwares sempre atualizados;

Aplique todas as correções de segurança (patches) disponibilizadas pelos fabricantes, para corrigir eventuais vulnerabilidades existentes nos Softwares utilizados;

Instale um firewall pessoal, que em alguns casos pode evitar que uma vulnerabilidade existente seja explorada ou que um worm ou bot se propague.

### Incidente de Segurança e Uso Abusivo na Rede

O incidente de segurança está relacionado a qualquer problema confirmado ou não e tem relação com redes de computadores ou sistemas de computação. Pode ser caracterizado por tentativas de acesso aos dados de um sistema, acessos não autorizados, mudanças no sistema sem prévia autorização ou sem conhecimento da execução, etc.

O uso abusivo na rede é um conceito mais difícil de ser definido, mas possui características específicas como envio de spams e correntes, distribuição de documentação protegida por direito autoral, uso indevido da internet para ameaçar e difamar pessoas, ataques a outros computadores, etc.

### Registros de Eventos (logs)

Os logs são registros de tarefas realizados com programas de computador e geralmente são detectados por firewalls. Os logs podem ser acusados no momento em que uma pessoa tenta entrar em um computador e é impedido pelo firewall. Verifique sempre os logs do firewall pessoal e de IDSs que estejam instalados no computador e confira se não é um falso positivo, antes de notificar um incidente.



### Notificações de Incidentes

Muitas vezes um computador é atacado por um programa ou pessoa mal intencionada. Caso seja um ataque proveniente de um computador, avise aos responsáveis pela máquina para que sejam tomadas medidas necessárias. No entanto, caso esse ataque venha de uma pessoa que invadiu seu sistema com um computador é importante avisá-lo de tal atitude para que tome as medidas cabíveis.

### Segurança na Internet

A internet já faz parte do cotidiano da maioria das pessoas e atualmente elas não se imaginam sem ela. Os sites são usados para trabalhos escolares, conhecer pessoas, realizar pagamentos, publicar documentos e fotos, estudar, ouvir música, assistir vídeos, dentre outros.

No entanto, ela também possui muitos perigos, pois qualquer um está sujeito a sofrer ataques de hackers ou ter seu computador invadido por vírus ao acessar emails e documentos mal intencionados. É necessário que as pessoas saibam os riscos e estabeleçam medidas de segurança na internet.

Os principais riscos na internet são:

Acesso de conteúdos indevidos e ofensivos: sites que contenham imagens relacionadas a pornografia, pedofilia e fotos que alimentem ódio a determinada raça ou população;

Contato com pessoas ruins: sequestros, furtos e estelionatos são apenas alguns dos golpes que podem ser aplicados na internet e os golpistas aproveitam o anonimato da internet para cometer esses crimes;

Roubo de Identidade: pessoas mal-intencionadas podem utilizar sua identidade para aplicar golpes causando sérios problemas a seu nome e reputação;

Roubo e perda de dados: com códigos e ações na internet, diversas pessoas podem roubar ou apagar dados do seu computador;

Phishing: fraude na qual uma pessoa tenta roubar dados de um usuário utilizando engenharia social e mensagens eletrônicas. Eles podem fazer páginas falsas de redes sociais, bancos e lojas de venda eletrônica;

Invasão de Privacidade: seus dados, documentos e fotos muitas vezes podem ser acessados com apenas um clique na internet. Muitas redes sociais possuem condições de privacidade específicas e é importante estar atento a tudo que é compartilhado em seu perfil.

Problemas para Excluir Dados: um dado inserido na internet as vezes não pode ser apagado ou pode ter sido repassado antes da exclusão. Sendo assim, o que foi dito poderá ser acessado por pessoas do seu meio social.

Plágio e Direitos Autorais: muitos conteúdos e arquivos na internet são roubados causando transtornos para milhares de usuários. O autor do plágio pode ser processado e terá que responder judicialmente.

Engenharia Social: ações realizadas acessando dados sigilosos de empresas ou utilizando sistemas para enganar pessoas aplicando golpes.

### Browsers

Mantenha seu browser sempre atualizado e desative a execução de programas Java na configuração, a menos que seja estritamente necessário. Além disso, desative a execução de JavaScripts antes de entrar em uma página desconhecida. Outra maneira de manter a segurança do seu computador ao acessar um browser é permitir que os programas ActiveX sejam executados em seu computador apenas quando vierem de sites conhecidos e confiáveis.

Tenha maior controle sobre o uso de cookies e caso você queira ter maior privacidade ao navegar na Internet, bloqueie pop-up windows e permita apenas para sites conhecidos e confiáveis ou onde fo-

rem realmente necessárias. Certifique-se da procedência do site e da utilização de conexões seguras ao realizar transações via Web (bancos, compras coletivas, etc).

### **Protegendo Seus E-Mails**

Mantenha seu programa leitor de e-mails sempre atualizado;

Não clique em links no conteúdo do e-mail. Se você realmente quiser acessar àquela página, digite o endereço diretamente no seu browser;

Desligue as opções que permitem abrir ou executar automaticamente arquivos ou programas anexados às mensagens;

Não abra arquivos ou execute programas anexados aos e-mails. Sem antes verificá-los com um anti-vírus;

Desconfie sempre dos arquivos anexados à mensagem, mesmo que tenham sido enviados por pessoas ou instituições conhecidas. O endereço do remetente pode ter sido forjado e o arquivo anexo pode ser, por exemplo, um vírus ou um cavalo de tróia;

Faça download de programas diretamente do site do fabricante;

Somente acesse sites de instituições financeiras e de comércio eletrônico digitando o endereço diretamente no seu browser, nunca clicando em um link existente em uma página ou em um e-mail.

### **Programa de Bate Papo**

Mantenha seu programa de troca de mensagens sempre atualizado;

Não aceite arquivos de pessoas desconhecidas, principalmente programas de computadores;

Evite fornecer muita informação, principalmente para pessoas que você acabou de conhecer e não libere informações sensíveis, tais como senhas ou números de cartões de crédito;

Configure o programa para ocultar o seu endereço IP.

### **Programas de Distribuição de Arquivos e Compartilhamento**

Mantenha seu programa de distribuição de arquivos sempre atualizado e bem configurado e certifique-se que os arquivos obtidos ou distribuídos são livres, ou seja, não violam as leis de direitos autorais. Tenha um bom antivírus e estabeleça senhas para os compartilhamentos, caso seja estritamente necessário compartilhar recursos do seu computador.

Faça Cópias de Segurança (Backup): Procure sempre fazer cópias dos dados do computador regularmente e criptografe dados sensíveis. Armazene as cópias em local acondicionado, de acesso restrito e com segurança física e considere a necessidade de armazenar as cópias em um local diferente daquele onde está o computador.

### **Fraudes na Internet**

A fraude ocorre quando uma pessoa tenta enganar a outra a fim de obter dados sigilosos e pessoais para ser usado em benefício próprio. É importante que todos os usuários tomem as seguintes precauções: não forneça dados pessoais, números de cartões e senhas através de contato telefônico; fique atento a e-mails ou telefonemas solicitando informações pessoais; não acesse sites ou seguir links recebidos por e-mail e sempre que houver dúvida sobre a real identidade do autor de uma mensagem ou ligação telefônica, entrar em contato com a instituição, provedor ou empresa para verificar a veracidade dos fatos.

### **Transações Bancárias ou Comerciais**

Siga todas as recomendações sobre utilização do programa leitor de e-mails e do browser de maneira segura;

Fique atento e prevena-se dos ataques de engenharia social;

Realize transações somente em sites de instituições que você considere confiáveis;

Procure sempre digitar em seu browser o endereço desejado. Não utilize links em páginas de terceiros ou recebidos por e-mail;

Certifique-se de que o endereço apresentado em seu browser corresponde ao site que você realmente quer acessar antes de realizar qualquer ação;

Certifique-se que o site faça uso de conexão segura (ou seja, que os dados transmitidos entre seu browser e o site serão criptografados).

Antes de aceitar um novo certificado verifique junto à instituição que mantém o site sobre sua emissão e quais são os dados nele contidos. Então verifique o certificado do site antes de iniciar qualquer transação para assegurar-se que ele foi emitido para a instituição que se deseja acessar e está dentro do prazo de validade;

Não acesse sites de comércio eletrônico ou Internet Banking através de computadores de terceiros;

Desligue sua Webcam (caso você possua alguma) ao acessar um site de comércio eletrônico ou Internet banking.

### **"Boatos" na Internet**

Um boato é um conteúdo divulgado na internet que é falso e que muitas vezes se trata de uma tentativa de golpe. Eles podem causar diversos problemas como prejudicar uma pessoa ou empresa, aumentar a quantidade de emails de um determinado lugar, reduzir a credibilidade de uma empresa ou espalhar vírus pela internet.

As principais características dos boatos são: a afirmação de que aquilo não é um boato, possui erros de ortografia, afirma que se aquilo não for lido algo grave poderá ocorrer (como as conhecidas "correntes"), foi enviado para diversas pessoas e garante retorno financeiro para quem lê.

Dicas para Prevenção de Boatos:

-Verifique sempre a procedência da mensagem e se o fato sendo descrito é verídico;

-Verifique em sites especializados e em publicações da área se o e-mail recebido já não está catalogado como um boato.

### **Privacidade na Internet**

Dados Pessoais Sites, Blogs e Sites de Relacionamento

Evite disponibilizar seus dados pessoais ou de familiares e amigos (e-mail, telefone, endereço, data de aniversário, etc);

Procure não repassar dados sobre o seu computador ou softwares que utiliza;

Evite fornecer informações do seu cotidiano (como, por exemplo, hora que saiu e voltou para casa, data de uma viagem programada, horário que foi ao caixa eletrônico, etc);

Nunca forneça informações sensíveis (como senhas e números de cartão de crédito), a menos que esteja sendo realizada uma transação (comercial ou financeira) e se tenha certeza da idoneidade da instituição que mantém o site.

### **Dados Armazenados em um Disco Rígido**

Criptografe todos os dados sensíveis, principalmente se for um notebook e sobrescreva os dados do disco rígido antes de vender ou se desfazer do seu computador usado.

Telefones Celulares, PDAs e Outros Aparelhos com Bluetooth

Mantenha o bluetooth do seu aparelho desabilitado e somente faça isso quando for necessário;

Fique atento às notícias, principalmente àquelas sobre segurança, veiculadas no site do fabricante do seu aparelho;

Aplice todas as correções de segurança (patches) que forem disponibilizadas pelo fabricante do seu aparelho, para evitar que possua vulnerabilidades;

Caso você tenha comprado um aparelho usado, restaurar as opções de fábrica.

## **Windows 8**

É o sistema operacional da Microsoft que substituiu o Windows 7 em tablets, computadores, notebooks, celulares, etc. Ele trouxe diversas mudanças, principalmente no layout, que acabou surpreendendo milhares de usuários acostumados com o antigo visual desse sistema.

A tela inicial completamente alterada foi a mudança que mais impactou os usuários. Nela encontra-se todas as aplicações do computador que ficavam no Menu Iniciar e também é possível visualizar previsão do tempo, cotação da bolsa, etc. O usuário tem que organizar as pequenas miniaturas que aparecem em sua tela inicial para ter acesso aos programas que mais utiliza.

Caso você fique perdido no novo sistema ou dentro de uma pasta, clique com o botão direito e irá aparecer um painel no rodapé da tela. Caso você esteja utilizando uma das pastas e não encontre algum comando, clique com o botão direito do mouse para que esse painel apareça.

### **Organizar a Tela Start do Windows 8**

Essa tela nova funciona como o antigo Menu Iniciar e consiste em um mosaico com imagens animadas. Cada mosaico representa um aplicativo que está instalado no computador. Os atalhos dessa área de trabalho, que representam aplicativos de versões anteriores, ficam com o nome na parte de cima e um pequeno ícone na parte inferior. Novos mosaicos possuem tamanhos diferentes, cores diferentes e são atualizados automaticamente.

A tela pode ser customizada conforme a conveniência do usuário. Alguns utilitários não aparecem nessa tela, mas podem ser encontrados clicando com o botão direito do mouse em um espaço vazio da tela. Se deseja que um desses aplicativos apareça na sua tela inicial, clique com o botão direito sobre o ícone e vá para a opção Fixar na Tela Inicial.

### **Charms Bar**

O objetivo do Windows 8 é ter uma tela mais limpa e esse recurso possibilita “esconder” algumas configurações e aplicações. É uma barra localizada na lateral que pode ser acessada colocando o mouse no canto direito e inferior da tela ou clicando no atalho Tecla do Windows + C. Essa função substitui a barra de ferramentas presente no sistema e configurada de acordo com a página em que você está.

### **Personalizando o Windows 8**

#### **Cor do Papel de Parede**

Com a Charm Bar ativada, digite Personalizar na busca em configurações. Depois escolha a opção tela inicial e em seguida escolha a cor da tela. O usuário também pode selecionar desenhos durante a personalização do papel de parede.

#### **Redimensionar as Tiles**

Na tela esses mosaicos ficam uns maiores que os outros, mas isso pode ser alterado clicando com o botão direito na divisão entre eles e optando pela opção menor. Você pode deixar maior os aplicativos que você quiser destacar no computador.

#### **Grupos de Aplicativos**

Pode-se criar divisões e grupos para unir programas parecidos. Isso pode ser feito várias vezes e os grupos podem ser renomeados.

### Visualizar as Pastas

A interface do programas no computador podem ser vistos de maneira horizontal com painéis dispostos lado a lado. Para passar de um painel para outro é necessário usar a barra de rolagem que fica no rodapé.

### Compartilhar e Receber

Comando utilizado para compartilhar conteúdo, enviar uma foto, etc. Tecle Windows + C, clique na opção Compartilhar e depois escolha qual meio vai usar. Há também a opção Dispositivo que é usada para receber e enviar conteúdos de aparelhos conectados ao computador.

### Alternar Tarefas

Com o atalho Alt + Tab, é possível mudar entre os programas abertos no desktop e os aplicativos novos do SO. Com o atalho Windows + Tab é possível abrir uma lista na lateral esquerda que mostra os aplicativos modernos.

### Telas Lado a Lado

Esse sistema operacional não trabalha com o conceito de janelas, mas o usuário pode usar dois programas ao mesmo tempo. É indicado para quem precisa acompanhar o Facebook e o Twitter, pois ocupa  $\frac{1}{4}$  da tela do computador.

### Visualizar Imagens

O sistema operacional agora faz com que cada vez que você clica em uma figura, um programa específico abre e isso pode deixar seu sistema lento. Para alterar isso é preciso ir em Programas – Programas Default – Selecionar Windows Photo Viewer e marcar a caixa Set this Program as Default.

### Imagem e Senha

O usuário pode utilizar uma imagem como senha ao invés de escolher uma senha digitada. Para fazer isso, acesse a Charm Bar, selecione a opção Settings e logo em seguida clique em More PC settings. Acesse a opção Usuários e depois clique na opção “Criar uma senha com imagem”. Em seguida, o computador pedirá para você colocar sua senha e redirecionará para uma tela com um pequeno texto e dando a opção para escolher uma foto. Escolha uma imagem no seu computador e verifique se a imagem está correta clicando em “Use this Picture”. Você terá que desenhar três formas em touch ou com o mouse: uma linha reta, um círculo e um ponto. Depois, finalize o processo e sua senha estará pronta. Na próxima vez, repita os movimentos para acessar seu computador.

### Internet Explorer no Windows 8

Se você clicar no quadrinho Internet Explorer da página inicial, você terá acesso ao software sem a barra de ferramentas e menus.

O que é o BrOffice.org?

O BrOffice.org ou, simplesmente, BrOffice é um conjunto de programas para escritório. O que nós também podemos dizer chamar de suite. O BrOffice encontra-se disponível para download no site [www.broffice.org.br](http://www.broffice.org.br). Capaz de atender perfeitamente a maioria das necessidades de corporações e usuários domésticos com recursos compatíveis aos oferecidos pelos principais aplicativos comerciais. BrOffice é também multiplataforma, ou seja, ele pode ser instalado em diferentes sistemas operacionais, como por exemplo: Windows, Linux e FreeBSD.

O BrOffice.org é um produto nacional, mantido por uma equipe brasileira, baseado num conjunto de programas mundialmente conhecido: o OpenOffice.org. Então, guardando-se as devidas proporções, BrOffice.org e OpenOffice.org são a mesma coisa. Claro que o BrOffice é mais adequado para o público e as necessidades de trabalho das empresas brasileiras, por isso há uma orientação para que

substitua o Microsoft Office nos órgãos do Governo Federal e estaduais. Por isso, que atualmente, tantas autarquias exigem em edital conhecimentos de BrOffice dos seus futuros servidores.

### **Open Source e Freeware**

Como o BrOffice é um software livre e gratuito, logo pode-se afirmar que ele é Open Source e também Freeware. Segundo a definição criada por Richard Stallman, fundador da Free Software Foundation, Software Livre é qualquer programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído sem nenhuma restrição. A liberdade de tais diretrizes é central ao conceito, o qual se opõe ao conceito de software proprietário, mas não ao software que é vendido almejando lucro (software comercial). Em outras palavras podemos dizer que o fato do Software ser livre não quer dizer que seja gratuito. A maneira usual de distribuição de software livre é anexar a este uma licença de software livre, e tornar o código fonte do programa disponível (Open Source).

Mas, como o BrOffice além de Livre é Gratuito, significa dizer que ele é um Freeware. Por definição, Software gratuito ou Freeware é qualquer programa de computador cuja utilização não implica no pagamento de licenças de uso ou royalties.

E quais são seus Programas?

Os programas que formam BrOffice.org são: Writer – um processador de textos à altura com o Word; o Calc – para confecção de planilhas eletrônicas, assim como o Excel; o Impress – é um editor gráfico voltado para a confecção de apresentação em slides, similar ao Powerpoint; Base - para criação de bancos de dados, com funcionalidades que lembram o Access; Draw - para criação de desenhos vetoriais, não há concorrentes para este aplicativo no Microsoft Office.

Nosso alvo de estudo será o trio de programas: Impress, Writer e Calc. Os programas serão apresentados nesta ordem buscando tornar mais confortável, ao candidato, a localização do conteúdo de acordo com o edital publicado pelo NCE – Núcleo de Computação e Eletrônica, para o concurso da PRF em 2007 e mantido pelo CESPE/UnB, instituição que retoma o processo de seleção da PRF. O edital cita ainda um aplicativo cliente de e-mails, o Mozilla Thunderbird.

### **Principais Vantagens no uso do BrOffice**

Conforme já foi dito é gratuito;

É um Software Livre, portanto permite modificações e pode ser distribuído livremente;

É mais leve que o Microsoft Office tanto no que se refere à instalação quanto no tamanho dos documentos gerados, cerca de 70% menor, o que auxilia no desempenho da máquina;

É multiplataforma, ou seja, permite ser instalado em sistemas operacionais diferentes;

Capaz de abrir, editar e salvar seus documentos em formatos criados em outros aplicativos, principalmente aqueles criados no MS-Office (doc, xls e ppt);

Exporta arquivos diretamente para o formato PDF;

A integração entre os arquivos do BrOffice é bem maior, todos os arquivos se comportam como se fossem de um mesmo programa, tornando mais fácil a edição de diferentes tipos de documentos;

É similar ao MS-Office, facilitando o aprendizado de quem está migrando da suíte Microsoft.

### **Desvantagens**

O programa demora para carregar, mas ao abrir um documento (Writer ou Calc, por exemplo), os demais abrem mais rapidamente pois todos ficam sob uma mesma janela. Ele vem com programa de ilustração, mas não tem cliente de e-mail, editor de páginas HTML (apesar do Writer também poder editar nesta linguagem), publicador de calendários e agenda. Outro problema é que tabelas e textos perdem a formatação quando colados no programa.

Boa parte de seus comandos, principalmente os de teclas de atalho não são os mesmos do BrOffice.

## Novos Formatos de Arquivos

A partir da versão 2.0 o BrOffice.Org passou a adotar o formato XML denominado OpenDocument, desenvolvido pela OASIS (Organization for the Advancement of Structured Information Standards), ou Organização para o Avanço de Padrões Estruturados de Informações que visa padronizar o formato de arquivos para facilitar seu intercâmbio entre diversas aplicações existentes no mercado, passando a usar os seguintes formatos de arquivo:

Formato do Documento	Extensão do Arquivo
Texto e modelo de texto do opendocument	*.odt e *.ott
Documento mestre do opendocument	*.odm
Documento html	.html
Modelo de documento html	*.oth
Planilha e modelo de planilha do opendocument	*.ods e *.ots
Desenho e modelo de desenho do opendocument	*.odg e *.otg
Apresentação e modelo de apresentação do opendocument	*.odp e *.otp

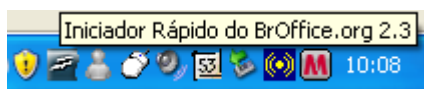
## Iniciando o BrOffice

Uma vez instalado em um computador que possua o Sistema Operacional Windows, um novo documento do BrOffice pode ser iniciado de três formas:

Abrindo o menu Iniciar / Todos os Programas / BrOffice.org 2.0.x e escolhendo o aplicativo que desejar (Writer, Calc, Impress, Base, Draw ou Math);

Com o botão direito do mouse sobre um ponto vazio da Área de Trabalho, clicar em Novo / Texto\* do OpenDocument (\*ou Planilha, Apresentação, etc.).

Clicando com o botão direito do mouse sobre o ícone do BrOffice.org na Área de Notificação (Bandeja), da Barra de Tarefas.



Usando Menus, Barras de Ferramentas e Teclas de Atalho

Você pode usar comandos de menu, barras de ferramentas, teclas de atalho ou uma combinação dos três métodos para executar tarefas no BrOffice.org. Ícones em uma barra de ferramentas ou teclas de atalho podem acelerar tarefas executadas com frequência, como copiar e colar texto em um documento. Mas, vale lembrar aos usuários de Microsoft Office que nem todos os comandos são iguais quanto aos ícones, atalhos e comandos na Barra de Menus.

### Usando Menus

A barra de menus na parte superior da janela do BrOffice.org lista os comandos que você pode usar com os itens selecionados atualmente. Clique no nome de um menu e clique no comando que deseja usar. Comandos indisponíveis ou desabilitados são exibidos em cinza.

Por exemplo, para abrir um documento, pressione Ctrl+O, ou seja, mantenha pressionada a tecla Ctrl e pressione a tecla O. Ou usar a tecla ALT em conjunto com a letra sublinhada do respectivo menu.

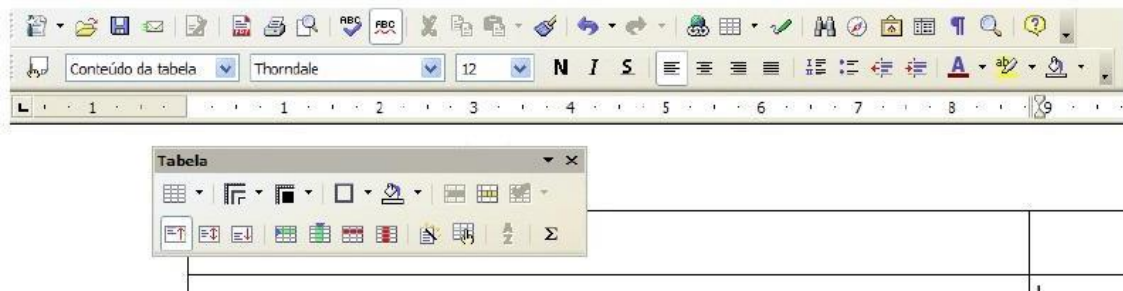
Exemplo: ALT+A para abrir o menu Arquivo.

Outra maneira de acessar os comandos do BrOffice.org é por meio de um menu de contexto. Para abrir um menu de contexto, selecione um trecho de texto ou um objeto no documento e clique com o botão direito do mouse.

### Barras de Ferramentas

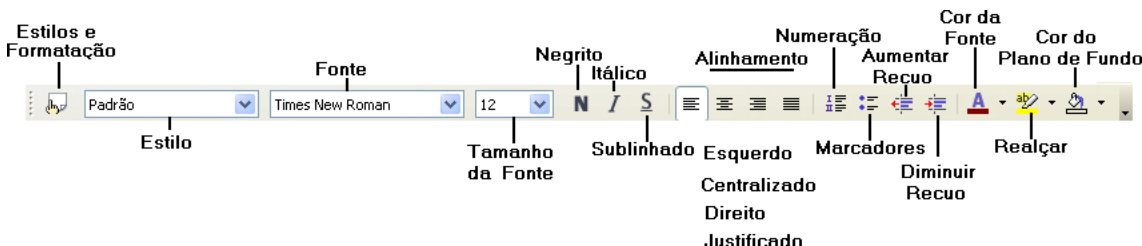
Uma barra de ferramentas é um conjunto de ícones que representam comandos comuns. Para acessar um desses comandos, clique no ícone do comando na barra de ferramentas. A barra de ferramentas Padrão localiza-se embaixo dos menus, na parte superior de cada janela de aplicativo do BrOffice.org.

Na figura abaixo, a barra de ferramentas Formatação que contém ferramentas para formatar texto fica embaixo da barra de ferramentas Padrão. A barra de ferramentas Tabela representa uma barra de ferramentas separada ou flutuante que se abre quando o cursor está em uma tabela.

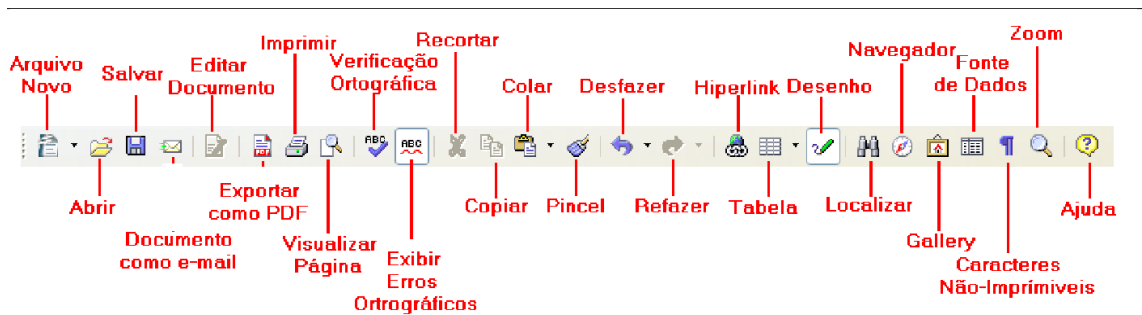


Se posicionar o ponteiro do mouse sobre um ícone da barra de ferramentas, o nome do comando que o ícone representa será exibido em uma caixa amarela. Para ver dicas de ajuda adicionais, pressione Shift+F1 e depois

**Aponte Para o Ícone da Barra de Ferramentas.**



Os comandos mais cobrados em provas de concursos são aqueles que se encontram na barra de ferramentas Padrão e também na barra de ferramentas Formatação. A figura abaixo apresenta ícones da barra de ferramentas Padrão. Perceba que alguns ícones encontram-se na cor cinza, isso ocorre porque estes comandos necessitam de algum pré-requisito para estarem ativos. Por exemplo, os comandos COPIAR e RECORTAR só ficarão ativos se houver algum item selecionado no documento.




**Mostrando e Ocultando Barras de Ferramentas**

As barras de ferramentas, na maioria, abrem-se e fecham-se com base nas suas ações. Por exemplo, ao clicar em uma tabela no documento de texto, abre-se a barra de ferramentas Tabela. Ao clicar fora da tabela, a barra de ferramentas fecha-se automaticamente.

O BrOffice.org controla o estado, o tamanho e a posição das barras de ferramentas. Quando você fecha uma barra de ferramentas flutuante, ela continua fechada até você clicar em Exibir / Barras de ferramentas e sobre o nome da barra de ferramentas que pretende exibir.



Algumas barras de ferramentas também podem ser acionadas por botões disponíveis em outras barras de ferramentas padrão, por exemplo, o botão  aciona a barra de ferramentas desenho.

Para mostrar uma barra de ferramentas oculta

Clique em Exibir / Barra de ferramentas e clique no nome da barra de ferramentas

Movendo e dimensionando barras de ferramentas

Para mover uma barra de ferramentas, arraste a alça na borda esquerda de uma barra de ferramentas encaixada ou a barra de título de uma barra de ferramentas flutuante para outra posição na tela. Para anexar a barra de ferramentas, arraste-a para uma borda da janela do programa.



Para redimensionar uma barra de ferramentas flutuante, arraste uma borda da barra de ferramentas. Por exemplo, você pode arrastar a barra de ferramentas Tabela horizontal para uma forma vertical.

Para adicionar ou remover ícones de uma barra de ferramentas, clique na seta na extremidade da barra de ferramentas, escolha Botões visíveis e selecione um botão. Para adicionar mais comandos à barra de ferramentas, clique na seta na extremidade da barra de ferramentas e escolha Personalizar barra de ferramentas.

### Teclas de Atalho

Você pode usar teclas de atalho para executar rapidamente tarefas comuns no BrOffice.org. Teclas de atalho são ou uma ou mais teclas que você pressiona no teclado para concluir uma tarefa. Por exemplo, pressione Ctrl+S para salvar o documento atual ou Ctrl+B para formatar um texto em negrito.

Nas tabelas a seguir, você encontrará listas dos atalhos do teclado mais usados para o BrOffice.org. Pressione F1 e abra a Ajuda On-line para localizar listas completas de todos os atalhos do teclado disponíveis e listas de atalhos do teclado para serem usados em lugar do mouse.

O seu sistema talvez use um mapeamento do teclado que utiliza pressionamentos de teclas para ações do sistema. Neste caso, o BrOffice.org não responde ao mesmo comando do teclado. Na maioria dos casos, você pode atribuir outras teclas às configurações do sistema ou ao BrOffice.org.

Atalhos do teclado para todos os aplicativos BrOffice.org

Ação	Atalho
Abre um documento	Ctrl+o
Salva o documento atual	Ctrl+s
Cria um novo documento	Ctrl+n
Abre a caixa de diálogo modelos e documentos	Ctrl+ shift+n
Imprime o documento atual	Ctrl+p
Sai do openoffice.org	Ctrl+q
Recorta texto ou objetos selecionados e os armazena na área de transferência	Ctrl+x
Copia texto ou objetos selecionados na área de transferência	Ctrl+c
Cola conteúdo da área de transferência	Ctrl+v
Abre a caixa de diálogo colar especial	Shift+ctrl+v
Seleciona tudo	Ctrl+a
Desfaz a última ação	Ctrl+z
Abre a caixa de diálogo localizar e substituir	Ctrl+f
Continua a procura do último termo da pesquisa	Ctrl+shift+f

### Usando Assistentes

Um assistente é uma série de caixas de diálogo que guiam você no processo de criar um documento, como por exemplo, uma carta. Para iniciar um assistente, clique em Arquivo / Assistentes em qualquer aplicativo do BrOffice.org.

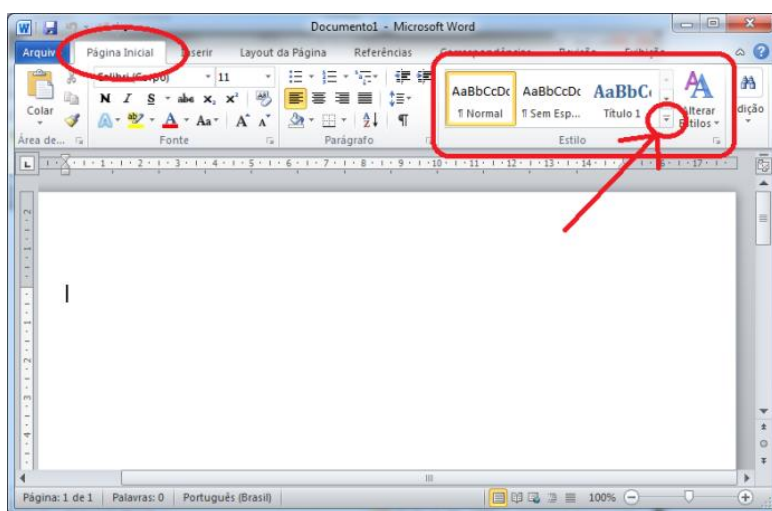
Alguns assistentes são iniciados automaticamente, por exemplo, quando você abre o Impress automaticamente ele abre o Assistente de Apresentações.

### Edição e Formatação de Textos

Editor de Texto:

Edição e formatação de textos (operações do menu: Formatar, Inserir tabelas, Exibir – cabeçalho e rodapé, Arquivo – configurar página e impressão, Ferramentas – ortografia e gramática).

Editores de texto: Os editores de texto são softwares aplicativos destinados a criação e edição de textos como cartas, currículos, memorandos, ofícios e demais tipos de documentos de texto.

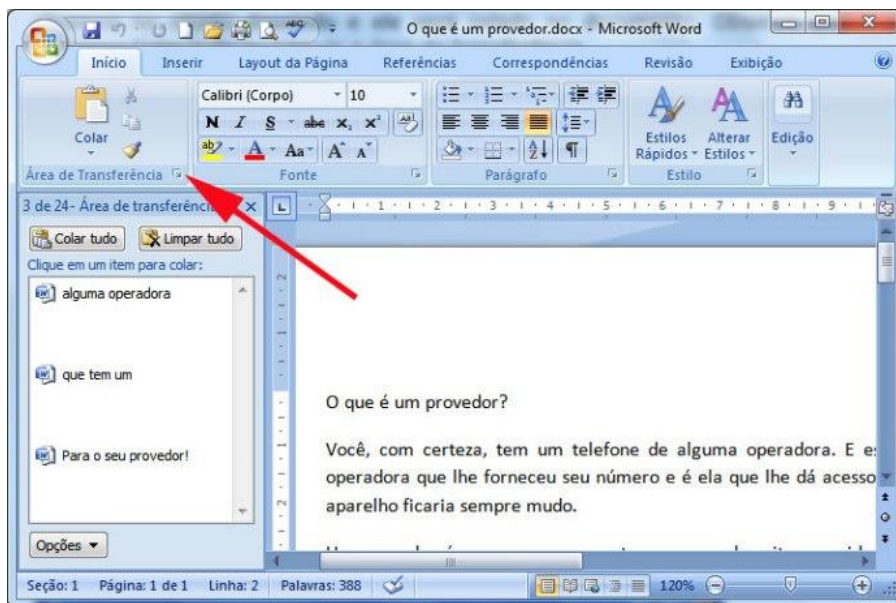


O Microsoft Word é o programa utilizado para criar e editar textos da Empresa Microsoft. Geralmente chamado apenas de Word, ditou padrões de layout e comandos para diversos outros softwares utilizados para a mesma finalidade. O Word faz parte do conjunto de programas chamado Microsoft Office, que foi especialmente desenvolvido para atender as necessidades de uso de um escritório, mas é largamente utilizado em empresas e por usuários domésticos. O conjunto de programas que compõem o Office é proprietário e comercializável.

Algumas informações básicas são necessárias para entendermos e usarmos o Word 2010. Ele é composto por Guias. Cada guia possui um grupo de elementos formado por uma série de botões de comandos.

### Edição e Formatação de Textos área de Transferência

Guia Página Inicial, Grupo Área de Transferência



A área de transferência armazena temporariamente trechos de textos copiados ou recortados para facilitar seu gerenciamento.

Ela está presente na Guia Início e é composta pelos botões de comando Copiar, Recortar, Colar e Pincel de Formatação.

Se optarmos por Mostrar o Painel de Tarefas da Área de Transferência, poderemos verificar todo o conteúdo que foi copiado ou recortado e escolher se desejamos colar ou excluir esse conteúdo. Mas o que vem a ser copiar, recortar e colar? copiar: quando desejamos duplicar um texto ou objeto (uma imagem, por exemplo) que já existe em um texto, podemos selecioná-la e clicar em copiar, ou usar as teclas de atalho CTRL+C. Este procedimento armazena o que foi selecionado, temporariamente na memória do computador e mantém o que foi selecionado no texto original.

recortar: permite retirar o texto ou objeto selecionado de uma parte do texto e colocá-lo em outro lugar do mesmo arquivo ou em outro documento. Quando utilizamos o recortar, o que foi recortado desaparece do texto original e fica armazenado temporariamente na memória do computador.

colar: aplica no lugar selecionado o que foi copiado ou colado. Para entendermos bem os procedimentos acima mencionados, podemos imaginar uma revista cheia de figuras. Quando desejamos copiar uma figura a transferimos para um outro papel, mas mantemos a original. Quando desejamos retirar a figura da revista e colocá-la em outro local, recortamos a figura fazendo sua remoção do local original.

Pincel de formatação: este botão de comando copia a formatação aplicada em uma fonte e a transfere para o texto que será selecionado com o pincel. Os passos necessários para realizar esse procedimento são:

- a) Selecionar o texto do qual desejamos copiar a formatação;
- b) Clicar no botão de comando Formatar Pincel;
- c) O ponteiro do mouse irá se transformar no desenho de um pincel e com ele devemos selecionar o texto onde desejamos aplicar a formatação copiada.

Configuração de fonte Configurar uma fonte é realizar alterações na estrutura dessa fonte. Podemos realizar diversos tipos de formatação em uma fonte como: alterar o tipo da fonte, seu tamanho, sua cor, entre outros.

No Word 2010, o caminho mais rápido para realizar a formatação da fonte, é a Guia Página Inicial, Grupo Fonte.

## Guia Página Inicial, Grupo Fonte

Neste grupo, encontramos os seguintes botões de comando:

1) fonte (ctrl+Shift+f): Através dele, após selecionar o texto desejados, alteramos o layout da fonte. Podemos escolher, por exemplo, Arial, Bell MT, Alegrian, entre outras.

2) tamanho da fonte (ctrl+Shift+P): Após selecionar a fonte, podemos escolher um tamanho para esta fonte através deste comando.

3) Aumentar Fonte (CTRL+>) e Reduzir Fonte (CTRL+<):

Como os próprios nomes sugerem, com estes botões de comando, é possível tornar a fonte maior ou menor em relação ao seu tamanho atual.

4) Maiúsculas e Minúsculas: Altera o texto selecionado para letras MAIÚSCULAS, minúsculas ou outros usos comuns de maiúsculas/minúsculas.

5) Limpar formatação: Retira as formatações aplicadas em um texto, voltando suas configurações para o estado inicial.

6) Negrito (CTRL+N): Aplica um efeito no texto deixando-o com maior espessura. Exemplo: texto com negrito aplicado.

7) itálico (ctrl+i): Aplica um efeito no texto selecionado, deixando-o com eixo um pouco inclinado. Exemplo: texto com itálico aplicado.

8) Sublinhado (CTRL+S): É usado para sublinhar o texto selecionado. Através desse comando também é possível escolher o estilo e cor para o sublinhado. Exemplos: sublinhado simples, sublinhado com estilo.

9) Tachado: Desenha uma linha no meio do texto selecionado. Exemplo: texto tachado.

10) Subscrito (CTRL+=): Faz com que a letra ou texto selecionado fique abaixo da linha de base do texto. Muito usado para fórmulas de Química e equações matemáticas. Exemplo: H<sub>2</sub> O.

11) Sobrescrito (CTRL+Shift++): Faz com que a letra ou texto selecionado fique acima da linha do texto. Exemplo: x<sup>2</sup>.

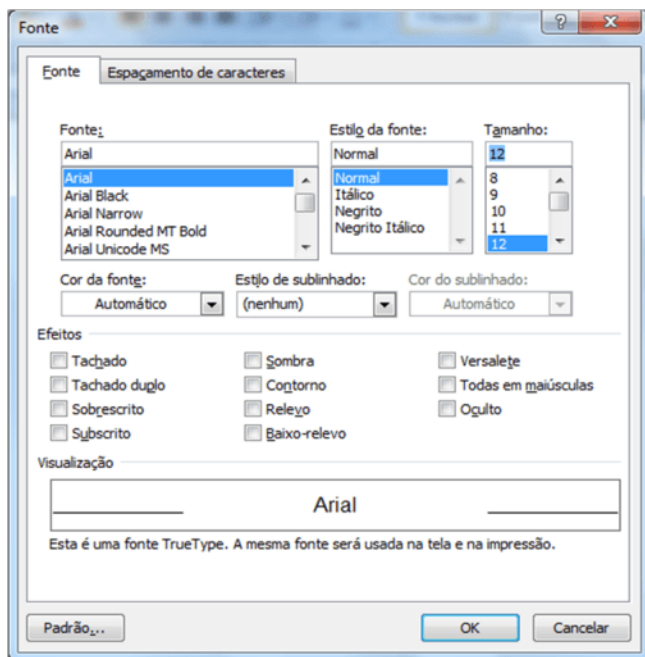
12) Efeitos de texto: Aplica um efeito visual ao texto selecionado, como sombra, brilho, reflexo.

13) Cor do realce do texto: Faz com que o texto seja realçado, como se tivesse sido marcado com uma caneta marca texto. Exemplo: texto realçado.

14) Cor da fonte: Altera a cor do texto selecionado.

Ainda podemos encontrar mais comandos para formatar uma fonte na janela Fonte.

Com os comandos disponíveis nesta janela, é possível alterar o tipo de fonte, o tamanho, aumentar ou diminuir a fonte, aplicar negrito, itálico, sublinhado, fazer com que os caracteres selecionados fiquem sobrescritos ou subscritos, formatá-los para maiúsculas ou minúsculas, alterar a cor da fonte e o estilo de sublinhado e aplicar outros efeitos como tachado, tachado duplo, sombra, contorno, relevo entre outros.



Formatar Fonte Configuração de Parágrafo e Estilo

Guia Página Inicial, Grupo Parágrafo

Os grupos Parágrafo e Estilo Também são Encontrados na Guia Página Inicial.

No grupo Parágrafo, encontramos os seguintes botões de comando:

1) marcadores: permite a criação de uma lista com marcadores. Para isto, basta selecionar uma lista de itens e clicar sobre este botão. Para implementar uma lista com níveis distintos, após a aplicação dos marcadores, use o aumentar ou diminuir recuo, que veremos a seguir.

Exemplo de lista com marcadores e vários níveis:

- Mauro e Cida
- Camila o Isabela
- João e Elisabeti
- Jaqueline
- Beatriz
- Olívia

2) Numeração: semelhante aos marcadores, cria listas numeradas. Exemplo de lista numerada:

Convidados:

1. Marisa e Wilian a. Letícia b. Lívia
2. Michele e Sérgio a. Giovana
3. Simone e Alexandre a. Vinícius b. Manuela
4. Elaine e Nilton a. Linda
5. Cristiane e Ademir a. Evandro b. Andrew

3) Lista de vários níveis: facilita a criação de listas com níveis diferenciados.

4) Diminuir Recuo: Diminui o recuo do parágrafo.

- 5) Aumentar Recuo: Aumenta o recuo do parágrafo.
- 6) Classificar: Coloca o texto selecionado em ordem alfabética ou classifica dados numéricos.
- 7) Mostrar Tudo (CTRL+\*): Mostra marcas de parágrafo e outros símbolos de formatação ocultos.

Essas marcas não são imprimíveis.

Auxilia na manutenção ou cópia de formatação de documentos específicos, identificando cada ação usada no teclado com um símbolo diferente.

- 8) Alinhar Texto à Esquerda (CTRL + O): Alinha o texto selecionado à esquerda da tela.
- 9) Centralizar: Alinha o texto selecionado de forma centralizada na página.
- 10) alinhar texto à Direita (ctrl+G): Alinha o texto selecionado à direita da tela.
- 11) Justificar (CTRL+J): Alinha o texto às margens esquerda e direita, adicionando espaço extra entre as palavras conforme necessário. Este recurso promove uma aparência organizada nas laterais esquerda e direita da página.
- 12) Espaçamento de Linha e Parágrafo: Altera o espaçamento entre linhas de texto. Também podemos personalizar a quantidade de espaço adicionado antes e depois dos parágrafos.
- 13) Sombreamento: Permite colorir o plano de fundo atrás do texto ou do parágrafo selecionado.
- 14) Bordas: Permite inserir bordas diferenciadas no texto ou palavra selecionada.

Também podemos aplicar formatações de parágrafo através da janela a seguir.

### **Formatar Parágrafo**

#### **Guia Página Inicial, Grupo Estilo**

No Grupo Estilo, encontramos botões que alteram, em um único clique, as formatações de tamanho, cor e tipo de fonte, além de formatações de parágrafo do texto selecionado. Como vimos até agora, para aplicar uma formatação em um texto, temos que selecioná-lo e clicar nos itens de formatação desejados.

Por exemplo: se desejamos que uma palavra tenha a seguinte formatação “palavra a ser observada” , temos que selecioná-la e clicar uma vez no negrito, uma vez no itálico e uma vez no sublinhado.

Se essa formatação tiver que ser usada em várias palavras de um documento, podemos criar um estilo contendo todas as formatações usadas e salvá-lo com um nome. Dessa forma, da próxima vez que precisarmos usar este grupo de formatações, podemos substituir os três cliques anteriores (negrito, itálico e sublinhado) por apenas um clique no nome que demos ao nosso estilo.

- 12) Espaçamento de Linha e Parágrafo: Altera o espaçamento entre linhas de texto. Também podemos personalizar a quantidade de espaço adicionado antes e depois dos parágrafos.
- 13) Sombreamento: Permite colorir o plano de fundo atrás do texto ou do parágrafo selecionado.
- 14) Bordas: Permite inserir bordas diferenciadas no texto ou palavra selecionada.

Também podemos aplicar formatações de parágrafo através da janela a seguir.

### **Formatar Parágrafo**

#### **Guia Página Inicial, Grupo Estilo**

No Grupo Estilo, encontramos botões que alteram, em um único clique, as formatações de tamanho, cor e tipo de fonte, além de formatações de parágrafo do texto selecionado. Como vimos até agora,

para aplicar uma formatação em um texto, temos que selecioná-lo e clicar nos itens de formatação desejados.

Por exemplo: se desejamos que uma palavra tenha a seguinte formatação “palavra a ser observada” , temos que selecioná-la e clicar uma vez no negrito, uma vez no itálico e uma vez no sublinhado.

Se essa formatação tiver que ser usada em várias palavras de um documento, podemos criar um estilo contendo todas as formatações usadas e salvá-lo com um nome. Dessa forma, da próxima vez que precisarmos usar este grupo de formatações, podemos substituir os três cliques anteriores (negrito, itálico e sublinhado) por apenas um clique no nome que demos ao nosso estilo.

O Word já traz vários estilos prontos como o Normal, o Sem Espaço, o Título 1, o Título 2, entre outros.

## Inserção e Manipulação de Tabelas

### Guia Inserir, Grupo Tabelas

As tabelas são estruturas importantes para representar dados e organizá-los de forma que facilite sua interpretação e entendimento.

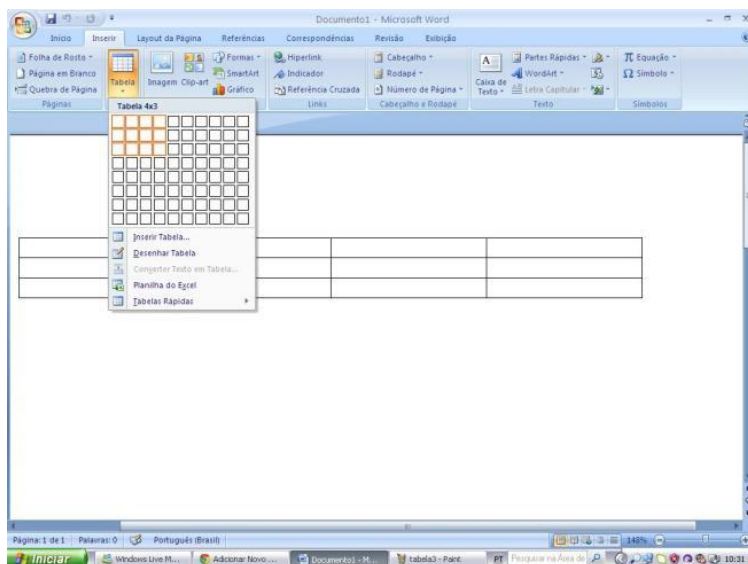
No Word, clicando no botão de comando Tabela, podemos utilizar as seguintes formas para inserir uma tabela no documento:

### Formas de Inserir Tabela

1) Selecionando os quadradinhos na horizontal, indicamos a quantidade de colunas que nossa tabela terá. Selecionando os quadradinhos na vertical, indicamos a quantidade de linhas da tabela. Ela será aplicada automaticamente ao documento.

2) Inserir Tabela:

Inserir Tabela



Pela janela da figura anterior, podemos digitar a quantidade de linhas e colunas que queremos em uma tabela. Podemos também, determinar a largura da coluna, a forma de ajuste da tabela em relação ao conteúdo e à janela. Após realizar estas escolhas e clicar no botão “OK” a tabela será criada no documento. 3) Desenhar Tabela: quando clicamos neste botão, o ponteiro do mouse se transforma no desenho de um lápis. Com ele criamos as bordas internas e externas da nossa tabela como se estivéssemos criando uma auto forma. Fica à nossa disposição a Guia Ferramentas de Tabela que podemos usar para formatar nossa tabela.

### Guia Ferramentas de Tabela

Com esta guia podemos mostrar/ocultar linha de cabeçalho, primeira coluna, linha de totais, última coluna, linhas em tiras, colunas em tiras. Além disso, podemos escolher um estilo pré-definido de tabela ou formatar o sombreamento e as bordas da mesma. Mesmo com a tabela pronta, podemos desenhar colunas ou linhas adicionais, formatar suas linhas ou apagar colunas, linhas e células.

### **Inserção e Quebra de Páginas e Colunas**

Uma quebra consiste na interrupção da formatação que estava sendo utilizada no documento até o ponto da quebra para iniciar uma nova formatação em outro ponto do documento. Podemos utilizar este recurso em páginas e colunas.

O recurso da quebra pode ser localizado na Guia Inserir, Grupo Páginas.

Cabeçalho e rodapé

Botões de Comando do Grupo Cabeçalho e Rodapé

Esta opção, presente na Guia Inserir, no Grupo Cabeçalho e Rodapé, nos oferece meios de mostrar uma área da página definida para o cabeçalho de um documento e para seu rodapé.

O cabeçalho é a parte superior da página, na qual podemos inserir itens como logotipo de empresas, símbolos, número de página e outros elementos.

O rodapé é a parte inferior da página, onde podemos inserir itens como os do cabeçalho ou informações de endereço e contato de uma empresa, por exemplo.

As opções de cabeçalho e rodapé se encontram no menu exibir nas versões do Word 2003, por exemplo. Na versão do Word 2007, 2010 e no BrOffice.org Writer, ficam no Inserir. Como exemplo, tomaremos a versão 2010 do Microsoft Word, para mostrar os comandos encontrados no Grupo cabeçalho e rodapé.

Cabeçalho Na guia inserir, encontraremos o grupo Cabeçalho e Rodapé.

Neste grupo, teremos os comandos Cabeçalho, Rodapé e Número de Página. Clicando no botão de comando Cabeçalho, podemos optar por deixar o cabeçalho da página:

- Em branco: que limpa possíveis formatações ou inserções anteriores.
- Em três colunas: deixa o cabeçalho em branco, mas com formatação pré-definida para inserirmos dados nos alinhamentos direito, centralizado e esquerdo.
- Alfabeto: que dispõe o título do documento centralizado sobre borda de linha dupla.
- Animação: deixa o título do capítulo com número de página em caixa de ênfase. “Ideal para documento com layout de livro”.
- Editar cabeçalho: mostra um local com borda tracejada, destinado ao cabeçalho para que este possa ser alterado na página. Destaca o local do cabeçalho para que o usuário possa trabalhar esta área do documento.
- Remover cabeçalho: exclui os dados inseridos no cabeçalho de um documento.

No rodapé são encontrados os mesmos comandos, mas estes são aplicados na parte inferior da página de um documento.

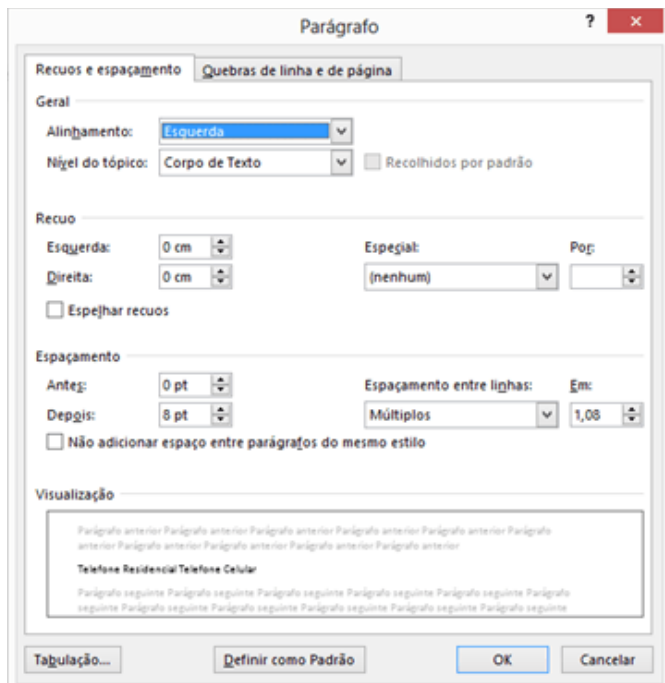
Configuração da página e do parágrafo Na Guia Layout da Página, temos, entre outros, o Grupo Configurar Página.

Através dele é possível alterar o tamanho das margens esquerda, direita, inferior e superior. A orientação do papel, para retrato ou paisagem também é um item que alteramos através deste grupo bem como o tipo do papel e o layout da página.



Na mesma Guia, encontramos o Grupo Parágrafo que traz as opções de recuar à esquerda ou à direita, aumentar ou diminuir o espaçamento antes e depois do parágrafo e também o acesso à janela Parágrafo:

### Parágrafo



A janela da figura acima nos permite configurar o alinhamento do parágrafo, os recuos, os espaçamentos e obter uma visualização prévia da formatação que estamos aplicando ao parágrafo.

Modos de exibição de documento e zoom Os modos de exibição consistem na forma que o documento será mostrado na tela. Suas opções de configuração estão disponíveis na Guia Exibição. Esta Guia possui os seguintes Grupos:

Modos de Exibição de Documento: alteram a forma que o documento está sendo mostrado na tela, nos oferecendo as opções Layout de Impressão, Leitura em Tela Inteira, Layout da Web, Estrutura de Tópicos e Rascunho.

Mostrar/Ocultar: permite mostrar ou ocultar régua, linhas de grade, barra de mensagens, mapa do documento, miniatura.

Zoom: permite configurar a aparência de proximidade do documento na tela, alterando a porcentagem dessa proximidade, mostrando o documento página por página, duas páginas ou com a largura da página ocupando a tela.

### Ortografia e Gramática

#### Botão de Comando Ortografia e Gramática

Faz a correção ortográfica e gramatical do documento. Encontramos este recurso na Guia Revisão, no Grupo Revisão de Texto. Assim que clicamos na opção “Ortografia e gramática”, a seguinte tela será aberta:

A verificação ortográfica e gramatical do Word, já busca trechos do texto ou palavras que não se enquadrem no perfil de seus dicionários ou regras gramaticais e ortográficas. Na parte de cima da janela “Verificar ortografia e gramática”, aparecerá o trecho do texto ou palavra considerada inadequada.

Em baixo, aparecerão as sugestões. Caso esteja correto e a sugestão do Word não se aplique, podemos clicar em “Ignorar uma vez”; caso a regra apresentada esteja incorreta ou não se aplique ao trecho do texto selecionado, podemos clicar em “Ignorar regra”; caso a sugestão do Word seja ade-

quada, clicamos em “Alterar” e podemos continuar a verificação de ortografia e gramática clicando no botão “Próxima sentença”.

Se tivermos uma palavra sublinhada em vermelho, indicando que o Word a considera incorreta, podemos apenas clicar com o botão direito do mouse sobre ela e verificar se uma das sugestões propostas se enquadra.

Por exemplo, a palavra informática. Se clicarmos com o botão direito do mouse sobre ela, um menu suspenso nos será mostrado, nos dando a opção de escolher a palavra informática. Clicando sobre ela, a palavra do texto será substituída e o texto ficará correto.

Criação, gravação, abertura e impressão de documentos

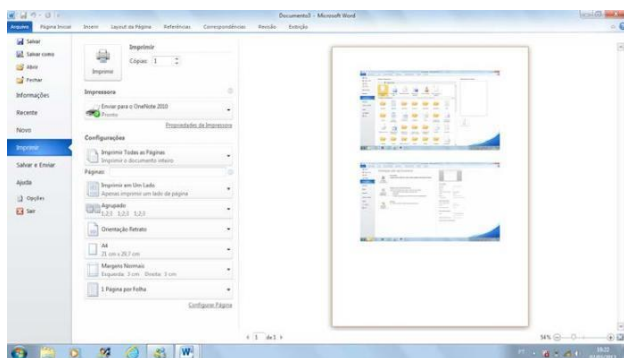
### **Guia Arquivo**

Para criar novos documentos, gravar alterações, abrir documentos existentes e imprimir, recorreremos à Guia Arquivo e ao seu menu, conforme mostrado na figura acima.

Quando clicamos no menu Novo, a seguinte tela será exibida:

### **Tela do Menu Novo**

Nesta tela, podemos escolher iniciar um documento em branco ou usar algum dos modelos disponíveis.



Após selecionar o modelo desejado, basta clicar no botão

“Criar”.

Para gravar o documento alterado ou criado, podemos usar o ícone Salvar, presente na Barra de Ferramentas de Acesso Rápido, ou clicar na Guia Arquivo e no botão Salvar.

Vale esclarecer a diferença entre o botão Salvar e o botão Salvar como.

O botão Salvar sobrepõe as alterações realizadas no mesmo documento. Então, se alteramos um documento e salvamos, perdemos o documento anterior e ficamos apenas com o que alteramos. Se usarmos a opção Salvar como, podemos escolher outro nome e outro lugar para gravar o arquivo, mantendo, se desejarmos o documento original.

Quando clicamos no botão Salvar a primeira vez, será aberta a mesma janela do Salvar como, onde podemos escolher o nome, local e tipo de arquivo que gravaremos o nosso documento. Mas se após salvarmos a primeira vez, clicarmos no mesmo botão, ele só salvará as alterações em cima do mesmo documento e no mesmo local.

Para abrir um documento já existente, podemos clicar no ícone da Guia Arquivo. Essa opção nos mostrará uma janela onde podemos localizar o arquivo no local onde está gravado e exibí-lo na tela.

Para imprimir um arquivo, após abri-lo, clicamos na Guia Arquivo e na opção Imprimir. Será exibida na tela, entre outros, os itens a seguir:

## Imprimir Documentos

Podemos escolher a quantidade de cópias, a impressora para qual enviaremos a impressão, imprimir o documento inteiro ou apenas algumas de suas páginas, imprimir apenas um lado da página, como as páginas serão agrupadas na saída da impressora, a orientação do papel, o tipo do papel, a configuração das margens e se desejamos 1 página por folha ou mais.

teclas de atalho: Para complementar nossos estudos sobre o Word, vamos ver uma lista com diversas teclas de atalho, que substituem funções de cliques do mouse, agilizando nosso trabalho. A lista que vamos estudar pode ser usada em vários outros programas, mas vale a ressalva de que nem sempre as teclas de atalho que desempenham uma função em um programa, fazem o correspondente em outro.

CTRL+C: copia a palavra ou o texto selecionado. CTRL+V: cola o que está na área de transferência, no local onde estiver o ponto de inserção.

CTRL+X: recorta a palavra ou o texto selecionado. CTRL+N: formata o texto selecionado para o negrito. CTRL+I: formata o texto selecionado para o itálico. CTRL+S: sublinha o texto selecionado. CTRL+Z: desfaz a última ação. CTRL+Y: refaz a última ação desfeita. CTRL+=: deixa o texto selecionado em subscripto. CTRL+Shift++: deixa o texto selecionado em sobrescrito. CTRL+Shift+P: abre a janela para formatação de fonte, com o tamanho da fonte selecionado para alteração.

CTRL+Shift+F: abre a janela para formatação de fonte, com o tipo de fonte selecionado para alteração.

CTRL+Shift+C: aciona a ferramenta "Formatar Pincel". CTRL+>: aumenta o tamanho da fonte. CTRL+<: diminui o tamanho da fonte. CTRL+Q: alinha o texto à esquerda. CTRL+E: centraliza o alinhamento do texto. CTRL+J: justifica o alinhamento do texto. CTRL+\*: aciona a função do botão "Mostrar tudo". CTRL+L: abre a janela do "Localizar". CTRL+U: abre a janela do "Substituir". CTRL+K: abre a janela "Inserir Hiperlink".

F7: abre a janela do "Verificar Ortografia e Gramática". Alt+Clique: abre o painel de tarefas do "Pesquisar". Shift+F7: abre a janela do dicionário de sinônimos. CTRL+A: abre a janela para abrir um documento existente. CTRL+B: salva o documento em edição. CTRL+P: imprime o documento.

Para encerrar, vale ressaltar que muitas das informações que vimos em nossos estudos sobre o Word, foram retiradas do próprio programa, da sua ajuda ou dos popups que o próprio programa oferece quando pausamos o mouse sobre seus recursos

## Microsoft Office 2013

O Microsoft Office é uma suíte de aplicativos para escritório que contém vários programas já conhecidos de outras versões desta suíte. Os mais comuns são o processador de textos (Word), de planilhas eletrônicas (Excel), de apresentações (PowerPoint), de bancos de dados (Access), e o cliente de e-mails (Outlook), dentre outros.

O pacote Microsoft Office 2013 passa a ser oferecido como uma combinação de serviço na nuvem com aplicativos locais. São quatro edições disponíveis: Home Premium, Home & Student, Home & Business e Professional

O Office 2013 permite que usuários alternem entre vários dispositivos e encontre o mesmo ambiente de trabalho em todos eles. A suíte Office 2013 traz aplicativos para serem baixados na nuvem, com o pagamento de uma assinatura. Na configuração padrão, os documentos podem ser armazenados no SkyDrive, o serviço de armazenamento na nuvem da Microsoft. Assim, estão disponíveis em qualquer lugar onde haja acesso à internet.

Uma cópia do documento armazenada localmente permite o acesso quando não houver conexão.

O Office teve sua interface refeita para responder melhor a comandos por toque. Ficou mais fácil mover objetos e acionar o zoom com os dedos, como em outros apps para tablets. Trazidos pelas inovações do Windows 8 e pela popularização dos tablets, os recursos de tela sensível, no modo de leitura do Word, o Touch Mode aumenta os botões e permite que o arquivo seja manipulado com

gestos sobre a tela. Movimentos dos dedos como a pinça ou o zoom também funcionam em outros aplicativos do pacote. É possível passar páginas de um documento ou transitar em slides de apresentações do PowerPoint, por exemplo.

Quanto à interface, o Office 2013 segue o Windows 8 na sua interface. A identidade visual é a Modern. Telas de apresentação oferecem opções como modelos definidos, conexão ao SkyDrive ou arquivos em branco. Cada programa tem uma cor. O Word, azul, o Excel, por exemplo, tem detalhes em verde, o PowerPoint, em vermelho.

### **Microsoft Office 2013**

Na nova versão, os aplicativos possuem uma tela inicial (landing page), apresenta modelos e outras opções para criar ou abrir documentos, totalmente integrada ao Windows 8. Abaixo, a tela de entrada do Word: Uma novidade no Excel, por exemplo, e que quando se seleciona um grupo de dados na planilha, o aplicativo pode sugerir o tipo de gráfico mais adequado para representá-lo.

Ao criar uma tabela dinâmica, em vez de ficar ajustando configurações, o usuário escolhe entre vários layouts propostos pelo Excel. Dos novos recursos, o Flash Fill identifica um padrão no preenchimento de uma linha e sugere o complemento da linha seguindo este padrão.

O Word passa a contar com um modo de leitura em que a tela adquire um aspecto mais limpo. O novo modo de leitura (Read Mode) que oculta a barra de ferramentas (Ribbon) e exhibe os documentos como se fossem um livro impresso. Nesse modo não é possível editar o documento, mas é possível usar ferramentas de busca. No caso de textos longos, o Word marca a página onde a leitura foi interrompida. E há, ainda, um recurso de zoom para gráficos e fotos. Ao tocar num objeto desse tipo, uma versão ampliada é exibida. Clicando novamente, a imagem volta ao tamanho original. Quando se desloca ou redimensiona uma foto, tabela ou gráfico, o texto se distribui automaticamente pelo espaço restante na página.

O Power Point, editor de apresentações do Office, ganhou um modo do apresentador, em que o palestrante enxerga o próximo slide a ser exibido e pode fazer anotações sem que a plateia veja. Também há uma função de zoom que pode ser usada para destacar uma parte do slide durante a apresentação. Temas novos, mais bonitos e widescreen. Há novas guias dinâmicas para facilitar o alinhamento de imagens e objetos e um conta-gotas para capturar cores e aplicá-las em elementos e textos. O Outlook, o aplicativo de e-mail do Office, traz um gerenciador de contatos People Card, integrado a redes sociais.

Ele trabalha integrado ao Facebook e ao LinkedIn.

Também ficou mais fácil consultar o calendário e há uma área que exhibe a previsão do tempo no aplicativo.

O Office 2013 suporta anotações feitas com caneta na tela sensível ao toque. Anotações manuscritas podem ser convertidas em texto digital. Dependendo do aplicativo, a caneta também pode ser usada para apagar, desenhar e colorir. Nas apresentações com o PowerPoint, ela funciona como uma espécie de apontador a laser virtual.

Quanto aos elementos das telas dos aplicativos, o Microsoft Office 2013, assim como na versão 2010, traz as guias (muitas vezes ainda chamadas de menus, em provas de concursos), os grupos, a barra de acesso rápido e a Faixa de Opções (que reúne os grupos e seus botões).

Uma diferença visível entre o Microsoft Office 2010 e o 2013, é que na nova versão, a guia Arquivo, não abre mais como um menu drop down, mas como uma tela inteira.

A Barra de Acesso Rápido, aparece em todos os aplicativos, com os botões Salvar, Desfazer, Refazer (que alterna com o botão Repetir), o Botão do Menu de Controle, sempre mostrando o símbolo do programa que está sendo utilizado no momento: no exemplo abaixo, o Word, aparece no canto esquerdo desta barra. Com ele é possível acessar as velhas opções como (Restaurar, Mover, Tamanho, Minimizar, Maximizar e Fechar).

A Barra de Acesso Rápido.

Uma novidade no Word 2013 é a inclusão da guia Design.

E os nomes das guias na versão 2013 aparecem em caixa alta.

O Word é o processador de textos da Microsoft, sendo um dos programas mais utilizados para este fim. Com ele, é possível criar vários tipos de documentos, sendo uma ferramenta repleta de recursos e funcionalidades, sendo apta à elaboração dos mais diversos tipos de documentos, incluindo mala direta, páginas da Web, com a inclusão de tabelas e objetos multimídia.

O Word também possui guias contextuais, ou seja, relacionada ao objeto selecionado, como por exemplo, ao selecionar uma imagem, ele cria acima da barra de guias, uma guia com opções direcionadas à manipulação do objeto selecionado (Ferramentas de imagem).

### Correio Eletrônico

Um correio eletrônico ou e-mail é um método que permite compor, enviar e receber mensagens através de sistemas eletrônicos de comunicação. O termo e-mail é aplicado tanto aos sistemas que utilizam a Internet e são baseados no protocolo SMTP, como aqueles sistemas conhecidos como intranets, que permitem a troca de mensagens dentro de uma empresa ou organização e são, normalmente, baseados em protocolos proprietários.

### Tecnologia



E-mail sendo composto num programa gráfico de e-mail. (E-mail escrito em Alemão.)

### Sistema De E-Mail

O envio e recebimento de uma mensagem de e-mail é realizada através de um sistema de correio eletrônico. Um sistema de correio eletrônico é composto de programas de computador que suportam a funcionalidade de cliente de e-mail e de um ou mais servidores de e-mail que, através de um endereço de correio eletrônico, conseguem transferir uma mensagem de um usuário para outro. Estes sistemas utilizam protocolos de Internet que permitem o tráfego de mensagens de um remetente para um ou mais destinatários que possuem computadores conectados à Internet.

### Características do E-mail

O formato na Internet para mensagens de e-mail é definido na RFC 2822 e uma série de outras RFCs (RFC 2045 até a RFC 2049) que são conhecidas como MIME.

Mensagens de e-Mail consistem basicamente de duas seções principais:

Cabeçalho (header) — é estruturado em campos que contém o remetente, destinatário e outras informações sobre a mensagem.

Corpo (body) — contém o texto da mensagem.

O corpo é separado do cabeçalho por uma linha em branco.

### **Funcionalidades**

Hoje os grandes sítios da Internet criaram uma série de facilidades para o usuário. Note que essa variação é só uma facilidade e não um novo tipo de e-mail. Entre estas podemos citar:

#### **E-Mail Restrito**

Alguns sítios restringem alguns tipos de e-mail. Esse tipo de restrição normalmente é usado a fim de evitar a atuação de um spammer ou divulgador não autorizado de mensagens em massa. Normalmente esse tipo de mensagem eletrônica é mais usado em empresas.

#### **E-mail com Privacidade Segura**

Normalmente usado por autoridades e seu uso é controlado. Por medida de segurança alguns organismos e entidades internacionais ou mesmo ligados a Governos, categorizam o e-mail como:

Privativo ou de uso exclusivo da autoridade: Esse e-mail, apesar de ter acesso a rede é tão restrito que a própria autoridade deve configurá-lo de quem recebe as mensagens;

Semi-privativo: O mesmo que privativo, porém menos restrito.

Os norte-americanos chegam ao cúmulo de dar níveis e subníveis a esse tipo de mensagem;

Entretanto, vêm crescendo o uso da criação de chaves criptográficas pessoais (facilidade provida por aplicativos especializados), assegurando a privacidade das informações “de qualquer importância” de cada indivíduo. Tais chaves possuem uma grande flexibilidade, escalabilidade e confiabilidade.

Aqui vão algumas dicas de segurança: Nunca abrir ou responder e-mails desconhecidos; nunca abrir arquivos (ficheiros) de e-mails desconhecidos, pois podem conter vírus; e ter sempre um anti-spyware (contra os programas-espiões) e antivírus instalados no seu computador.

#### **E-mail Categorizado ou Especial**

Especial ou categorizado em níveis, que são de uso exclusivo dos provedores de Internet. Servem para testes e verificar se funciona ou não o seu sistema anti-spam (contra as mensagens eletrônicas em massa).

#### **E-mails Gratuitos e WebMail**

Com a popularização da Internet através dos provedores gratuitos (cujos usuários ganhavam também uma caixa de correio eletrônico grátis), muitos sítios começaram a oferecer endereços de e-mail gratuitos desvinculados de qualquer outro serviço. Essas mensagens de e-mail podem ser lidas com o uso do próprio navegador, sem a necessidade de um programa específico, sendo por isso também chamados webmail.

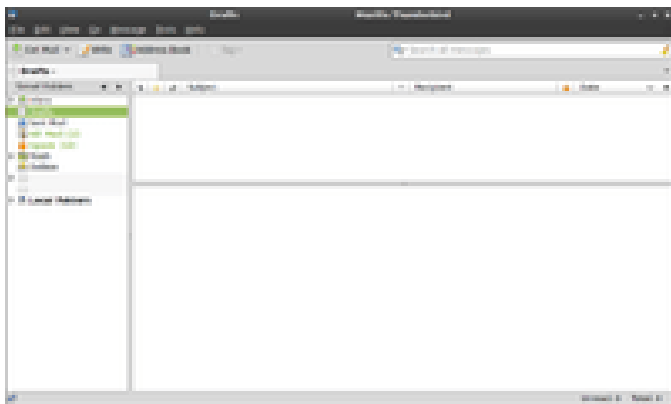
#### **Popularidade**

O correio eletrônico se tornou tão popular devido a sua grande facilidade em quebrar barreiras geográficas. Pessoas que estão em diferentes continentes podem se comunicar, desde que possuam computadores ou qualquer outro dispositivo com tal funcionalidade conectados a Internet, eles podem enviar e receber mensagens a qualquer hora do dia e para qualquer parte do mundo.

Observa-se que o correio eletrônico deixa de ser apenas um meio de troca de mensagens entre pessoas para se tornar um grande fator na produtividade das empresas. Grandes empresas estão cada vez mais usando o correio eletrônico para desempenhar papéis decisivos em suas negociações.

A Intranet pode ser usada para tornar a comunicação de funcionários com outros grupos tornando assim mais fácil o trabalho e eliminando mensagens em massa e outras mensagens indesejadas.

### Áreas de Aplicações



A Interface de um Cliente de e-mail, Thunderbird.

As aplicações de correio eletrônico normalmente oferecem ao usuário uma série de facilidades. A maior parte delas fornece um editor de textos embutido e a possibilidade do envio de arquivos anexados a correspondência. Além disso, a maioria das aplicações permite o envio de correspondências para um único destinatário ou o envio para mais de uma pessoa ou para um grupo de pessoas.

Embora não tenha sido desenvolvida como uma ferramenta de trabalho cooperativo, os serviços de correio eletrônico adaptaram-se muito bem ao ambiente de grupos de trabalho onde se tornaram indispensáveis nas organizações, agilizando processos, democratizando o acesso as informações e diminuindo os custos. Esta é uma das formas mais usadas para o estabelecimento de comunicações por meio do computador.

Muitas organizações também usam o correio eletrônico como forma de troca de mensagens, mas se quiserem usar recursos de groupware poderão incluí-los de forma simples e com baixo custo, com uma boa segurança.

### Terminologia Usada

auto-responders (resposta automática) — O software do receptor responde automaticamente após receber a mensagem.

bounce backs (Retorno – sem atingir seu destino) — O e-mail enviado de volta ao servidor que originou a mensagem sem atingir seu destino final.

bounce rate” (índice de retorno) — Índice de e-mails retornados sem atingir seu destino final.

bulk, bulking (“baciada”) — Sinônimo de SPAM, utilizado principalmente pelos spammers.

call to action (chamada para ação) — Palavras que incentivam uma ação do receptor.

click-through — A ação de clicar em um link.

click-through rate (CTR) — Índice de click trough dos e-mails enviados.

commercial e-mail (e-mail comercial) — E-mail enviado com finalidade comercial.

demographic — Características de um grupo alvo para recebimento de e-mails.

double opt-in (opt-in duplo) — O receptor reitera seu desejo de recebimento de e-mails de uma determinada fonte. A primeira, inserindo seu e-mail em algum campo do site. Após isso, receberá um e-mail de confirmação, pedindo o envio de seu cadastro completo, que deverá ser fornecido antes de receber seus e-mails. Outra possibilidade é a adoção de uma chave de confirmação, permitindo veri-

ficar se o endereço existe e se o cadastrado é de fato o proprietário. Também chamado de confirmed subscription (assinatura confirmada) ou closed-loop opt-in (opt-in fechado).

double opt-out (opt-out duplo) — O mesmo procedimento do opt-in, mas para o opt-out. Geralmente utilizado por spammers que procuram dificultar o cancelamento da assinatura de suas listas. Alguns spammers mal intencionados utilizam a manifestação de opt-out do receptor como um forma de confirmar a existência de seu endereço de e-mail.

express consent (consentimento expresso) — O receptor concorda ativamente em receber e-mails selecionando uma opção em um formulário na web ou qualquer outra forma. Se por exemplo essa opção já estiver selecionada e o receptor não desativar a seleção, esse consentimento não é expresso.

false positives (positivo falso) — E-mails identificados como spam pelo filtro do receptor quando de fato não o são.

format (formatos) — E-mails podem ser enviados em texto, HTML, ou rich text format.

hard bounce — E-mail retornado por nunca ter atingido seu destino porque o endereço de e-mail não existe.

list broker (revendedor de listas) — Revendedor de listas de endereços de e-mails.

list building (construção de listas) — Processo de geração de listas de endereços de e-mails usados por campanhas de e-mails.

list host (hospedeiro de listas) — Serviço que proporciona ferramentas para armazenar grandes listas de e-mail, bem como o disparo de grande quantidade de e-mails.

list manager (administrador de listas) — Dono, operador, ou software, responsável por administrar listas de endereços de e-mails.

look and feel — Sensação causada pela aparência, layout, design, funcionalidade e qualquer outra coisa não diretamente relacionada ao conteúdo do e-mail.

open rate (índice de visualização) — Índice criado a partir da quantidade de e-mails abertos em relação aos e-mails enviados. O índice mais utilizado é: e-mails entregues (enviados – retornados) / aberturas únicas.

opt-in — A ação de concordar em receber e-mails de uma determinada fonte cadastrando-se em uma lista de e-mail.

opt-out — A ação de descadastramento de uma determinada lista de e-mails.

personalization (personalização) — O uso de tecnologia combinado com as informações disponíveis dos clientes permite customizar a relação entre o remetente e o receptor.

rental list (lista alugada) — Lista de e-mails que é alugada por tempo ou ação determinada.

segmentation (segmentação) — Utilização de informações previamente coletadas para direcionar a mensagem a segmentos específicos da lista.

soft bounce — O e-mail chega até o servidor do receptor, mas retorna antes de chegar ao receptor. Isto pode ocorrer devido ao fato de a caixa de entrada de mensagens estar cheia.

spam or UCE (unsolicited Commercial e-mail-UCE) — E-mail encaminhado sem o consentimento do receptor.

spam filter — Software utilizado para filtrar e-mails, evitando ou anunciando a presença de spam.

subject line (assunto) — Campo destinado a dizer qual a finalidade da correspondência.

tracking (acompanhamento) — Monitoramento de CTR, índice de abertura, retornos etc.



trigger based messaging (mensagens de disparo programado) — Envio de mensagem condicionado a um outro evento ou a uma outra mensagem. Geralmente utilizado para o fornecimento de informação adicional.

unique click (clique único) — Durante um determinado período, um receptor pode vir a clicar diversas vezes em um mesmo link. Ainda assim será considerado como clique único.

### **Problemas**

A desvantagem está na falta de conhecimento da grande maioria dos internautas e, ainda, os spammers ou geradores de spam, grandes remetentes de vírus. Como podemos ver em seguida:

Spam – mensagens de e-mail não desejadas e enviadas em massa para múltiplas pessoas por um spammer, agente difundidor dessas mensagens, que normalmente possui propagandas indesejadas, códigos maliciosos e vírus diversos;

Vírus – As mensagens de e-mail são um excelente veículo de propagação de vírus, sobretudo através dos ficheiros (arquivos) anexos. Por isso recomenda-se nunca baixar um ficheiro (arquivo) tipo .exe ( executáveis) ou outros suspeitos;

É aconselhável nunca abrir e-mail desconhecido, exceto se for de um site confiável, não sem antes observar os procedimentos de segurança.

### **Fraudes**

Com o grande aumento do uso da Internet e do correio eletrônico na vida das pessoas, tornou-se grande o número de pessoas maliciosas que tentam utilizar esses meios para realizar fraudes. O grande foco desses fraudadores são pessoas que utilizam sítios de instituições financeiras na Internet. Os fraudadores eletrônicos utilizam a grande facilidade com que uma caixa de correio pode ser forjada e falsificada. Eles utilizam listas e programas para envio de spam em grande escala juntamente com arquivos executáveis e serviços de hospedagem gratuitos e que não necessitem de identificação legítima.

Esses fraudadores enviam mensagens de e-mail se passando por bancos e outras instituições financeiras, solicitando dados pessoais, número de conta corrente, cartão bancário e, às vezes, até mesmo o número de senhas de clientes. Esses clientes desavisados enviam esses dados pensando se tratar realmente de um pedido dessas instituições, sem saberem que estão a se tornar vítimas de fraudadores. Cada vez mais cresce o número de pessoas que tem suas contas fraudadas, compras através de seus cartões e outros tipos de fraudes. A falta de legislação e meios de segurança que controlem esse tipo de ação tem se tornado um fator positivo para que esses fraudadores continuem a atuar. Além disso não há nenhum mecanismo que permita rastrear, identificar e coibir a ação desses fraudadores tornando assim cada vez mais difícil a atuação das autoridades nesses casos. Mensagens de e-mail indesejadas de instituições que queiram solicitar dados pessoais devem ser ignoradas, pois essas não enviam tais mensagens para seus clientes.

A melhor maneira de se prevenir contra fraudes ao utilizar o correio eletrônico é mesmo procurar o máximo de informações sobre sua origem e desconfiar de qualquer indício que possa levantar alguma suspeita. Mensagens de e-mail que foram enviadas por pessoas ou empresas desconhecidas encabeçam essa lista. Deve-se ter uma atenção especial com estes tipos de mensagem, pois podem instalar programas-espiões maliciosos, que podem capturar dados que estejam ou foram digitados no computador em que tais programas sejam executados, tornando assim fácil a obtenção de dados de seus usuários.

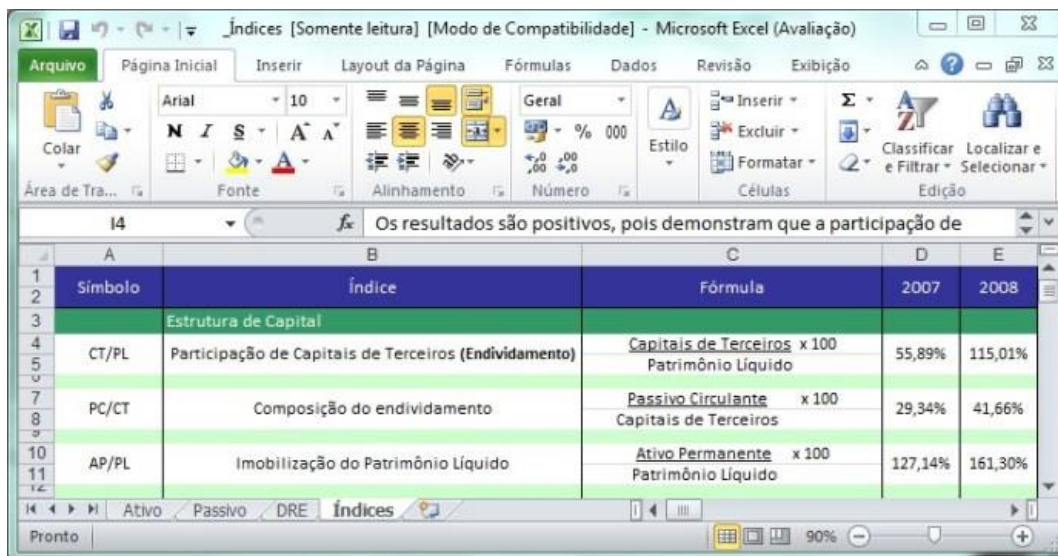
### **Noções dos Ambientes Microsoft Office e BROffice**

Microsoft Office:

A versão do pacote Office 2010 reúne os aplicativos mais conhecidos da empresa como Word, Excel, Power Point, Access, Outlook e Publisher.

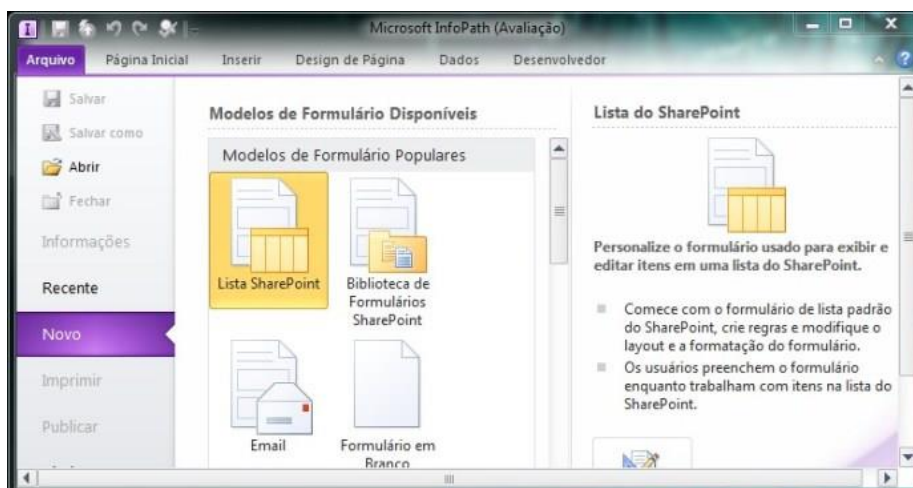
### **Enfoque nos Aspectos Visuais**

A aposta da empresa, mais uma vez, recai nos elementos visuais de fácil identificação por parte do usuário. A ideia é trazer novas ferramentas que transformem a concepção do seu trabalho em uma experiência dinâmica com cores e elementos visuais. Além disso, um pacote de temas e SmartArt layouts gráficos dá uma ideia a você de como interagir com as novas opções.



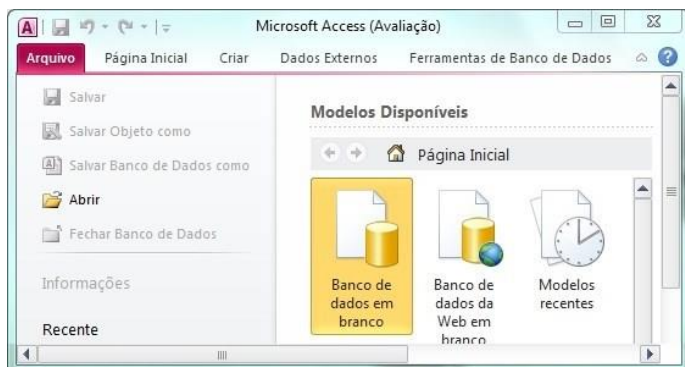
### Trabalho em Conjunto

Se o GoogleDocs conquistou muitos usuários graças à sua plataforma online que permite trabalhar de forma colaborativa, a Microsoft também passa a integrar algo do gênero em seu pacote de aplicativos. No Microsoft Word, Microsoft Excel e Microsoft PowerPoint, graças ao novo conceito de Web App, agora é possível trabalhar de maneira online e em tempo real na edição de documentos.



### Mobilidade

O Office 2010 aposta também na mobilidade como diferencial para interação com o usuário. A proposta é que seja possível trabalhar a partir de um smartphone ou até virtualmente. Para isso, basta ao salvar o arquivo no seu desktop enviá-lo também para o live space. Ao acessá-lo virtualmente você pode editar o texto como desejar e, ao voltar para o seu desktop, automaticamente a versão mais recente é aberta, caso você esteja conectado à internet.



**Personalização de Vídeos no PowerPoint**

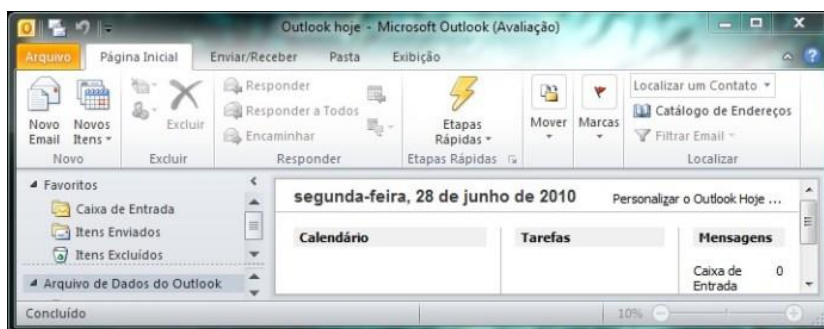
Sim, agora é possível editar trechos e incluir alguns efeitos simples em vídeos dentro do PowerPoint. Além disso, a edição de dados e gráficos ficou ainda mais fácil, uma vez que o programa adota o trabalho por layers (camadas), similar ao de editores de imagens como o Adobe Photoshop.



**Compressão de E-mails em uma Única Categoria**

Esta novidade é do Outlook, mas você já deve conhecer algo similar se possui uma conta do Gmail. Suas trocas de e-mails agora passam a ser agrupadas em um único tópico. Um exemplo: suponha que em uma conversa com um amigo você troquem dez mensagens entre si.

Todas são listadas em um único tópico e organizadas da mais nova para a mais antiga. Isso evita que sua caixa de mensagens seja poluída por dezenas de confirmações de leitura ou respostas simples que caberiam em uma caixa de conversação.

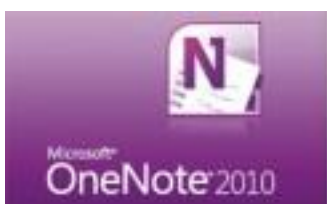


Com o Microsoft Access é possível desenvolver desde aplicações simples como por exemplo, um cadastro de clientes, controle de pedidos até aplicações mais complexas, como por exemplo, todo o controle operacional, administrativo e financeiro de uma pequena ou até mesmo de uma média ou grande empresa, pois os aplicativos desenvolvidos podem rodar perfeitamente numa rede de computadores e os dados armazenados pelo sistema podem ser publicados na Intranet ou até mesmo na Internet.



O Microsoft Office Excel é um editor de planilhas produzido pela Microsoft para computadores que utilizam o sistema operacional Microsoft Windows, além de computadores Macintosh da Apple Inc. e dispositivos móveis como o Windows Phone, Android ou o iOS. Seus recursos incluem uma interface intuitiva e capacitadas ferramentas de cálculo e de construção de gráficos que, juntamente com marketing agressivo, tornaram o Excel um dos mais populares aplicativos de computador até hoje. É, com grande vantagem, o aplicativo de planilha eletrônica dominante, disponível para essas plataformas e o tem sido desde a versão 5 em 1993 e sua inclusão como parte do Microsoft Office.

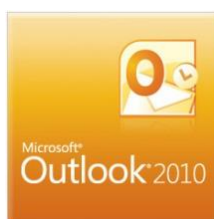
Microsoft InfoPath (Microsoft Office InfoPath) é um aplicativo da Microsoft utilizado para desenvolver dados no formato XML. Ele padroniza os vários tipos de formulários, o que ajuda a reduzir os custos do desenvolvimento personalizado de cada empresa. O programa entrou em ação no pacote do Office 2003 foi incluído também no Office 2007 e Office 2010. Não está disponível na versão Microsoft Works 6.0.



O Microsoft Office OneNote, habitualmente referido como Microsoft OneNote, é uma ferramenta para anotações, coleta de informações e colaboração multi-usuário desenvolvida pela Microsoft.

Embora muitos sistemas anteriores tenham se baseado em texto de fluxo linear (simples listas), OneNote visualiza as notas em uma página bidimensional. OneNote acrescenta também características modernas, tais como desenhos, fotos, áudio e vídeo (multimídia), bem como compartilhamento multi-usuário de notas.

Uma ferramenta muito útil para busca de anotações é a ferramenta busca que o OneNote oferece. Todas as notas são indexadas, o que significa que em um tempo muito curto o software tem a capacidade de encontrar arquivos e textos. A busca ocorre igualmente dentro dos textos de figuras e nas palavras gravadas por áudio, já que o programa possui reconhecimento de texto e fala automáticos. Uma versão web do OneNote é parte integrante do OneDrive ou Office Web Apps e possibilita aos usuários que editem notas através de um navegador de internet (browser).



Integrante do pacote Microsoft Office. Diferentemente do Outlook Express, que é usado basicamente para receber e enviar e-mail, o Microsoft Outlook além das funções de e-mail, ele é um calendário

completo, onde você pode agendar seus compromissos diários, semanais e mensais. Ele traz também um rico gerenciador de contatos, onde você pode além de cadastrar o nome e email de seus contatos, todas as informações relevantes sobre os mesmos, como endereço, telefones, Ramo de atividade, detalhes sobre emprego, Apelido, etc. Oferece também um Gerenciador de tarefas, as quais você pode organizar em forma de lista, com todos os detalhes sobre determinada atividade a ser realizada. Conta ainda com um campo de anotações, onde ele simula aqueles post-its, papéis amarelos pequenos autoadesivos. Utilizado geralmente no sistema operacional Windows.



Microsoft PowerPoint é um programa utilizado para criação/edição e exibição de apresentações gráficas, originalmente escrito para o sistema operacional Windows e portado para a plataforma Mac OS X. A versão para Windows também funciona no Linux através da camada de compatibilidade Wine. Há ainda uma versão mobile para smartphones que rodam o sistema Windows Phone.

O PowerPoint é usado em apresentações, cujo objetivo é informar sobre um determinado tema, podendo usar: imagens, sons, textos e vídeos que podem ser animados de diferentes maneiras. O PowerPoint tem suporte a objetos OLE e inclui uma ferramenta especial de formatação de texto (WordArt), modelos de apresentação pré-definidos, galeria de objetos gráficos e uma gama de efeitos de animação e composição de slides.

O formato nativo do PowerPoint é o PPT, para arquivos de apresentações, e o PPS, para apresentações diretas. A partir da versão 2007 do programa, a Microsoft introduziu o formato .PPTX. Para executar o Powerpoint em máquinas que não o tenham instalado, é necessário usar o software PowerPoint Viewer, uma vez que o PowerPoint não tem suporte nativo para outros formatos como o SWF, o PDF e mesmo o OpenDocument Format. Os arquivos do PowerPoint em geral são lidos sem problemas por outros softwares similares como o Impress.



Microsoft Publisher é um programa da suite Microsoft Office, que é basicamente usado para diagramação eletrônica, como elaboração de layouts com texto, gráficos, fotografias e outros elementos. Esse programa é comparado com softwares tais como o QuarkXPress, Scribus, Adobe InDesign e Draw. Foi criado em 1991.

É Capaz de Criar

Publicações para impressão;

Páginas da Web (que não requerem conexão com a internet ao criar uma página da web);

Edições de e-mail.

Criar panfletos

Boletins informativos



Este artigo descreve a principal funcionalidade e a arquitetura do Microsoft SharePoint Workspace 2010. SharePoint Workspace 2010 sucede e é o novo nome do Microsoft Office Groove 2007. O SharePoint Workspace 2010 é um aplicativo cliente que oferece acesso interativo rápido e a qualquer instante a bibliotecas de documentos e listas no Microsoft SharePoint Server 2010 e no Microsoft SharePoint Foundation 2010. O SharePoint Workspace 2010 também oferece opções para criar espaços de trabalho conjuntos do e espaços de trabalho da Pasta Compartilhada. O SharePoint Workspace 2010 é mais versátil que o Microsoft Office Groove 2007 e pode ser integrado ao Microsoft SharePoint Server 2010 ou ser executado de forma independente.

O Microsoft SharePoint Workspace 2010 fornece um cliente para Microsoft SharePoint Server 2010 e Microsoft SharePoint Foundation 2010, o qual habilita a sincronização em tempo real do conteúdo da área de trabalho com documentos e listas do SharePoint. O SharePoint Workspace 2010 também oferece opções para criação de espaços de trabalho de colaboração do Groove e pastas compartilhadas sincronizadas. Com o uso do SharePoint Workspace 2010, os profissionais de informações podem sincronizar facilmente conteúdo online e offline com um site designado do SharePoint ou colaborar com parceiros externos e membros da equipe externa por meio de espaços de trabalho compartilhados. O SharePoint Workspace 2010 está incluído no Microsoft Office Professional Plus 2010.



O Microsoft Word é um processador de texto produzido pela Microsoft. Foi criado por Richard Brodie para computadores IBM PC com o sistema operacional DOS em 1983. Mais tarde foram criadas versões para o Apple Macintosh (1984), SCO UNIX e Microsoft Windows (1989). Faz parte do conjunto de aplicativos Microsoft Office. Utiliza atualmente como extensão padrão dos arquivo de texto: “.docx”.

### **BrOffice**

BrOffice.org é o nome adotado no Brasil da suíte para escritório OpenOffice.org. A mudança do nome surgiu em função de um processo movido pela BWS Informática, uma microempresa de comércio de equipamentos e prestação de serviços de informática do Rio de Janeiro que anteriormente já havia registrado a marca Open Office, sob a alegação de que o nome OpenOffice.org, mesmo não sendo exatamente igual, poderia causar confusão aos usuários.

Desta maneira, o pacote OpenOffice.org não é mais distribuído oficialmente no português do Brasil, sendo em seu lugar disponibilizado oBrOffice.org. Para tanto, foi criada uma ONG, sendo seu primeiro presidente, Claudio Ferreira Filho. Já a partir da versão 2.1.0 foi adotado o novo nome BrOffice.org em detrimento do anterior OpenOffice.org.

Pode ser feito um download para se testar no Windows. No Linux, já vem instalado em várias distribuições, sendo disponibilizado no repositório da maioria das outras, ou por pacotes na página do próprio BrOffice.org.

É um programa destinado às tarefas de escritório, com diversos módulos, ou seja, possui editor de textos, planilha eletrônica para cálculos, gerenciador de apresentações, editor de páginas web, ferramenta para ilustrações, além de outras ferramentas.



## Servidores de E-MAIL, Servidores Web, Servidores Proxy

O sistema de e-mail consiste no trabalho de dois servidores diferentes. Um é chamada de servidor SMTP (Simple Mail Transfer Protocol – protocolo de transferência de correio simples). O outro pode ser tanto um servidor POP (Post Office Protocol) quanto um servidor IMAP (Internet Message Access Protocol). IMAP e POP são protocolos padrão da Internet para recebimento de e-mail, que baixa as mensagens do servidor para a sua máquina, smartphone ou tablet. SMTP é o protocolo para envio de e-mail.

O que é protocolo? Protocolo é a palavra que denomina “as regras” que organizam e regem a sincronização da comunicação entre dois sistemas computacionais. Ou seja, controla e possibilita a transferência de dados.

SMTP é usado quando o e-mail é entregue a partir de um cliente de e-mail a um servidor de e-mail ou quando o e-mail é entregue a partir de um servidor de e-mail para outro.

O IMAP tem acesso a todas as pastas de sua conta e deixa o status das mensagens igual tanto no servidor como no software e realiza a sincronia das mensagens, mantendo a conexão para que as alterações e mensagens novas recebidas sejam atualizadas quase que em tempo real.

O acesso via POP baixa as mensagens do servidor e salva as mesmas localmente em seu computador, não deixando uma cópia das mensagens no servidor – a menos que você marque a opção “deixar uma cópia no servidor” nas configurações do seu programa de e-mail. Esse protocolo tem acesso apenas à Caixa de Entrada, não conseguindo baixar nenhuma outra pasta de sua conta.



## Servidores de Email

Um servidor de email é um computador que envia, recebe e armazena emails para os usuários. Quase todos os Provedor de Serviços da Internet (ISP) inclui pelo menos uma caixa de correio em seu servidor de email como parte do seu serviço básico. Cada caixa de correio no servidor tem um nome exclusivo que é geralmente, mas nem sempre, o nome de logon usado para se conectar ao ISP. Cada caixa de correio está associada a um endereço de email exclusivo que geralmente consiste no seu nome de logon combinado com o nome de domínio do ISP no formato nome\_de\_logon@exemplo.com. Para acessar email em um servidor, você usa um programa de computador chamado cliente de email ou leitor de email, como o Outlook Express.

Quando um cliente se conecta ao servidor, ambos os computadores devem estar falando a mesma linguagem, chamada de protocolo.

No caso de servidores de email há quatro protocolos que podem ser usados. O mais amplamente usado é o POP3, para Protocolo PPO3. É quase sempre usado em conjunto com o SMTP ou Simple Mail Transfer Protocol.

O POP3 é usado para recuperar email do servidor (de entrada) enquanto que o SMTP é usado para enviar email através do servidor (de saída). O IMAP, ou Internet Message Access Protocol, é um protocolo mais novo não usado tanto quanto o POP3. O HTTP, ou HyperText Transfer Protocol, é na verdade o protocolo usado pelos servidores Web, mas também pode ser usado para acessar email em casos especiais, como o Hotmail.

O Outlook Express oferece suporte a todos esses quatro protocolos. Enquanto que todos os quatro protocolos executam funções de email, há diferenças importantes em como eles funcionam no servidor.



### Servidores POP3

Os servidores POP3, por padrão, armazenam email no servidor somente até você baixá-los em seu cliente de email. Assim que uma mensagem for baixada, o cliente comunica o servidor que a mensagem foi recebida e a cópia do servidor deve ser excluída. O Outlook Express segue este comportamento padrão, mas também lhe fornece a opção de "deixar uma cópia de mensagens no servidor".

Dica: Se você deseja manter uma cópia das mensagens no seu servidor de email POP3, é possível fazê-lo acessando a página de Propriedades avançadas da conta:

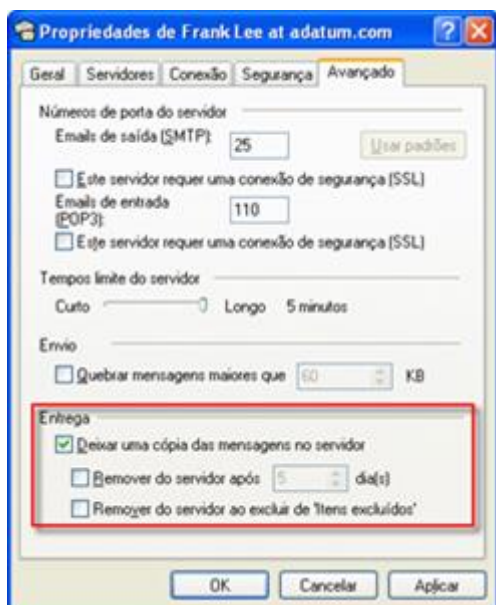
No menu Ferramentas do Outlook Express, clique em Contas.

1. Clique na guia Correio e clique duas vezes em sua conta de email.
2. Clique na guia Avançados.
3. Marque a caixa chamada "Deixar uma cópia das mensagens no servidor". Isso ativa duas opções de limpeza:

Se desejar, marque a caixa para "Remover do servidor após 5 dias". É possível alterar para qualquer número entre 1 e 100.

Se desejar, marque a caixa para "Remover do servidor quando excluído de 'Itens excluídos'".

Tome cuidado ao usar essa opção! A maioria dos ISPs limita o tamanho da caixa de correio POP3. Salvar mensagens no servidor pode fazer com que a caixa de correio encha muito rápido e talvez você não receba mensagens importantes até excluir as cópias do servidor.



### Servidores IMAP

Por outro lado, os servidores IMAP armazenam seu email permanentemente no servidor. O Outlook Express pode também armazenar cópias no seu próprio computador, desta forma, sua cópia local é um espelho exato de qualquer coisa que estiver no servidor. Ao excluir um email em uma conta IMAP, você está excluindo a cópia local e a cópia do servidor. Para evitar exclusões acidentais, os servidores IMAP não excluem a mensagem por completo até você emitir um comando. Limpar (Limpar itens excluídos no menu Editar ou no botão da barra de ferramentas Limpar). O IMAP também possui um comando para restaurar uma mensagem (também encontrado no menu Editar).

A vantagem principal do IMAP sobre POP3 é que, visto que todas as mensagens estão no servidor, você pode acessar todo seu email por qualquer computador com uma conexão à Internet. Também é possível criar subpastas no servidor (e na sua conta local do Outlook Express) para organizar melhor

seu email. A desvantagem é que a maioria dos ISPs limitam o tamanho da sua caixa de correio, desta forma uma caixa de correio IMAP encherá muito mais rápido do que uma caixa de correio POP3.

### **Servidores SMTP**

Os servidores SMTP são os verdadeiros trabalhadores de email, já que são os computadores enviando todos esses milhões de mensagens todos os dias. Mas como todos os usuários de email sabem, muitas destas mensagens estão infectadas com anexos com vírus ou são spam (email comercial não solicitado) ou são scams (deixe-me usar a sua conta bancária para tirar milhões de dólares da Nigéria). Para evitar remetentes de spam ou usuários maliciosos de enviar email através de seu servidor, a maioria dos ISPs colocou restrições sobre quem pode acessar seu servidor SMTP.

Para a maioria das pessoas este não é um problema, já que usam somente um ISP e uma conta de email hospedada por aquele ISP. Se você às vezes usa um segundo ISP ou se usa outro endereço de email além do fornecido pelo ISP, o servidor SMTP pode não o reconhecer como um usuário autorizado e recusará enviar as suas mensagens. Em vez disso, o servidor emitirá uma mensagem de erro que o Outlook Express exibirá. O texto exato varia, mas geralmente inclui o código de erro "550 - Retransmissão não permitida".

Os ISPs utilizam vários métodos para detectar usuários autorizados. Alguns configuram os servidores SMTP para aceitar somente email de computadores conectados à Internet usando o banco de modem do ISP, cabo ou linha DSL. Alguns também usam o endereço de email do remetente e rejeitam todos os emails enviados de endereços não hospedados pelo próprio ISP. Alguns permitem autorizar o servidor SMTP usando o mesmo nome e senha de logon usados para acessar o servidor POP3. Alguns usam um método conhecido como "POP antes do SMTP" pelo qual você deve primeiro acessar o servidor POP3 usando seu nome e senha de logon, o que fornece a permissão para acessar o servidor SMTP por alguns minutos. Se você encontrar problemas de retransmissão, precisará perguntar para a assistência técnica do ISP qual método eles usam e obter ajuda para configurar suas contas do Outlook Express.

O provedor de email mais comum que usa o HTTP no Outlook Express é o Hotmail. Para a maioria dos fins, o HTTP/Hotmail funciona de maneira semelhante ao IMAP, onde seu email é armazenado no servidor de email e o Outlook Express cria uma cópia espelhada no seu computador. A diferença mais importante é que o Hotmail não usa as funções Restaurar e Limpar. Excluir uma mensagem no Hotmail move a mensagem para a pasta Itens excluídos do Hotmail. A mensagem pode ser recuperada desta pasta, caso você mude de ideia, simplesmente arrastando de volta para a Caixa de correio ou outra pasta. Além disso, a pasta de Itens excluídos é excluída automaticamente após um intervalo de tempo.

### **Uma Resposta Simples**

A resposta mais simples à pergunta feita no início é realmente simples: verifique com seu ISP. Verifique o site do seu ISP para "configurar email" ou "nomes de servidor" ou até mesmo "endereços IP" e você provavelmente encontrará o que precisa. Se não for possível localizar as informações no site da Web, você pode ligar para o suporte técnico do seu ISP. Somente seu ISP pode informar com certeza os nomes de seus servidores e quais métodos de autenticação você deve usar para configurar seus servidores de email no Outlook Express de forma correta. Também é possível postar um mensagem no fórum do Outlook Express, mas entenda que apenas aqueles que utilizam o mesmo ISP provavelmente sabem os métodos de nomes dos servidores.

Assim que estiver munido do nome do servidor, o nome e senha de logon do seu servidor e o tipo de servidor, você está pronto para concluir o assistente Adicionar conta com o qual nós começamos. Se você está configurando uma conta do Hotmail, digite seu endereço do Hotmail quando o assistente solicitar seu endereço de email. O Outlook Express irá reconhecer como Hotmail e configurar o servidor automaticamente para você.

"Servidor web (web server)" pode referir ao hardware ou ao software, ou ambos trabalhando juntos.

1. Referente ao hardware, um servidor web é um computador que armazena arquivos que compõem os sites (por exemplo, documentos HTML, imagens, folhas de estilo, e arquivos JavaScript) e os entrega para o dispositivo do usuário final. Está conectado a Internet e pode ser acessado através do seu nome de domínio (DNS), como por exemplo mozilla.org.

2. Referente ao software, um servidor web inclui diversos componentes que controlam como os usuários acessam os arquivos hospedados (armazenados para disponibilização), no mínimo um servidor HTTP. Um servidor HTTP é um software que compreende URLs (endereços web) and e HTTP (o protocolo que seu navegador utiliza para visualizar páginas web).

Em um nível mais básico, o navegador fará uma requisição utilizando o protocolo HTTP sempre que necessitar de um um arquivo hospedado em um servidor web. Quando a requisição alcançar o servidor web correto (hardware), o servidor HTTP (software) enviará o documento requerido, também via HTTP.

Para publicar um website, é necessário ou um servidor web estático ou um dinâmico.

Um servidor web estático consiste em um computador (hardware) com um servidor HTTP (software). É chamado "estático" porque o servidor envia seus arquivos tal como foram criados e armazenados (hospedados) ao navegador.

Um servidor web dinâmico consiste em um servidor web estático com software adicional, mais comumente um servidor de aplicações (application server) e um banco de dados (database). É chamado "dinâmico" porque o servidor de aplicações atualiza os arquivos hospedados antes de enviá-los ao navegador através do servidor HTTP.

Por exemplo, para produzir as páginas finalizadas que você vê em seu navegador, o servidor de aplicações pode completar um modelo de página HTML (HTML template) com o conteúdo obtido de um banco de dados. Sites como o MDN ou a Wikipédia possuem vários milhares de páginas web, mas elas não são realmente documentos HTML, mas apenas alguns poucos templatesHTML e uma gigantesca base de dados. Essa configuração agiliza e facilita o gerenciamento e a entrega do conteúdo

Os servidores web são responsáveis por armazenar e trocar informações com outras máquinas. Por causa disso, pelo menos dois participantes são envolvidos em cada troca de informações: um cliente, que solicita informações, e um servidor, que atende a esses pedidos.

Cada lado exige também um programa especializado para negociar a troca de dados. No caso do cliente, é usado um browser, como o Internet Explore ou Firefox. No lado do servidor, porém, as coisas não são tão simples. Existem várias opções de software disponível, mais todos têm uma tarefa semelhante: negociar transferência de dados entre clientes e servidores via http (Protocolo de Transferência de Hipertexto), o protocolo de comunicações da Web. O software depende do sistema operacional escolhido para o servidor, mas iremos estudar um pouco sobre o Apache, uma opção disponível tanto para Linux quanto para Windows.

### **Como Funciona um Servidor Web**

Os servidores web estão disponíveis dia e noite, pois a capacidade de conexão deve estar disponível a qualquer momento. Os pedidos http que se referem às páginas HTML são feitos através de browsers. O processo é iniciado com conexão entre o computador onde está instalado o servidor web e o computador do usuário/cliente.

Em seguida, há o processamento do pedido do usuários e, conforme as parametrizações de segurança e a disponibilidade das informações solicitadas, o servidor entrega os dados (arquivos de texto, imagens, sons etc.) — tudo o que se enquadre no conceito de ficheiro pode ser enviado como resultado de um pedido http.

É importante também dizer que os servidores web também podem executar programas e scripts que os permitam interagir mais com os usuários.

Roxy é o termo utilizado para definir os intermediários entre o usuário e seu servidor. E por isso desempenha a função de conexão do computador (local) à rede externa (Internet). Como os endereços locais do computador não são válidos para acessos externos, cabe ao proxy enviar a solicitação do endereço local para o servidor, traduzindo e repassando-a para o seu computador.

Todas as requisições feitas ao servidor (o site que você quer acessar) passarão pelo seu proxy. Ao chegar ao site, o IP (Internet Protocol / Protocolo de Internet) do proxy fica registrado no cache do seu destino e não o seu. É pelo IP que os hackers conseguem invadir computadores, portanto deve-se

manter o nível de segurança do seu gateway (porta de ligação com o proxy) seguro. Os riscos são vários, no entanto, dois deles podem ser enumerados como os mais fortes: ter seu computador invadido ou ter alguém navegando com o seu IP.

O IP funciona como o número da carteira de identidade do seu computador. É ele que determina quem é você na rede; e é por este número que outras pessoas poderão encontrá-lo (caso você possua um endereço FTP para envio de arquivos direto para as suas pastas). O que o proxy faz é mascarar o seu número de IP legítimo para que quando um cracker quiser invadir seu computador, fique mais complicado.

Os proxies ajudam na aceleração do acesso à internet no caso de empresas que precisam de velocidade na hora de navegar. O registro da página acessada fica guardado na sua cache. Com este arquivo já gravado, o próximo acesso fica muito mais rápido uma vez que não será necessário refazer o primeiro reconhecimento do destino.

### **Web Proxy**

Existe, ainda, um outro tipo de proxy, os web proxies. Eles são uma versão que esconde o seu IP real e lhe permite navegar anonimamente. Muitos deles são utilizados em redes fechadas como universidades e ambientes de trabalho para burlar uma determinação de bloqueio a alguns sites da internet. Os conteúdos campeões de bloqueio são: sites de relacionamento (Orkut, facebook e outros), programas de troca de mensagem instantânea (Msn Messenger, Yahoo! Messenger e outros), sem contar os tão proibidos sites de pornografia.

### **Open Proxy**

As conexões abertas de proxy (open proxy) são o tipo mais perigoso e convidativo aos crackers e usuários mal intencionados. Quando um destes usuários consegue acessar um computador, instala um servidor proxy nele para que possa entrar quando quiser na máquina e promover diversos tipos de ilegalidade como scripts que roubam senhas de bancos, fraudes envolvendo cartões de crédito e uma grande variedade de atos ilegais.

### **Redes Proxy**

As redes proxy são baseadas em códigos criptados que permitem a comunicação anônima entre os usuários. Exemplo deste tipo de rede são as conexões P2P (peer to peer) em que um usuário se conecta ao outro sem saber sua identidade e trocam arquivos entre si.

Estas redes se caracterizam por não permitirem o controle dos servidores, os usuários comuns é quem providenciam todo o conteúdo e os arquivos. Certamente, muitos destes computadores são usados por pessoas mal-intencionadas com segundas intenções. Por isso, deve-se ter em mente que qualquer promessa de privacidade e segurança é mais do que rara.

O proxy serve como um intermediário entre os PCs de uma rede e a Internet. Um servidor proxy pode ser usado com basicamente três objetivos: 1- Compartilhar a conexão com a Internet quando existe apenas um IP disponível (o proxy é o único realmente conectado à Web, os outros PCs acessam através dele).

2- Melhorar o desempenho do acesso através de um cache de páginas; o proxy armazena as páginas e arquivos mais acessados, quando alguém solicitar uma das páginas já armazenadas do cache, esta será automaticamente transmitida, sem necessidade de baixá-la novamente.

3- Bloquear acesso a determinadas páginas (pornográficas, etc.), como tipo passa pelo proxy é fácil implantar uma lista de endereços ou palavras que devem ser bloqueadas, para evitar por exemplo que os funcionários percam tempo em sites pornográficos em horário de trabalho.

Hoje em dia os servidores proxy são extremamente comuns, mesmo em redes domésticas, não é necessário um PC dedicado a esta função, basta instalar um dos vários programas de servidor proxy disponíveis no PC com a conexão à Internet: Wingate, Analog-X, etc.

## O que é um Servidor Proxy Web?

Você já deve ter se deparado com o termo algumas vezes, principalmente se já trabalhou com redes de computadores. Mas o que são estes serviços, e como funcionam?

Os serviços de web proxy surgiram a partir da necessidade de conectar uma rede local a internet através de um equipamento que compartilha a conexão com as demais máquinas. Assim, o proxy web é um serviço que atua como intermediário entre um dispositivo e os serviços de internet. No momento em que o endereço de um site é digitado no navegador, a solicitação é enviada ao proxy, que então realiza esta solicitação ao servidor no qual o site é hospedado, devolvendo para o cliente o resultado.

Em outras palavras, isso significa que o proxy faz o intermédio da conexão para o dispositivo cliente, e por conta disso, tem controle absoluto sobre o tráfego em si, podendo permitir ou bloquear de acordo com as políticas estabelecidas pela empresa.

## Por que Utilizar o Proxy Web?

### Controle do Uso Da Internet

O proxy web permite controlar os serviços acessados na internet através do protocolo HTTP, sendo responsável gestão de acesso a sites e outras aplicações baseadas no protocolo. O proxy é amplamente utilizado por empresas dos mais variados portes e segmentos, contribuindo para a manutenção da estratégia de segurança do negócio.

O serviço de proxy web permite que a empresa gerencie os acessos de colaboradores, tendo por base URLs, horários, grupos de usuários, além de outros níveis de controle. Com a implantação do serviço é possível ter acesso a relatórios detalhados de navegação, incluindo consumo de recurso, tempo de navegação, sites mais acessados, além de outras funções que auxiliam na manutenção da integridade do recurso de internet e da rede corporativa.

## O que é servidor Apache?

Criado em 1995 por Rob McCool, na época funcionário da NCSA (National Center for Supercomputing Applications), o servidor Apache ou Servidor HTTP Apache é o mais bem-sucedido servidor web livre que existe. Trata-se de um servidor web muito popular, utilizado principalmente no Linux.

Assim como qualquer servidor do tipo, o Apache é responsável por disponibilizar páginas e todos os recursos que podem ser acessados pelo internauta. Envio de e-mails, mensagens, compras online e diversas outras funções podem ser executadas graças a servidores como o Apache. O que vale destacar no Apache é que, apesar de tudo, ele é distribuído sob a licença GNU, ou seja, é gratuito e pode ser estudado e modificado através de seu código fonte por qualquer pessoa.

Uma pesquisa realizada em dezembro de 2007 revelou que o servidor Apache representa cerca de 47,20% dos servidores ativos no mundo. Esse número aumentou em maio de 2012, quando foi constatado que o Apache servia aproximadamente 54,68% de todos os sites e 66% dos milhões de sites mais movimentados do mundo.

A Apache Software Foundation é a responsável pelo projeto, além de desenvolver e trabalhar com outras tecnologias de transmissão via web, processamento de dados e execução de aplicativos distribuídos. Aliás, o termo que nomeia o servidor e a empresa responsável por ele faz referência à nação Apache, uma tribo de nativos americanos que possuía grande resistência e estratégias superiores em combate. O significado do nome seria uma alusão à resistência da comunidade do software livre aos ataques de interesses privados de alguns setores e corporações. O significado do nome também discorre sobre a estabilidade do servidor e a sua diversidade de ferramentas e recursos que são capazes de lidar com qualquer tipo de solicitação executada na web.

A respeito do nome ainda existem duas outras razões do servidor ser chamado de Apache. A primeira delas, refutada pela Fundação, é de que o nome estaria relacionado a "patchy server", que em inglês significa algo como servidor remendado ou melhoria do software. A outra explicação diz respeito ao lançamento do Tomcat, que é um sistema auxiliar ao Apache. Tomcat, na verdade, é o nome de uma aeronave norte-americana e Apache é o nome de um helicóptero de ataque estadunidense.

O servidor Apache é compatível com o protocolo HTTP versão 1.13. Suas funcionalidades são mantidas através de uma estrutura de módulos, o que permite que os usuários escrevam seus próprios módulos por meio da API do software. Ele está disponibilizado para Windows, Novell Netware, OS/2 e outros sistemas do padrão POSIX, como o Unix e o Linux, onde é amplamente utilizado.

Com relação ao hardware para suporte ao servidor, tudo depende de sua aplicação. Contudo, um computador com processador Pentium e 64 MB de memória RAM já é capaz de executá-lo sem grandes problemas em um ambiente corporativo pequeno. Os computadores disponíveis no mercado, então, são mais do que capazes de colocar um servidor Apache em funcionamento, lembrando, logicamente, que depende-se de sua utilização.

O Apache dispõe de um módulo denominado de mod\_ssl, o qual adiciona a capacidade do servidor de atender solicitações usando o protocolo HTTPS. Este protocolo faz uso da camada SSL para a criptografia de todos os dados transferidos, proporcionando maior segurança entre o tráfego de dados entre cliente e servidor.

### **Servidor WEB Apache, o que esperar?**

O que é um Servidor WEB?

O Servidor WEB é o programa responsável pela publicação de documentos, imagens ou qualquer outro objeto que venha a ser acessado por um cliente através de um navegador.

Este Servidor WEB pode ser configurado para ser acessado apenas em uma rede interna (Intranet) ou uma rede externa (Internet), cada qual com suas funcionalidades definidas pelas necessidades da publicação.

O Servidor WEB mais utilizado no mundo segundo a pesquisa realizada "TOP SERVERS ACROSS ALL DOMAINS AUGUST 1995 - OCTOBER 2006" da NETCRAFT (<https://news.netcraft.com/>), é conhecido como? Alguém arrisca? o nosso bom Apache, chegando a 61.44% de reconhecimento e utilização, ficando na frente de outras grandes empresas (Microsoft, Zeus e Sun).

Logo, durante nossa Jornada no mundo dos Servidores WEB Apache, estaremos tratando de um assunto que tem reconhecimento e aceitação mundial.

Algumas boas e importantes qualidades do Apache é ele ser configurável, robusto, de alta performance (quando bem configurado todas essas qualidades são vistas facilmente), de fácil instalação e seu código fonte ser distribuído gratuitamente pela equipe de desenvolvedores do Apache Software Foundation.

Vamos iniciar aprendendo a Instalar o Apache em ambiente Windows

Para maiores informações sobre os arquivos de instalação do Apache, consulte a página de download citada acima. Clique duas vezes no arquivo de instalação e uma tela como essa deverá aparecer:



Figura. Instalação do Apache

Prossiga clicando no botão "Next", aceitando os termos da licença de uso e, na tela de Informações do Servidor ("Server Information"), onde é solicitado "Network Domain", "Server Name" e "Administra-

tor's Email Address", informe, respectivamente: "localdomain", "localhost" e o seu endereço de e-mail. Deixe também selecionado a opção "Run as a service for All users", como indicado abaixo:



Figura. Configurando a instalação do Apache

Prossiga selecionando a instalação completa ("Complete") e finalize a instalação.

O procedimento abaixo não torna-se necessário no Windows XP/2000/NT, uma vez que se escolheu a opção Run as a service for All Users.

Instalação concluída agora para INICIAR o Servidor Web Apache, vá no botão Iniciar->Programas->Apache HTTP Server e clique em Start Apache in Console. Uma janela como esta deverá aparecer, indicando que o Apache está em execução:

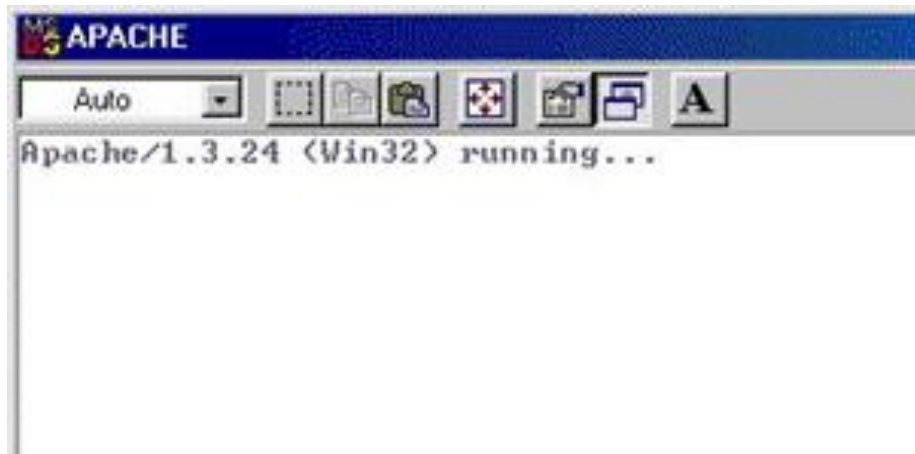


Figura. Start Apache in Console

Para PARAR o Apache, não é aconselhável que se feche esta janela diretamente. Ao invés disso, vá em Iniciar->Programas->Apache HTTP Server e clique em Stop Apache. Isso dará início ao processo de shutdown do servidor Web, o que fará com que a janela acima se feche.

Para testar a instalação, primeiramente, certifique-se de que o Apache esteja em execução. Abra o seu navegador favorito e digite o seguinte endereço: <http://localhost>. Este é o endereço do seu Servidor Web local.

Uma página como esta abaixo deverá aparecer, indicando que você instalou e configurou corretamente o Servidor Web Apache em seu sistema.



Figura. Configurando corretamente o Servidor Web Apache em seu sistema

## Servidor Apache

O servidor Apache (ou Servidor HTTP Apache, em inglês: Apache HTTP Server, ou simplesmente: Apache) é o mais bem sucedido servidor web livre. Foi criado em 1995 por Rob McCool, então funcionário do NCSA (National Center for Supercomputing Applications). Numa pesquisa realizada em dezembro de 2007, foi constatado que a utilização do Apache representa 47.20% dos servidores ativos no mundo. Em setembro de 2009 Apache serviu mais de 54,48% de todos os sites e mais de 66% dos milhões de sites mais movimentados.

É a principal tecnologia da Apache Software Foundation, responsável por mais de uma dezena de projetos envolvendo tecnologias de transmissão via web, processamento de dados e execução de aplicativos distribuídos.

O servidor é compatível com o protocolo HTTP versão 1.1. Suas funcionalidades são mantidas através de uma estrutura de módulos, permitindo inclusive que o usuário escreva seus próprios módulos utilizando a API do software.

É disponibilizado em versões para os sistemas Windows, Novell Netware, OS/2 e diversos outros do padrão POSIX (Unix, Linux, FreeBSD, etc).

A razão para o nome "Apache" dada pela Apache Software Foundation, é que faz referência à nação Apache, tribo de nativos americanos que tinha, em combate, grande resistência e estratégias superiores. Isso seria uma alusão à estabilidade do servidor Apache e a sua variedade de ferramentas capazes de lidar com qualquer tipo de solicitação na web.

Uma segunda razão, que é aceita popularmente, reconhecida porém refutada pela Fundação, é que o nome viria de "a patchy server", que em inglês significa algo como um servidor remendado, ou melhoria no software, dada a origem do programa, criado sobre o código do servidor da NCSA já existente, no qual foram adicionados diversos patches.

Surge uma terceira explicação para o nome com o lançamento do Tomcat (um sistema auxiliar ao Apache que cuida basicamente de processamento de aplicativos em Java), já que "Tomcat" nome de uma aeronave estadunidense, o F-14 Tomcat e "Apache" é o nome de um helicóptero de ataque, o AH-64 Apache).

Para garantir segurança nas transações HTTP, o servidor dispõe de um módulo chamado mod\_ssl, o qual adiciona a capacidade do servidor atender requisições utilizando o protocolo HTTPS.

Este protocolo utiliza uma camada SSL para criptografar todos os dados transferidos entre o cliente e o servidor, provendo maior grau de segurança, confidencialidade e confiabilidade dos dados. A camada SSL é compatível com certificados X.509, que são os certificados digitais fornecidos e assinados por grandes entidades certificadoras no mundo.

O servidor é configurado por um arquivo mestre nomeado httpd.conf e opcionalmente pode haver configurações para cada diretório utilizando arquivos com o nome .htaccess, onde é possível utilizar

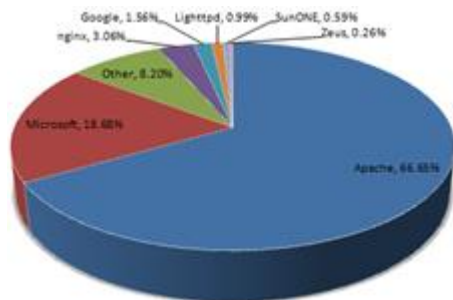


autenticação de usuário pelo próprio protocolo HTTP utilizando uma combinação de arquivo .htaccess com um arquivo .htpasswd, que guardará os usuários e senhas (criptografadas).

O que é o Apache?

Toda vez que você acessa um site, uma requisição é enviada ao servidor em que o site está rodando, este por sua vez realiza algum processamento baseado nestas requisições e responde à solicitação com algum serviço. Um servidor é um computador disponibilizado em uma rede com o objetivo de prover serviços para outros hosts. No caso de sites estamos falando do servidor Web.

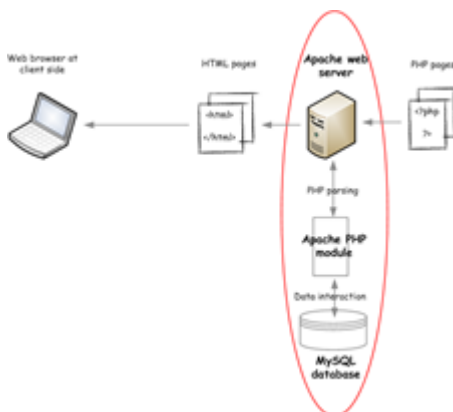
O servidor Apache ou Servidor HTTP Apache é o servidor web mais utilizado do mundo. De acordo com uma análise realizada em 2009 pela Netcraft 66,6% de todos os sites da web rodavam em Apache.



Como funciona o servidor Web Apache?

Funcionando na estrutura cliente-servidor, o servidor web Apache, recebe as requisições do cliente (browser) e responde ao cliente em codificação HTML. O servidor web interpreta a codificação HTML e não interpreta o código PHP.

Ao receber uma solicitação .php o servidor aciona o Interpretador PHP que processa as solicitações do código PHP tais como, acessar banco de dados, sistema de arquivos, acesso ao servidor de correio eletrônico etc e retorna para o Apache em formato HTML e ele manda para o browser. O browser lê o código HTML e monta a página web para o usuário.



O Apache é desenvolvido na plataforma de software livre e segue as quatro liberdades da Free Software Foundation, e é fundamental para o funcionamento da web.

Instalando o Apache em ambiente Linux:

Para facilitar a instalação usaremos o gerenciador de pacotes Aptitude.

Atualizar a base de dados local:

```
#aptitude update
```

Procurar pacotes:

#aptitude search apache2

Instalar Apache

#aptitude install apache2

Para testar se o Apache está instalado e se o serviço está inicializado, acesse no browser: <http://localhost/>. Irá aparecer a mensagem abaixo:

It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Pronto! Seu servidor web Apache está instalado e funcionando.

### **O Servidor Apache**

Ao acessar qualquer site, há um servidor por trás daquele endereço responsável por disponibilizar as páginas e todos os demais recursos que você pode acessar. Assim, quando você envia um e-mail através de um formulário, coloca uma mensagem em um fórum de discussão, faz uma compra on-line, etc, um servidor Web (ou um conjunto de servidores) é responsável por processar todas essas informações.

Sendo mais claro, um servidor Web é um computador que processa solicitações HTTP (Hyper-Text Transfer Protocol), o protocolo padrão da Web. Quando você usa um navegador de internet para acessar um site, este faz as solicitações devidas ao servidor Web do site através de HTTP e então recebe o conteúdo correspondente. No caso do Apache, ele não só executa o HTTP, como outros protocolos, tais como o HTTPS (O HTTP combinado com a camada de segurança SSL - Secure Socket Layer), o FTP (File Transfer Protocol), entre outros.

Como servidor Web, o Apache é o mais conhecido e usado. Os motivos incluem sua excelente performance, segurança, compatibilidade com diversas plataformas e todos os seus recursos.

O servidor Apache (ou Apache Server) surgiu no National Center of Supercomputing Applications (NCSA) através do trabalho de Rob McCool. Ao sair da NCSA, McCool parou de trabalhar no software (que nessa época recebia justamente a denominação NCSA) e então várias pessoas e grupos passaram a adaptar o servidor Web às suas necessidades. No entanto, foram Brian Behlendorf e Cliff Skolnick os principais responsáveis pela retomada do projeto, contando logo em seguida com o apoio de Brandon Long e Beth Frank. Estes últimos tinham a tarefa de continuar com o desenvolvimento do servidor, mas pela NCSA. Não demorou muito para que eles se juntassem ao Apache Group.

A origem da Fundação Apache (Apache Foundation) vem daí: Brian Behlendorf e Cliff Skolnick passaram a controlar "patches" (adição de recursos ou correções) atrás de uma lista de discussão. Dessa lista foi formado o grupo, constituído inicialmente de 8 pessoas. O nome da fundação parece ter sido baseado nessa característica (uso de patches), já que pode ser interpretado como um trocadilho com a expressão em inglês "a patchy". No entanto, segundo os desenvolvedores do servidor, o nome é baseado numa tribo americana de nome Apache.

O servidor Apache teve sua primeira versão oficial (0.6.2) lançada em 1995. No momento em que este artigo era escrito, a última versão era a 2.2.3/2.2.4. De 1995 até hoje houve várias mudanças no projeto. Talvez, a mais importante tenha sido o Shambhala, uma arquitetura para o Apache desenvolvida por Robert Thau.

O Shambhala agradou tanto na época (principalmente por seu melhor gerenciamento de memória) que acabou sendo incorporado definitivamente ao servidor, tirando de cena a base existente até então. Surgia a versão 0.8.8, que tinha, além do Shambhala, recursos da versão 0.7.x. Após uma série de aperfeiçoamentos, testes e de uma documentação nova (feita por David Robinson), o Apache 1.0 foi lançado em dezembro de 1995.

### **Características do Servidor Apache**

O Apache Server é um software livre, o que significa que qualquer um pode estudar ou alterar seu código-fonte, além de poder utilizá-lo gratuitamente. É graças a essa característica que o software foi (e continua sendo) melhorado ao passar dos anos. Graças ao trabalho muitas vezes voluntário de vários desenvolvedores, o Apache continua sendo o servidor Web mais usado no mundo.

Além de estar disponível para o Linux (e para outros sistemas operacionais baseados no Unix), o Apache também conta com versões para o Windows, para o Novell Netware e para o OS/2, o que o torna uma ótima opção para rodar em computadores obsoletos (desde que este atenda aos requisitos mínimos de hardware).

O servidor Apache é capaz de executar código em PHP, Perl, Shell Script e até em ASP e pode atuar como servidor FTP, HTTP, entre outros. Sua utilização mais conhecida é a que combina o Apache com a linguagem PHP e o banco de dados MySQL (combinação usada aqui no InfoWester, por exemplo).

A exigência de hardware do Apache depende de sua aplicação, mas um PC Pentium com 64 MB de memória RAM é capaz de executá-lo tranquilamente em um ambiente corporativo pequeno. No entanto, quando se trata de um site na internet, é interessante ter máquinas tão poderosas quanto o que exige o nível de acesso.

Abaixo, segue um resumo com as principais características (extraído do Guia Foca Linux):

- Possui suporte a scripts cgi usando linguagens como Perl, PHP, Shell Script, ASP, etc;
- Suporte a autorização de acesso podendo ser especificadas restrições de acesso separadamente para cada endereço/arquivo/diretório acessado no servidor;
- Autenticação requerendo um nome de usuário e senha válidos para acesso a alguma página/sub-diretório/arquivo (suportando criptografia via Crypto e MD5);
- Negociação de conteúdo, permitindo a exibição da página Web no idioma requisitado pelo Cliente Navegador;
- Suporte a tipos mime;
- Personalização de logs;
- Mensagens de erro;
- Suporte a virtual hosting (é possível servir 2 ou mais páginas com endereços/ portas diferentes através do mesmo processo ou usar mais de um processo para controlar mais de um endereço);
- Suporte a IP virtual hosting;
- Suporte a name virtual hosting;
- Suporte a servidor Proxy ftp e http, com limite de acesso, caching (todas flexivelmente configuráveis);
- Suporte a proxy e redirecionamentos baseados em URLs para endereços Internos;
- Suporte a criptografia via SSL, Certificados digitais;
- Módulos DSO (Dynamic Shared Objects) permitem adicionar/remover funcionalidades e recursos sem necessidade de recompilação do programa.

### **Instalando o Apache no Linux**

A instalação do servidor Apache no Linux é fácil, principalmente por estar, na maioria das vezes, atrelada a facilidades existentes na própria distribuição Linux. Por exemplo, se você usa um sistema operacional baseado no Debian (ou o próprio), como o Ubuntu, pode instalar o Apache através do comando apt-get:

```
apt-get install apache
```

O mais recomendável é instalar as versões do Apache preparadas para sua distribuição, já que provavelmente o software estará otimizado para rodar nela. Se preferir, você pode baixar pacotes "genéricos" no site da Apache Foundation. Feito isso, a instalação pode ser executada da seguinte forma:

Supondo que você baixou o pacote httpd-2.2.2.tar.bz2, digite em um terminal:

```
tar -xjvf httpd-2.2.2.tar.bz2 cd httpd-2.2.2 (supondo que esta é a pasta criada)
```

O comando acima descompactou o pacote e o de baixo inicia o processo de compilação e instalação:

```
./configure --prefix=/etc/httpd (se preferir, indique após o sinal de igual o local de instalação do Apache)
```

Na verdade, o comando anterior pode ser incrementado com a adição ou a remoção de módulos. Para isso, após "configure", informe "--enable-módulo" para adicionar um módulo ou "--disable-módulo" para fazer o contrário. Por exemplo, se você digitar:

```
./configure --enable-ssl --enable-rewrite --prefix=/etc/httpd
```

Significa que você está habilitando o suporte a SSL e ao "rewrite" que, basicamente, permite "reescrever" endereços usando expressões regulares.

Findo este processo, digite: make e, em seguida, make install para concluir a instalação.

Executando o Apache no Linux

Após instalar o Apache, é hora de executá-lo. Para isso há mais de uma forma, uma delas é a seguinte, que você pode digitar em um terminal:

```
/etc/httpd/bin/apachectl start
```

No comando acima, altere para o caminho correspondente, caso o Apache esteja instalado em um diretório diferente. Em alguns casos, basta abrir o terminal e digitar apenas "apachectl start". Isso será suficiente para iniciar o Apache.

O parâmetro start pode ser substituído por outros, como:

```
restart - para reiniciar o Apache;  
stop - para parar o Apache;  
status - para mostrar o status do processo do Apache.
```

Após iniciar o Apache, é possível testá-lo: abra um navegador de internet instalado na máquina e digite no campo de endereço: <http://localhost:80> (em alguns casos, basta digitar localhost). Se o Apache estiver funcionando corretamente.

Provavelmente você irá querer que o Apache inicie automaticamente toda vez que o sistema operacional for carregado. Isso pode ser feito da seguinte forma: abra o arquivo `/etc/rc.d/rc.local` (ou equivalente, de acordo com o seu sistema operacional) e acrescente nele a linha:

```
/etc/httpd/bin/apachectl start
```

Existem várias formas de iniciar o Apache automaticamente. Outra opção, para quem utiliza o Fedora Core em português, por exemplo, é a seguinte (considerando que o Apache está devidamente instalado): acesse o menu (chapéu vermelho), vá em Configurações do Sistema / Configurações do Servidor / Serviços. No menu à esquerda, habilite a opção "httpd" e clique no botão Salvar.

O passo seguinte é configurar o Apache (é recomendável fazer isso com o servidor parado). Para isso, deve-se acessar o diretório conf - em nosso caso, `/etc/httpd/conf` - e abrir os arquivos de configuração. Nessa pasta você encontra, basicamente, os arquivos `httpd.conf`, `srm.conf` (dependendo da versão) e `access.conf` (dependendo da versão).

Dos três citados, o primeiro é o mais importante, pois é responsável pelas configurações usadas pelo daemon do Apache, o `httpd` (daemon é uma espécie de programa que executa tarefas em segundo plano). O arquivo `srm.conf` tem a tarefa de indicar os recursos que o Apache precisa, enquanto que o `access.conf` contém diretivas de segurança para controlar o acesso a arquivos e diretórios.

Por este ser um artigo introdutório, não serão dados detalhes sobre como configurar esses arquivos, mas o último tópico desta página contém ótimos links que ensinam a fazer isso.

**TUX**



## Virtualização de Servidores

Antes de falar em Virtualização de Servidores, é importante ressaltar que a Tecnologia da Informação (TI) define rumos no dinâmico mundo dos negócios.

Certamente, você já faz uso de diferentes recursos tecnológicos em sua empresa, pois a empresa mais simples da atualidade dispõe de pelo menos, um computador e uma impressora.

A virtualização é a criação da versão virtual de um sistema operacional, um servidor, um sistema de armazenamento, etc. Quando falamos em virtualização de servidores estamos falando em dividir os recursos de um hardware (servidor físico) em diversos servidores virtuais que podem ser usados com finalidade diferentes.

Dessa forma, é possível executar muitos sistemas operacionais a partir de um mesmo hardware, sem que haja interferência entre esses sistemas.

Como isso funciona?

Primeiramente, é preciso ter o servidor físico com seus recursos (CPU, memória, discos, conexões de rede e conexões a SAN). Nesse hardware, é instalado um sistema operacional básico que apresenta a capacidade de dividi-lo em partes menores (fatiamento do hardware).

Esse sistema operacional recebe o nome de hypervisor e cada uma dessas partes corresponde a uma máquina virtual. Quanto menor for o espaço em disco e a memória, mais recursos terão as máquinas virtuais e menores serão as possibilidades de problemas e a necessidade de manutenção.

O hypervisor simula, em cada divisão do hardware, uma máquina virtual. Cada máquina virtual possui capacidades diferentes conforme suas próprias necessidades.

Assim, uma delas pode ter mais memória, a outra mais espaço em disco e outra mais processador. Na configuração dos hypervisores mais desenvolvidos, é possível dividir o tráfego, priorizando conforme a máquina virtual.

Os discos ficam arquivados dentro do sistema operacional ao passo que a CPU e a memória são utilizadas sob demanda.

Dentro de cada uma dessas máquinas virtuais, será instalado um sistema operacional diferente, que funcionará isolado dos outros e usará somente os recursos que lhe forem destinados.

Um servidor com 10 GB de memória pode ter 5 sistemas com 2 GB cada desde que a totalidade dos recursos não seja usada ao mesmo tempo.

Como é a virtualização de servidores em empresas?

Em ambientes empresariais que possuam alta disponibilidade, as máquinas virtuais ficarão armazenadas em uma SAN, ou seja, em um espaço de armazenamento compartilhado por todos os servidores. Esse espaço é chamado de storage. Caso o SAN seja virtual recebe o nome de VSAN.

Para as empresas, a possibilidade de gerenciar os discos a partir de um único ponto representa uma vantagem, permitindo que se faça uma distribuição de desempenho mais uniforme e que sejam definidas prioridades entre os sistemas operacionais instalados.

Quais os benefícios da virtualização de servidores?

A virtualização de servidores traz diversas vantagens para a empresa, entre as quais podemos destacar:

Redução de downtime (tempo de inatividade do sistema): a virtualização evita perdas de dados, paradas não planejadas, paradas no ambiente de produção e assim por diante;

Automação mais apurada dos processos;

Taxa de utilização do servidor pode subir para 65% a 90%;

Maior retorno sobre o investimento;

Redução de custo operacional (até 40%), inclusive de energia elétrica;

Otimização do espaço físico;

Melhor utilização dos recursos disponíveis;

Gerenciamento centralizado;

Backups facilitados e mais segurança;

Facilidade de recuperação em caso de desastres;

Independência do servidor físico;

Aplicações em plena compatibilidade;

Adaptação a diferentes cargas de trabalho;

Migração de ambientes fácil e transparente;

Uso de máquina virtual (VM) como ambiente de desenvolvimento;

Possibilidade de simulações.

Quais são os mitos sobre a virtualização de servidores?

Apesar de trazer muitos benefícios para a empresa, a virtualização ainda causa dúvidas em muitos interessados em razão das complexidades que traz consigo, como a configuração do hardware e as necessidades do monitoramento de desempenho e segurança.

Os especialistas em TI devem orientar seus clientes para que eles tomem as melhores decisões ao investir em novas tecnologias. No caso da virtualização de servidores, os principais mitos que precisam ser desfeitos são:

Se o servidor físico quebrar, não há mais acesso às VMs

Essa afirmação é incorreta. As VMs podem ser acessadas por outros servidores, sem haver a necessidade de reinstalar todos os seus programas e configurações.

Como o ambiente é virtual, não há controle sobre a segurança

Outra premissa falsa. Tanto servidores físicos quanto virtuais têm as mesmas configurações. São as empresas que devem definir as normas de segurança a serem seguidas e garantir seu cumprimento.

Empresas pequenas não se beneficiam da virtualização de servidores

Essa tecnologia é indicada para qualquer empresa que necessite de mais de um servidor, uma vez que economiza espaço físico e de infraestrutura, além de ser de fácil gerenciamento e manutenção.

### **A Tecnologia tem Custo Alto**

Os investimentos iniciais (contratação de provedores, aquisição de equipamentos e softwares) acabam retornando no médio e no longo prazo em forma de economia de energia, menos manutenção e diminuição de gastos operacionais.

Além disso, é comum que os provedores ofereçam treinamento para toda a equipe. Assim, os custos com a qualificação de pessoal são eliminados e eles estarão preparados para lidar com a tecnologia no dia a dia sem perda de tempo.

Sistemas virtuais têm desempenho inferior

Em termos de desempenho, os resultados são semelhantes. A verdade é que a virtualização tem a vantagem de não deixar computadores ociosos: todo o espaço contratado é dedicado ao trabalho da empresa, o que permite uma melhor alocação de recursos.

Por onde começar?

O primeiro passo é avaliar se o investimento na virtualização de servidores pode melhorar o desempenho da empresa. Vale pensar se:

haverá redução significativa de custos;

Facilitará os processos internos;

Dará agilidade ao processamento de informações;

Haverá aumento na produtividade.

Caso as respostas para essas questões não estejam claras ou se os ganhos não forem significativos, talvez a virtualização não seja o melhor caminho a seguir. Além disso, é fundamental pensar em quem será responsável por esses servidores virtuais — que faça sua manutenção e ajustes quando necessário.

A transferência de dados e processos para o ambiente digital é um caminho natural que a tecnologia vem abrindo para todos nós. Antigamente, guardávamos inúmeros documentos impressos em casa, enquanto hoje eles são digitalizados. Basta voltar um pouco mais no tempo para lembrar da comunicação via cartas escritas à mão — papel desempenhado atualmente pelos e-mails.

A virtualização de servidores é um passo além nesse mesmo processo. Na era dos computadores, as empresas passaram a digitalizar seus documentos e armazená-los em uma infraestrutura de TI própria. Agora, com equipamentos mais poderosos e transmissões de dados ultrarrápidas, chegou a hora de virtualizar os próprios servidores.

Estamos falando de criar uma versão virtual de um sistema de armazenamento. Grosso modo, os recursos de hardware podem ser distribuídos entre diferentes espaços digitais que podem ser utilizados para finalidades diferentes.

Sua principal característica está na possibilidade de rodar diferentes sistemas operacionais e outras plataformas de forma independente, mas simultânea, dentro de um mesmo hardware. Afinal, os servidores virtuais são simulações do que seria a estrutura física do servidor como ele era compreendido anteriormente.

Qual é a diferença entre o servidor virtual e a computação em nuvem?

A virtualização de servidores está na essência da computação em nuvem (cloud computing). Isso causa uma certa confusão entre os dois conceitos, já que eles estão relacionados. Entretanto, é fundamental entender as características específicas de cada um, para que sua empresa possa optar pela solução que melhor atende as suas necessidades.

A computação em nuvem envolve a migração de dados, arquivos, plataformas, aplicações e outros elementos para um servidor fora da corporação. Esse servidor é dependente da virtualização para funcionar, mas isso não significa que todo servidor virtual depende de cloud computing.

Enquanto é possível criar um servidor virtual dentro de sua empresa, a computação em nuvem depende da contratação de uma empresa especializada para fornecer a infraestrutura e gerenciá-la. O acesso, em sua empresa, será feito de maneira remota, pela internet.

Toda a configuração dos servidores e plataformas, mesmo que de acordo com o que sua empresa solicita, é feita pelo provedor do serviço. Isso inclui questões de segurança, criptografia etc.

A virtualização é utilizada por ele para repartir o hardware em diferentes partes, permitindo que ele ofereça o serviço para diversos clientes ao mesmo tempo.

Qual é o impacto da virtualização de servidores nas empresas?



Implementar servidores virtuais pode ser uma estratégia interessante para diferentes empresas, independentemente de seu tamanho ou setor de atuação. O importante é fazer uma análise cuidadosa de seus objetivos e entender os impactos dessa mudança nos seus processos.

Partiremos da implementação para depois destacarmos seus prós e contras.

### **A Instalação de um Servidor Virtual**

Primeiro, é preciso contar com um servidor físico (CPU, memória, conexões de rede etc.). Nesse hardware, é instalado um sistema operacional básico (hypervisor) que pode ser dividido em partes menores (processo de fatiamento) — cada parte corresponde a uma máquina virtual independente.

Tenha em mente que máquinas com pouco armazenamento e processamento serão mais facilmente gerenciadas, evitando falhas inesperadas. Entretanto, é possível configurá-las em tamanhos variados, de acordo com o objetivo da empresa para cada uma.

Os hypervisors mais eficientes são capazes de, inclusive, dividir o tráfego, priorizando algumas máquinas para otimizar o desempenho do sistema. Em geral, pequenas empresas podem ter uma infraestrutura mais simples.

Ainda assim, é indicado utilizar um SAN, ou seja, um espaço compartilhado entre todos os servidores, também chamado de storage.

### **Prós e Contras**

Em primeiro lugar, é preciso destacar a redução significativa dos custos para a empresa. Os gastos operacionais podem ser até 40% menores, incluindo o uso de energia elétrica. E isso também se aplica ao processo de expansão da infraestrutura de TI. Tomemos como exemplo uma empresa em crescimento.

A contratação de novos funcionários exige a compra de novos computadores e equipamentos relacionados. O investimento pode ser caro, mas a virtualização oferece uma alternativa interessante — em vez de novas máquinas, a empresa pode virtualizar seus servidores, criar máquinas independentes dentro deles e disponibilizá-las aos funcionários.

Em vez de trinta novos computadores, é possível providenciar thin clients — uma espécie de mini PC sem HD e processador próprios — que utilizará os recursos virtuais. O impacto é sentido até mesmo no espaço físico ocupado pela infraestrutura de TI, que tende a ser reduzido e gerar ainda mais economia.

Vale destacar também que o ciclo de renovação dos hardwares se torna mais longo. Os backups ganham em agilidade e a recuperação do sistema após um eventual desastre é mais eficiente.

Entretanto, é preciso ter em mente que a virtualização de servidores gera um uso maior de memória e espaço para armazenamento. Afinal, são diversos computadores nos quais é preciso ter cada software instalado.

O maior uso de memória RAM é o preço a se pagar pela independência dos servidores emulados. Afinal, se por um lado eles podem contar com sistemas operacionais distintos, por outro é preciso destinar uma parcela exclusiva de processamento para rodá-los.

Consequentemente, é preciso garantir uma gestão inteligente desses recursos. O monitoramento deve ser constante, assim como a configuração e atualização dos sistemas. Entretanto, se isso for feito com sucesso, é possível criar um ambiente virtual extremamente eficiente para a empresa.

A soma desses fatores resulta não só na redução de custos, mas na otimização dos recursos de TI. Seguindo a tendência trazida pela transformação digital, a tecnologia da empresa é aplicada como estratégia corporativa e impacta positivamente nas suas finanças.

A virtualização de servidores é uma tecnologia baseada em um software que possibilita a execução de vários sistemas operacionais diferentes como convidados dentro de um único host de servidor físico.

São as chamadas máquinas virtuais (VMs) que executam em uma imitação virtual do hardware de servidor.

É como se os recursos de um servidor físico, por exemplo, fossem divididos em diversos servidores virtuais que podem ser usados com finalidade diferentes. Essa tecnologia representa uma das maneiras mais eficientes de reduzir os custos de infraestrutura de TI. Isso porque a virtualização pode ser aplicada tanto em servidores quanto em redes, aplicações e Data Centers. Além disso, ela também gera mais eficiência e agilidade para a empresa do cliente, com um investimento que não ultrapassa os orçamentos de TI tradicionais. É importante entender que, com a virtualização de servidores, os processos são distribuídos entre um número menor de computadores, de forma que cada computador aproveita melhor sua capacidade total.

Com isso, torna-se desnecessário ter vários sub-servidores que utilizam apenas uma parte dos recursos dos equipamentos. Como funciona a virtualização de servidores? É possível separar a virtualização em 4 passos - que nem sempre são os mesmos para cada tipo de cliente - mas que costumam seguir o seguinte formato: 1- Aquisição do servidor A contratação ou aquisição de um servidor físico é a premissa básica para utilizar servidores virtuais. O CIO precisa analisar qual opção é mais rentável para a empresa dele: contratar um servidor físico terceirizado ou adquirir um servidor físico moderno, com maior capacidade e mais recursos (não que isso não seja possível no servidor terceirizado), além de maior controle por parte da equipe de TI.

Com a análise feita, é hora de escolher sua opção. No entanto, é importante saber que, no caso da virtualização, ao invés de ter vários servidores pequenos, o melhor investimento é contratar ou adquirir um servidor maior e mais moderno e que possua os diversos recursos necessários para atender à demanda de trabalho do cliente.

2- Implementação do servidor virtual No caso dos servidores físicos, o sistema operacional costuma ser mais básico, com capacidade focada em dividir o hardware em várias partes menores. Este sistema operacional é conhecido como hypervisor. Com isso, há mais espaço para as máquinas virtuais (VMs), já que o espaço do hypervisor é menor. Isso diminui a chance de erros com códigos no sistema operacional, além de facilitar as questões de segurança digital e o tempo de manutenção. 3- Criação das máquinas virtuais Depois de dividir o servidor físico principal em diversas máquinas virtuais, é preciso estruturá-las e ativá-las corretamente dentro do hypervisor.

Essas máquinas virtuais ficam armazenadas dentro do sistema operacional do servidor e outros componentes, como memória e processamento de CPU, ficam alocados sob demanda. Isso significa que as máquinas virtuais possuem capacidades diferentes, variando de acordo com as necessidades de cada equipamento, as atividades dela e dos profissionais que a utilizam.

Ou seja, uma VM pode ter mais capacidade de processamento do que espaço, a outra mais memória do que capacidade de processamento, de forma que as aplicações se dividam corretamente e com base nas necessidades de cada empresa e do que foi adquirido pelo servidor principal. Além disso, o cliente também pode utilizar a função de direcionamento de tráfego para priorizar as máquinas mais importantes.

4- Instalação dos sistemas e conexão com SAN Nesse ponto, é possível escolher o tipo de sistema que vai dentro de cada uma das máquinas do servidor virtual, já que cada uma delas possui recursos diferentes adequados ao tipo de trabalho que realizam.

Feito isso, o próximo passo é conectar os servidores virtuais a uma SAN (Storage Area Network), um local de armazenamento para compartilhamento entre servidores. Uma vez que os servidores estão em uma rede SAN, passa a ser mais fácil gerenciar a distribuição de desempenho para definir quais são as máquinas com maior prioridade. Esses pontos são necessários já que, em um ambiente corporativo, as máquinas virtuais costumam ter exigências de alta demanda.

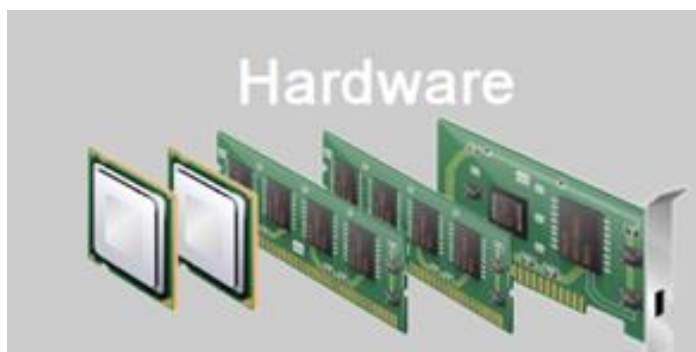
O grande ponto positivo da virtualização de servidores é que ela é indicada para qualquer tipo de empresa que necessite de mais de um servidor. Ela costuma não ter um alto custo, já que os investimentos iniciais com contratação de servidores, aquisição de equipamentos e softwares trazem um retorno no longo prazo com a diminuição dos gastos operacionais, a economia de energia e menor necessidade de manutenção. Além disso, ela também ajuda a economizar espaço físico, facilita o gerenciamento e manutenção dos equipamentos, reduzindo o downtime e otimizando a infraestrutura de tecnologia das empresas usuárias.

Virtualização de Servidores, como já explicamos nesse outro artigo, é uma forma de dividir os recursos de um servidor físico em vários servidores virtuais, também chamados de máquinas virtuais, de modo que possa executar diversos sistemas operacionais no mesmo hardware físico, isolados entre si.

Funciona da seguinte forma:

### **Aquisição do Servidor**

Um servidor físico vem com recursos físicos instalados de fábrica, entre eles: CPU, memória, discos, conexões de rede e conexões a SAN:



Um servidor moderno tem muito mais recursos do que os softwares são projetados para usar e é comum recursos como CPU e memória ficarem ociosos em alguns servidores, enquanto outros servidores tem gargalos.

Aí que entra a virtualização. No lugar de vários servidores de pequeno porte para diversas aplicações, é melhor investir em um servidor de maior porte e compartilhar os recursos entre os servidores virtuais sob demanda.

### **Instalação do Hypervisor**

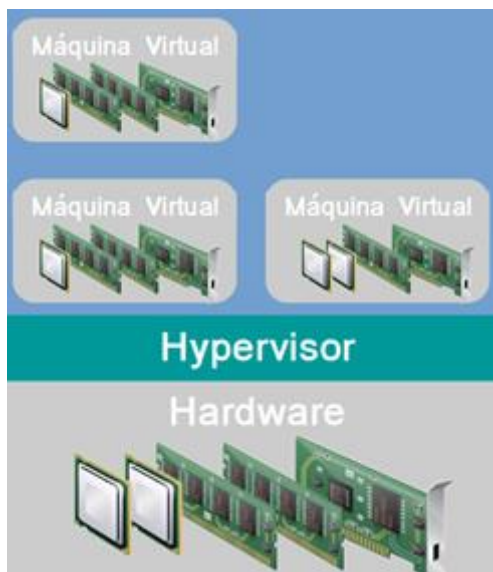
No servidor físico é instalado um sistema operacional básico, que possui a capacidade de dividir o hardware em pequenas partes. Esse sistema operacional é chamado de hypervisor.



Quanto menor for o espaço em disco e memória usada por esse hypervisor, mais recursos sobram para as máquinas virtuais, menor é a chance de ter problemas de código no hypervisor e menor são as paradas para manutenção do mesmo.

### **Criar as Máquinas Virtuais**

O hypervisor simula dentro de cada "fatia" do hardware um novo hardware, que são chamadas de máquinas virtuais. Os discos ficam armazenados em arquivos dentro do sistema operacional do hypervisor, enquanto que a CPU e memória são alocados sob demanda.

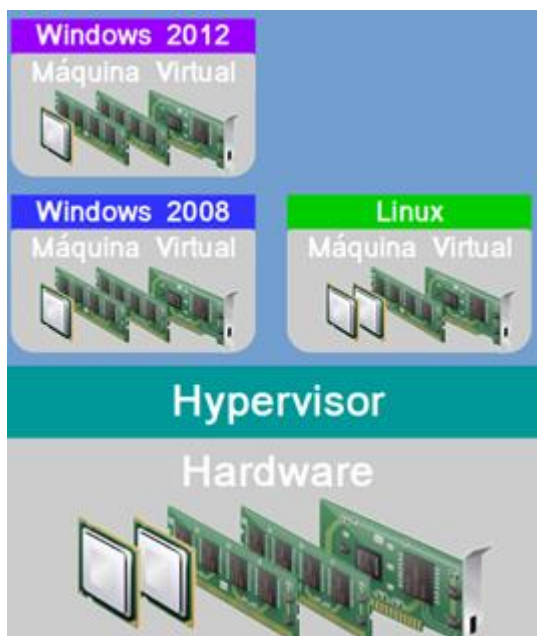


Cada máquina virtual pode ter capacidades diferentes de acordo com a necessidade, enquanto uma pode ter mais memória, a outra mais processador e a outra mais espaço em disco, cada uma dividindo uma fração do servidor original.

Na configuração de rede dos hypervisors mais avançados é possível dividir o tráfego e priorizar de acordo com a máquina virtual.

#### **Instalação do Sistema Operacional dentro das Máquinas Virtuais**

Dentro de cada máquina virtual pode ser instalado um sistema operacional diferente, de acordo com a necessidade. Cada um estará isolado dos demais, enxergará apenas os recursos que lhe foram dedicados e se comportará como se estivesse instalado em uma máquina física comum:



O hypervisor fica responsável por dividir os recursos entre as máquinas virtuais. Alguns recursos podem ser alocados em maior quantidade do que existe de verdade (over provisioning).

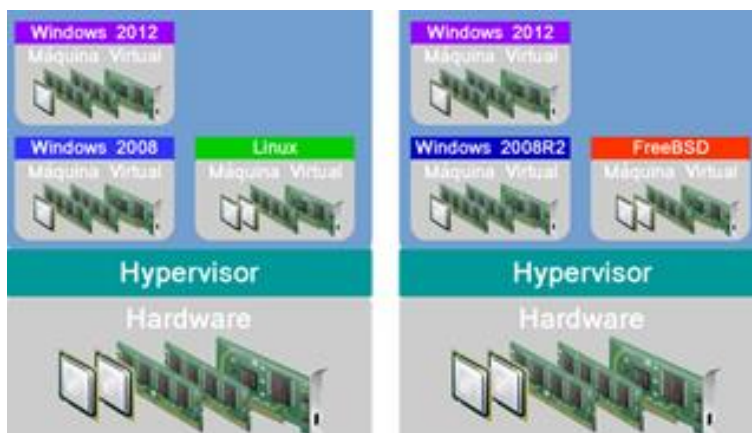
Por exemplo, um servidor com 10Gb de memória pode ter 7 máquinas virtuais com 2Gb de memória cada, o que totalizaria 14Gb, desde que essas não usem todo o recurso ao mesmo tempo. O hypervisor garante que nas situações de disputa, algumas máquinas virtuais tenham preferência de execução (maior prioridade).

### Conectar a uma SAN (Storage Area Network)

Em um ambiente empresarial com Alta Disponibilidade, as máquinas virtuais ficam armazenadas em uma SAN, que nada mais é do que um local de armazenamento (Storage) compartilhado entre os servidores. A SAN pode ser virtual, nesse caso é chamada de VSAN.

Uma SAN pode ser implementada também com um Storage dedicado, nesse caso, o equipamento possui toda a redundância necessária para garantir a alta disponibilidade do ambiente.

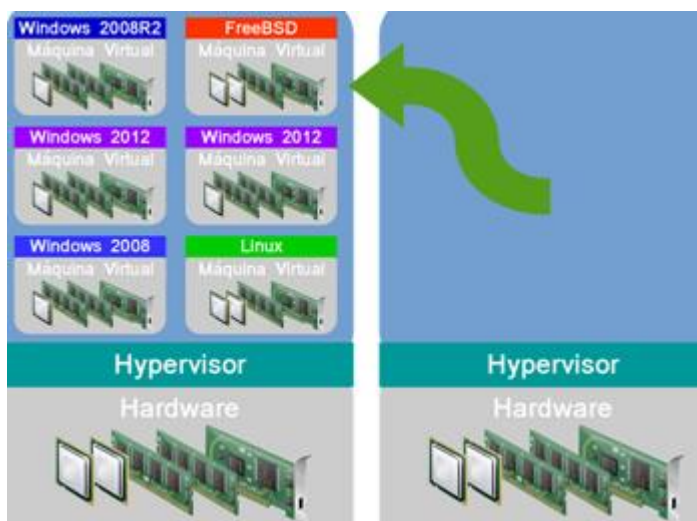
O fato de agregar os discos em um ponto único permite algumas facilidades de gerenciamento, além de poder distribuir a performance mais uniformemente e definir prioridades entre as máquinas virtuais.



Usando a SAN para manutenção programada de servidores

A SAN armazena os arquivos das máquinas virtuais, sendo assim, as máquinas virtuais podem ser desligadas de um servidor e ligadas em outro servidor sem necessidade de reinstalar o sistema operacional e aplicativos, ou de copiar arquivos entre os servidores físicos.

Também, dependendo da configuração e licenciamento, é possível migrar a máquina virtual entre um servidor e outro sem desligar. Esse recurso é chamado de vMotion, Live Migration ou XenMotion de acordo com o fabricante do hypervisor e permite a manutenção de um servidor físico sem downtime (parada) do ambiente.



### Crescendo o Ambiente

Quando o ambiente cresce, basta adicionar mais servidores ou espaço de Storage e todo o ambiente se beneficia. Pode-se crescer o ambiente em quantidade de máquinas virtuais, capacidade das máquinas virtuais, servidores físicos ou espaço no Storage. Cada servidor físico novo adiciona mais poder



## **Criptografia**

O que é Criptografia:

Criptografia é um mecanismo de segurança e privacidade que torna determinada comunicação (textos, imagens, vídeos e etc) ininteligível para quem não tem acesso aos códigos de “tradução” da mensagem.

Nas comunicações digitais, a criptografia auxilia na proteção de todos os conteúdos transmitidos entre duas ou mais fontes, evitando a interceptação por parte de cibercriminosos, hackers e espiões, por exemplo.

Atualmente, a maioria dos sites na internet utilizam comunicações criptografadas, principalmente em locais onde dados bancários, passwords e arquivos pessoais estejam armazenados.

Além de prevenir que pessoas não-autorizadas tenham acesso aos dados e informações trocadas na rede online, a criptografia também impede que backups sejam acessados por usuários indevidos.

Etimologicamente, o termo “criptografia” se originou a partir do grego, formado pela união dos elementos *kryptós*, que significa “secreto” ou “oculto”, e *graphía*, que quer dizer “escrita”. Assim, o significado literal de criptografia é “escrita secreta”.

No cotidiano, sistemas de criptografia são utilizadas pelos usuários de aplicativos e softwares de troca de mensagens instantâneas, como o Whatsapp, por exemplo.

### **Tipos de Criptografia**

Nas comunicações feitas através de dispositivos eletrônicos, o método mais utilizado de criptografia são as chamadas “chaves criptográficas”.

As chaves criptográficas consistem em conjuntos de algoritmos que codificam uma mensagem publicamente legível em um texto cifrado, ou seja, composto por valores secretos que só podem ser decifrados com o código de acesso correto.

Existem dois principais tipos de chaves criptográficas, estudadas através do ramo da Matemática conhecido por Criptologia: as simétricas e as assimétricas.

#### **Simétrica**

Também conhecida por “criptografia de chave única” ou “criptografia de chave privada”, este modelo utiliza apenas um conjunto de algoritmos responsáveis tanto pela cifragem de determinada operação, assim como a sua decifragem.

Neste caso, o pressuposto da confiabilidade entre os interlocutores deve ser total, visto que ambos partilham de uma única chave de criptografia, tanto para codificar como para decodificar uma mensagem, por exemplo.

#### **Assimétrica**

Também conhecido como “criptografia de chave pública”, este é um sistema de protocolos criptográficos que requer a formação de duas chaves, sendo uma privada (usada para decodificar) e a outra pública (utilizada para codificar e autenticar assinaturas digitais, por exemplo).

Com a criptografia assimétrica, qualquer pessoa pode enviar uma mensagem criptografada usando a chave pública, mas apenas os receptores com a chave privada conseguem decodificá-la. O segredo da informação consiste em manter em sigilo o código da chave privada, por exemplo.

Em linhas gerais, criptografia é o nome que se dá a técnicas que transformam informação inteligível em algo que um agente externo seja incapaz de compreender. De forma mais simples, a criptografia funciona como códigos: sem ela, um criminoso poderia interceptar a sua senha de e-mail durante o login.

Com a criptografia, caso ele intercepte seu acesso, mas não tenha a chave correta, verá apenas uma lista desordenada e aparentemente confusa de caracteres, que não leva a lugar nenhum.

A criptografia é um método de proteção e privacidade de dados muito importante e cada vez mais presente. Do ponto de vista prático para quem usa Internet e dispositivos que oferecem proteção criptográfica, há tipos ou termos, que é preciso conhecer: criptografia simétrica e assimétrica (ou de ponta a ponta).

### **Criptografia Simétrica**

O tipo de criptografia simétrica é o mais comum e pressupõe que uma mesma chave usada para ocultar informação precisa ser aplicada para revelá-la na outra ponta. É o tipo de criptografia usada na época da Segunda Guerra Mundial, por exemplo, e protagonista da história da invenção do computador, como conhecemos hoje.

### **Criptografia Assimétrica ou de Ponta-a-Ponta**

Atualmente, os dois protocolos mais usados para proteção de dados na Internet, o SSL (Secure Sockets Layer) e o TLS (Transport Layer Security) utilizam a criptografia simétrica para proteger os dados transmitidos e armazenados.

No entanto, a criptografia simétrica possui um desafio conceitual importante e impossível de ser resolvido. Como combinar uma chave secreta entre duas pessoas que querem se comunicar através da Internet de forma que ela não possa ser obtida por um invasor? Essa pergunta não teve solução até a década de 1970.

A solução foi dada pela criptografia assimétrica, na qual utiliza-se duas chaves distintas, mas que se complementam. Por essa propriedade, dá-se o nome de par de chaves, que é composto pela chave pública e pela chave privada. A chave pública é liberada para todos que desejam se comunicar com o emissor da chave enquanto a chave privada fica em poder de quem a emitiu.

O algoritmo de criptografia mais usado atualmente é o RSA, denominado pelas iniciais dos seus criadores, Ronald Rivest, Adi Shamir e Leonard Adleman. Uma desvantagem dos algoritmos de criptografia assimétrica existentes é o seu desempenho, que são mais lentos que os métodos simétricos.

Sendo assim, na prática, a criptografia assimétrica é utilizada para definir uma chave de sessão, que será usada na criptografia simétrica durante a comunicação. Esse é o funcionamento dos protocolos SSL e TLS, usados largamente na Internet.

Na criptografia assimétrica, as chaves públicas podem ser forjadas, fazendo com que o emissor não obtenha a chave pública correta do destinatário. Para solucionar esse problema, os engenheiros da Internet criaram a figura da Autoridade Certificadora, que funciona como um cartório, autenticando as chaves públicas das pessoas.

É essa autenticação da chave pública do seu banco, por exemplo, que faz o seu navegador exibir o selo de segurança, fazendo com que você saiba que o site é mesmo do banco e não de um criminoso.

Esses aplicativos de mensagens oferecem a criptografia de ponta-a-ponta, que pressupõe proteção de conteúdo das mensagens trocadas entre os usuários numa mecânica em que nem mesmo o próprio administrador dos aplicativos pode ler o conteúdo.

Ponta-a-ponta é um sinônimo para o tipo assimétrico, e no caso específico desses aplicativos, se refere ao fato de que cada usuário dentro dessas redes possui uma chave de criptografia específica que é combinada com a de seus contatos durante a troca de mensagens. Dessa forma, o conteúdo trocado entre duas pessoas pelos mensageiros só é visível por elas.

### **Criptografia no Computador e no Celular**

Ainda é muito comum associar o uso da criptografia diretamente com a proteção de dados na Internet: com a técnica, é muito mais difícil o criminoso descobrir seu login e senha de qualquer site e seus dados bancários são protegidos a cada compra.



Mas a criptografia tem aplicações que vão além disso. No computador, caso você decida criptografar seus dados, o windows ou macOS aplicarão uma chave criptográfica que protegerá todo o conteúdo armazenado na máquina de forma que só se torne visível por quem possua a chave, no caso o seu PIN, senha de usuário na máquina, ou qualquer tipo de autenticação biométrica oferecida pelo Windows, por exemplo.

Para celulares android e iPhone (iOS) a mesma coisa é válida. Ao criptografar os dados no aparelho, você os torna essencialmente inacessíveis a um invasor.

### **Níveis de Segurança**

A critpografia depende da aplicação e do nível de segurança exigido, mas em linhas gerais, uma critpografia de 128 bits é muito mais segura do que uma de 56 bits, por exemplo.

Uma chave de 56 bits oferece 72 quadrilhões de possibilidades de troca de caracteres para ocultar uma mensagem (parece absurdo, mas computadores já podem fazer bilhões de operações por segundo, então 56 bits pode não ser tão seguro assim se o hacker possuir um aplicativo que tenta milhões de alternativas para quebrar a critpografia a cada segundo).

Para comparar, uma chave de 128 bits tem 339,000,000,000,000,000,000,000,000,000 de possibilidades (arredondando, há uns trilhões a mais)

### **Criptografia**

A criptografia é uma técnica utilizada há anos que com o passar do tempo evoluiu a ponto de oferecer soluções eficazes no que diz respeito à segurança da informação. Hoje, ela é uma ferramenta de segurança amplamente utilizada nos meios de comunicação e consiste basicamente na transformação de determinado dado ou informação a fim de ocultar seu real significado.

Este artigo apresenta os conceitos sobre criptografia, seus tipos, aplicabilidade e como ela é empregada no .NET por meio do namespace System.Security.Cryptography. Ao final do artigo será desenvolvida uma aplicação para criptografar dados usando um algoritmo simétrico. Além disso, iremos criar uma DLL contendo a classe de criptografia implementada, que poderá ser reutilizada em outros projetos.

### **Em que Situação o Tema é Útil**

A criptografia pode ser utilizada em aplicações e ambientes cuja segurança das informações é algo relevante para o projeto, principalmente em sistemas WEB, onde o dado trafega em um meio público correndo um risco maior de ser interceptado, fato este que pode gerar prejuízos enormes para uma organização. O domínio das técnicas de criptografia não é algo complexo quando estamos trabalhando com o paradigma orientado a objetos, sendo essencial para a criação de aplicações seguras.

Há pouco tempo, quando a tecnologia ainda não era muito presente em nosso cotidiano, as informações e grande parte dos processos organizacionais eram geridos basicamente no papel, sendo armazenados em armários ou cofres protegidos por cadeados ou senhas.

Atualmente este paradigma mudou, pelo menos para uma parcela significativa da sociedade. As informações são processadas e armazenadas em meios digitais, criando uma forte dependência entre os sistemas de informação e as organizações. Com o advento da internet, os dados trafegam em meios públicos, podendo ser interceptado por qualquer um que esteja mal intencionado. Neste cenário, uma falha na segurança destes conteúdos pode acarretar em enormes prejuízos para uma corporação.

Então, o que fazer para garantir tal segurança? Existem diversos meios de proteção e um deles é o uso da criptografia. Ela não vai impedir que uma determinada informação seja interceptada, mas tem o objetivo de dificultar a compreensão do dado capturado. Mas como isso é feito? Há vários algoritmos de criptografia que cumprem este papel, cada um com suas particularidades, porém a ideia central é a mesma: modificar a informação de forma que apenas o destinatário consiga compreender a que foi transmitido.

Vale ressaltar que a criptografia não é aplicada apenas quando um dado é enviado de um local a outro, ela é utilizada também em dispositivos de armazenamento de dados (ex: discos rígidos, pen drives,

storages), que são alvos de ataques e roubos. Ou seja, de uma forma geral, a criptografia vai garantir a confidencialidade da informação. Nos próximos tópicos, veremos alguns conceitos relacionados a esta técnica.

### **Criptografia Simétrica**

A criptografia simétrica foi o primeiro tipo de criptografia criado. Os algoritmos que a utilizam têm como característica principal o uso de uma mesma chave criptográfica (Nota do DevMan 1) para criptografar ou descriptografar uma informação, por isso o adjetivo “simétrico” dá nome a esta técnica. Exemplificando um pouco este conceito, quando um emissor cifra uma mensagem com um algoritmo de criptografia simétrico, ele utiliza uma chave, que é representada por uma senha ou um conjunto de bits para codificar os dados. O receptor então faz uso do algoritmo para descriptografar a mensagem e aplica a mesma chave que foi utilizada pelo emissor para voltar à mensagem em sua forma original. Sem a mesma, não é possível decifrar a informação recebida.

#### **Nota do DevMan 1**

Chave criptográfica é um conjunto de caracteres formando uma sequência de bits que trabalhando em conjunto com um algoritmo de criptografia irão determinar o resultado final do processo de cifragem e decifragem da mensagem. O nível de segurança da codificação depende tanto do algoritmo quanto do tamanho da chave escolhida (total de bits que ela possui).

Uma forma muito utilizada por invasores para descobrir esta chave é utilizando a força bruta, onde são utilizadas inúmeras combinações de caracteres na tentativa de uma delas ser a chave do algoritmo. Veja na Figura 1 o processo de criptografia simétrica. Observe que a mesma chave é utilizada nos algoritmos para cifragem e decifragem do texto.

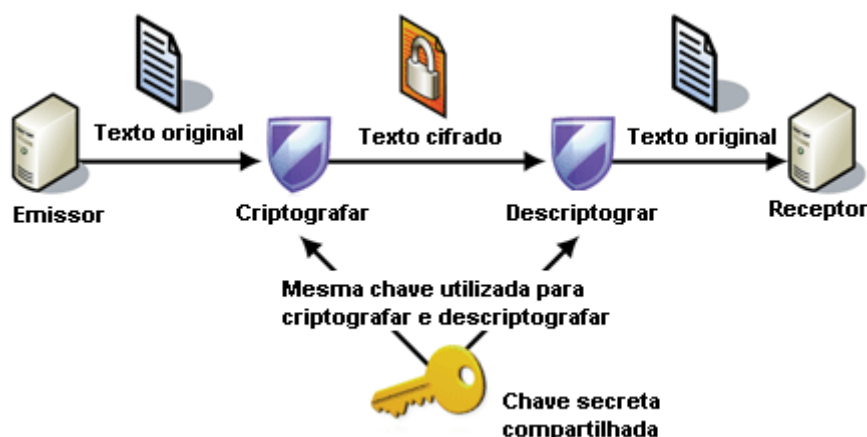


Figura 1. Processo de criptografia simétrica.

Como vantagens deste método podemos citar a simplicidade na sua implementação, uma vez que é utilizada uma única chave no processo de cifragem e decifragem do dado, além da velocidade deste processo em relação à criptografia assimétrica, que veremos nos próximos tópicos, possibilitando assim que uma grande quantidade de dados seja encriptada em pouco tempo.

Por outro lado este modelo de criptografia apresenta algumas falhas que estão relacionadas à geração e compartilhamento das chaves: no primeiro caso uma chave muito simples pode ser facilmente quebrada utilizando um algoritmo de força bruta. Já na segunda situação deve-se atentar para a forma como as chaves são compartilhadas entre os interessados na informação, a fim de evitar que a mesma seja obtida por um invasor.

Alguns algoritmos de criptografia simétrica bem conhecidos são: DES (Data Encryption Standart), Triple DES, AES (Advanced Encryption Standard), IDEA (International Data Encryption Algorithm), Blowfish, RC4.

### **Criptografia Assimétrica**

A criptografia assimétrica, também denominada como criptografia de chave pública, possui como característica básica o uso de duas chaves ao invés de uma, sendo elas:

Chave pública: Chave que pode ser distribuída para outros usuários.

Chave privada. Chave que deve ser mantida em segredo.

A criptografia diz respeito a conceitos e técnicas usadas para codificar uma informação, de tal forma que somente seu real destinatário e o emissor da mensagem possam acessá-la, com o objetivo de evitar que terceiros interceptem e entendam a mensagem.

Atualmente, as técnicas de criptografia mais conhecidas envolvem o conceito das chaves criptográficas, que são um conjunto de bits, baseados em um algoritmo capaz de interpretar a informação, ou seja, capaz de codificar e decodificar. Se a chave do receptor não for compatível com a do emissor, a informação então não será extraída.

O termo criptografia surgiu da fusão das palavras gregas "kryptós" e "gráphein", que significam "oculto" e "escrever", respectivamente. Trata-se de um conjunto de conceitos e técnicas que visa codificar uma informação de forma que somente o emissor e o receptor possam acessá-la, evitando que um intruso consiga interpretá-la. Para isso, uma série de técnicas são usadas e muitas outras surgem com o passar do tempo.

Na computação, as técnicas mais conhecidas envolvem o conceito de chaves, as chamadas chaves criptográficas. Trata-se de um conjunto de bits baseado em um determinado algoritmo capaz de codificar e de decodificar informações. Se o receptor da mensagem usar uma chave incompatível com a chave do emissor, não conseguirá extrair a informação.

Existem dois tipos de chave: a chave pública e a chave privada.

A chave pública é usada para codificar as informações, e a chave privada é usada para decodificar. Assim, na pública, todos têm acesso, mas para 'abrir' os dados da informação, que aparentemente são sem sentido, é preciso da chave privada, que só o emissor e receptor originais têm.

Atualmente, a criptografia pode ser considerada um método 100% seguro, ou seja, quem a utiliza para mandar e-mails e proteger seus arquivos, estará protegido contra fraudes e tentativas de invasão.

Os termos 'chave de 64 bits' e 'chave de 128 bits' são usados para expressar o tamanho da chave, assim, quanto mais bits forem utilizados, mais segura será essa criptografia. Um exemplo disso é se um algoritmo usa uma chave de 8 bits, por exemplo, apenas 256 chaves poderão ser utilizadas para decodificar essa informação, porque 2 elevado a 8 é igual a 256. Assim, um terceiro pode tentar gerar 256 tentativas de combinações e decodificar a mensagem, que mesmo sendo uma tarefa difícil, não é impossível. Por isso, quanto maior o número de bits, mais segura será a criptografia.

Existem dois tipos de chaves criptográficas, as chaves simétricas e as chaves assimétricas.

### **Chave Simétrica**

É um tipo de chave simples, que é usada para a codificação e decodificação. Entre os algoritmos que usam essa chave, estão:

DES (Data Encryption Standard): Faz uso de chaves de 56 bits, que corresponde à aproximadamente 72 quatrilhões de combinações. Mesmo sendo um número absurdamente alto, em 1997, conseguiram quebrar esse algoritmo através do método de 'tentativa e erro', em um desafio na internet.

RC (Ron's Code ou Rivest Cipher): É um algoritmo muito utilizado em e-mails e usa chaves de 8 a 1024 bits, além de possuir várias versões que se diferem uma das outras pelo tamanho das chaves.

EAS (Advanced Encryption Standard): Hoje em dia é um dos melhores e mais populares algoritmo de criptografia existente. Você pode definir o tamanho da chave como sendo de 128bits, 192bits ou 256bits.

IDEA (International Data Encryption Algorithm): É um algoritmo que usa chaves de 128 bits, parecido com o DES. Seu ponto forte é a fácil implementação de software.

As chaves simétricas não são totalmente seguras quando se trata de informações muito valiosas, principalmente pelo fato de que o emissor e o receptor têm que conhecer a mesma chave. Assim, a transmissão pode não ser segura e o conteúdo chegar a terceiros.

### Chave Assimétrica

A chave assimétrica utiliza duas chaves: a privada e a pública. Elas se resumem da seguinte forma: a chave pública para codificar e a chave privada para decodificar, levando-se em consideração que a chave privada é secreta.

Entre os algoritmos utilizados, estão:

RSA (Rivest, Shamir and Adleman): É um dos algoritmos de chave assimétrica mais utilizados, em que dois números primos (aqueles que só podem ser divididos por 1 e por eles mesmos) são multiplicados para a obtenção de um terceiro valor. Para isso, é preciso fazer fatoraçoão, que é descobrir os dois primeiros números a partir do terceiro, que é um cálculo trabalhoso. Assim, se números grandes forem utilizados, será praticamente impossível descobrir o código. A chave privada do RSA são os números que são multiplicados e a chave pública é o valor que será obtido.

O termo criptografia surgiu da fusão das palavras gregas "kryptós" e "gráphein", que significam "oculto" e "escrever", respectivamente. Trata-se de um conjunto de conceitos e técnicas que visa codificar uma informação de forma que somente o emissor e o receptor possam acessá-la, evitando que um intruso consiga interpretá-la.

Para isso, uma série de técnicas são usadas e muitas outras surgem com o passar do tempo.

Na computação, as técnicas mais conhecidas envolvem o conceito de chaves, as chamadas chaves criptográficas. Trata-se de um conjunto de bits baseado em um determinado algoritmo capaz de codificar e de decodificar informações. Se o receptor da mensagem usar uma chave incompatível com a chave do emissor, não conseguirá extrair a informação.

Os primeiros métodos criptográficos existentes usavam apenas um algoritmo de codificação. Assim, bastava que o receptor da informação conhecesse esse algoritmo para poder extraí-la. No entanto, se um intruso tivesse posse desse algoritmo, também poderia efetuar um processo de decifragem, caso capturasse os dados criptografados.

Há ainda outro problema: imagine que a pessoa A tivesse que enviar uma informação criptografada à pessoa B. Esta última teria que conhecer o algoritmo usado. Imagine agora que uma pessoa C também precisasse receber uma informação da pessoa A, porém a pessoa C não poderia descobrir qual é a informação a ser enviada à pessoa B. Se a pessoa C capturasse a informação enviada à pessoa B, também conseguiria decifrá-la, pois quando a pessoa A enviou sua informação, a pessoa C também teve que conhecer o algoritmo usado. Para a pessoa A evitar esse problema, a única solução seria utilizar um algoritmo diferente para cada receptor.

Com o uso de chaves, um emissor pode usar o mesmo algoritmo (o mesmo método) para vários receptores. Basta que cada um receba uma chave diferente. Além disso, caso um receptor perca ou exponha determinada chave, é possível trocá-la, mantendo-se o mesmo algoritmo.

Você já deve ter ouvido falar de chave de 64 bits, chave de 128 bits e assim por diante. Esses valores expressam o tamanho de uma determinada chave. Quanto mais bits forem utilizados, mais segura será a criptografia. Explica-se: caso um algoritmo use chaves de 8 bits, por exemplo, apenas 256 chaves poderão ser usadas na decodificação, pois 2 elevado a 8 é 256. Isso deixa claro que 8 bits é inseguro, pois até uma pessoa é capaz de gerar as 256 combinações (embora demore), imagine então um computador! Porém, se forem usados 128 ou mais bits para chaves (faça 2 elevado a 128 para ver o que acontece), teremos uma quantidade extremamente grande de combinações, deixando a informação criptografada bem mais segura.

### Chaves Simétricas e Assimétricas

Há dois tipos de chaves criptográficas: chaves simétricas e chaves assimétricas. Ambas são abordadas a seguir:

### Chave Simétrica

Esse é um tipo de chave mais simples, onde o emissor e o receptor fazem uso da mesma chave, isto é, uma única chave é usada na codificação e na decodificação da informação. Existem vários algoritmos que usam chaves simétricas, como o DES, o IDEA, e o RC:

- DES (Data Encryption Standard): criado pela IBM em 1977, faz uso de chaves de 56 bits. Isso corresponde a 72 quadrilhões de combinações. É um valor absurdamente alto, mas não para um computador potente. Em 1997, esse algoritmo foi quebrado por técnicas de "força bruta" (tentativa e erro) em um desafio promovido na internet;

- IDEA (International Data Encryption Algorithm): criado em 1991 por James Massey e Xuejia Lai, o IDEA é um algoritmo que faz uso de chaves de 128 bits e que tem uma estrutura semelhante ao DES. Sua implementação em software é mais fácil do que a implementação deste último;

- RC (Ron's Code ou Rivest Cipher): criado por Ron Rivest na empresa RSA Data Security, esse algoritmo é muito utilizado em e-mails e faz uso de chaves que vão de 8 a 1024 bits. Possui várias versões: RC2, RC4, RC5 e RC6. Essencialmente, cada versão difere da outra por trabalhar com chaves maiores.

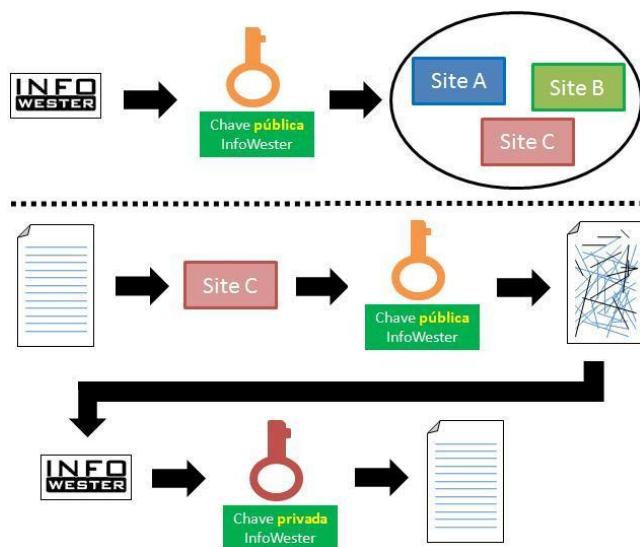
Há ainda outros algoritmos conhecidos, como o AES (Advanced Encryption Standard) - que é baseado no DES -, o 3DES, o Twofish e sua variante Blowfish, entre outros.

O uso de chaves simétricas tem algumas desvantagens, fazendo com que sua utilização não seja adequada em situações onde a informação é muito valiosa. Para começar, é necessário usar uma grande quantidade de chaves caso muitas pessoas ou entidades estejam envolvidas. Ainda, há o fato de que tanto o emissor quanto o receptor precisam conhecer a mesma chave. A transmissão dessa chave de um para o outro pode não ser tão segura e cair em "mãos erradas".

### Chave Assimétrica

Também conhecida como "chave pública", a chave assimétrica trabalha com duas chaves: uma denominada privada e outra denominada pública. Neste método, um emissor deve criar uma chave de codificação e enviá-la ao receptor. Essa é a chave pública. Uma outra chave deve ser criada para a decodificação. Esta, a chave privada, é secreta.

Para melhor compreensão, imagine o seguinte: O InfoWester criou uma chave pública e a enviou a vários outros sites. Quando qualquer desses sites quiser enviar uma informação criptografada ao InfoWester deverá utilizar a chave pública deste. Quando o InfoWester receber essa informação, apenas será possível extraí-la com o uso da chave privada, que só o InfoWester tem. Caso o InfoWester queira enviar uma informação criptografada a outro site, deverá obter uma chave pública fornecida por este.



Entre os algoritmos que usam chaves assimétricas, têm-se o RSA (o mais conhecido) e o Diffie-Hellman:

RSA (Rivest, Shamir and Adleman): criado em 1977 por Ron Rivest, Adi Shamir e Len Adleman nos laboratórios do MIT (Massachusetts Institute of Technology), é um dos algoritmos de chave assimétrica mais usados. Nele, números primos (número primo é aquele que só pode ser dividido por 1 e por ele mesmo) são utilizados da seguinte forma: dois números primos são multiplicados para se obter um terceiro valor.

Porém, descobrir os dois primeiros números a partir do terceiro (ou seja, fazer uma fatoração) é muito trabalhoso. Se dois números primos grandes (realmente grandes) forem usados na multiplicação, será necessário usar muito processamento para descobri-los, tornando essa tarefa praticamente inviável. Basicamente, a chave privada no RSA são os números multiplicados e a chave pública é o valor obtido;

ElGamal: criado por Taher ElGamal, esse algoritmo faz uso de um problema matemático conhecido por "logaritmo discreto" para se tornar seguro. Sua utilização é freqüente em

Existem ainda outros algoritmos, como o DSA (Digital Signature Algorithm), o Schnorr (praticamente usado apenas em assinaturas digitais) e Diffie-Hellman.

### **Certificação Digital**

Um recurso conhecido por certificação digital é muito utilizado com chaves públicas. Trata-se de um meio que permite, por exemplo, provar que um certo documento eletrônico foi mesmo emitido por uma determinada entidade ou pessoa. O receptor da informação usará a chave pública fornecida pelo emissor para se certificar da origem. Além disso, a chave fica integrada ao documento de forma que qualquer alteração por terceiros imediatamente a invalide.

Criptografia (do grego kryptos, oculto, e graphein, escrever) é o nome dado a um conjunto de regras que visa codificar a informação de maneira que só o emissor e o receptor consiga decifrá-la.

A troca de informações sigilosas é uma prática antiga, existente há centenas de anos, e que até bem pouco tempo era predominante em meio aos livros e documentos. O surgimento da internet e a facilidade que esta proporciona de transmitir dados de maneira precisa e extremamente rápida fez de tal prática um recurso essencial para permitir que apenas emissor e receptor obtenham acesso livre à informação tratada.

A criptografia segue quatro princípios básicos: confidencialidade, autenticação, integridade da informação e não repudiabilidade (ou seja, o remetente não pode negar o envio da informação). Apesar de ser recurso importante na transmissão de informações pela internet, a criptografia não é capaz de garantir total segurança, pois sempre existe alguém que consegue desenvolver uma maneira de "quebrar" o código. Assim, as técnicas são constantemente aperfeiçoadas e tantas outras são criadas, como por exemplo a "criptografia quântica".

A primeira técnica utilizava apenas um algoritmo de decodificação. Assim, bastava o receptor do algoritmo para decifrá-la, mas caso um intruso conhecesse esse mesmo algoritmo, ele poderia decifrar a informações se interceptasse os dados criptografados. Hoje, entre as técnicas mais conhecidas há o conceito de chaves, ou então chaves criptográficas, no qual um conjunto de bits baseado em um determinado algoritmo é capaz de codificar e de decodificar informações.

Há dois tipos de chaves, a simétrica e a assimétrica, ou chave pública. Caso o receptor da mensagem resolva usar uma chave incompatível com a chave do emissor, a informação não será compartilhada. Há ainda outros conceitos envolvidos na área da criptografia, como a Função Hashing, usada em assinaturas digitais para garantir integridade, e as aplicações, como a certificação digital.

O avanço das técnicas de invasão e interceptação de dados forçou a conseqüente evolução da criptografia, que adotou codificações de 256, 512 e até 1024 bits. Isso significa que são geradas 21024 combinações diferentes de chaves para cada mensagem enviada, sendo que apenas uma é correta, de conhecimento apenas do emissor e do receptor.

Com a intenção de ajudar na defesa da liberdade individual nos Estados Unidos e no mundo inteiro, Philip Zimmermann desenvolveu o PGP (Pretty Good Privacy) em 1991. Disponibilizado gratuitamente, o PGP se tornou um dos meios de criptografia mais conhecidos, principalmente na troca de e-mails, utilizando chaves assimétricas. O software pode realizar também um segundo tipo de criptografia através de uma "chave de sessão" método que representa um tipo de chave simétrica.



## **Análise E Projeto Oo & Uml 2.0**

Neste capítulo são apresentados os conceitos fundamentais do curso: modelo, UML, análise e projeto orientado a objetos, objeto e classes de objetos. Antes de entrar nos detalhes de UML, é preciso ater-se ao conceito de modelo.

Por exemplo, um projeto arquitetônico é feito segundo diversas perspectivas: do arquiteto, projeto arquitetônico em si, do engenheiro eletricista, projeto elétrico, do engenheiro civil, projeto hidráulico e estrutural. Construímos modelos de sistemas complexos para melhor compreendê-los.

Abstrair e refinar incrementalmente são palavras-chaves. Em certos momentos, o projetista deve focalizar na interação entre componentes do sistema sem se preocupar com seus detalhes internos de funcionamento, então ele abstrai estes detalhes. Em outros momentos, é preciso detalhar o comportamento dos componentes. Enfim projetar um sistema significa fazer modelos sob diferentes perspectivas e graus de abstração, representando-os por meio de uma notação precisa, refinando-os sucessivamente até transformá-los em algo próximo da implementação lembrando sempre de verificar se os requisitos são satisfeitos.

A modelagem visual (com auxílio de diagramas) ajuda a manter a consistência entre os artefatos (produtos) ligados ao desenvolvimento de um sistema: requisitos, projeto e implementação.

Resumidamente, a modelagem visual pode melhorar a capacidade de uma equipe a gerenciar a complexidade de software.

### **UML**

UML significa Unified Modeling Language ou Linguagem de Modelagem Unificada de projetos orientados a objetos. Como o próprio nome diz, UML é uma linguagem e não um método!

Por notação entende-se especificar, visualizar e documentar os elementos de um sistema OO. A UML é importante, pois:

- ◇ serve como linguagem para expressar decisões de projeto que não são óbvias ou que não podem ser deduzidas do código;
- ◇ provê uma semântica que permite capturar as decisões estratégicas e táticas;
- ◇ provê uma forma concreta o suficiente para a compreensão das pessoas e para ser manipulada pelas máquinas.
- ◇ É independente das linguagens de programação e dos métodos de desenvolvimento.

#### **Breve histórico**

Nos anos 90, conhecida como a época da “guerra dos métodos”, vários métodos coexistiam com notações muitas vezes conflitantes entre si. Dentre estes, os mais conhecidos eram:

- ◇ OMT (Object Modelling Technique) de Rumbaugh;
- ◇ Método de Booch;
- ◇ OOSE (Object Oriented Software Engineering) de Jacobson;

Inicialmente, Rumbaugh (OMT) e Booch fundiram seus métodos (e notações) resultando no Método Unificado em 1995 quando trabalhavam juntos na Rational Software (atualmente uma divisão da IBM). Jacobson juntou-se a eles mais tarde e seu método OOSE foi incorporado à nova metodologia (RUP).

Salienta-se que além do método, eles unificaram a notação de projeto e a chamaram UML. Então, UML representa a unificação das notações de Booch, OMT e Jacobson. Também agrega as idéias de inúmeros autores, tais como Harel e seu diagramas de estados, Shlaer-Mellor e o ciclo de vida dos objetos. Em suma, UML é uma tentativa de padronizar os artefatos de análise e projeto: modelos semânticos, sintaxe de notação e diagramas.



Na década de 90, surge uma organização importante no mundo dos objetos a OMG (Object Management Group), uma entidade sem fins lucrativos onde participam empresas e acadêmicos para definir padrões de tecnologias OO.

- ◇ Outubro de 1995: primeira versão rascunho, versão 0.8 draft.
- ◇ Julho de 1996: revisão devido ao ingresso de Jacobson, versão 0.9 draft.
- ◇ Parceiros UML (HP, IBM, Microsoft, Oracle e Rational Software) desenvolveram a versão 1.1 e a propuseram OMG
- ◇ A OMG aceita a proposta em novembro de 1997 e assume a responsabilidade de realizar manutenção é revisão da UML
- ◇ Em março de 2003, a OMG lançou a versão 1.5
- ◇ Em outubro de 2004, a OMG lançou versão 2.0 adotada<sup>1</sup>

### **Análise e Projeto Orientados a Objetos**

Há vários métodos de desenvolvimento de software. Na década de 80 houve preponderância dos métodos estruturados. Atualmente o paradigma OO é mais forte e, por isso, é importante diferenciar análise e projeto estruturado e orientado a objetos.

Vários autores participam da corrente de análise, projeto e programação estruturados: 1979 - Tom DeMarco: Análise estruturada (DEMARCO, 1989)

1982 - Gane e Sarson: Análise estruturada (GANE & SARSON, 1983)

1985 - Ward e Mellor: Análise estruturada para sistemas tempo real (WARD & MELLOR, 1986) 1989 - Yourdon: Análise estruturada moderna (YOURDON, 1990)

Na análise e projeto estruturados, o processo a ser informatizado é visto como um conjunto de funções com dados de entrada, processamento e dados de saída, ou seja, a ênfase está em funções que agem sobre dados. Estas funções podem ser decompostas em subfunções (decomposição funcional). As principais características são:

- ◇ preocupação com a modularidade e coesão;
- ◇ desenvolvimento em diferentes níveis de abstração (top-down).

Os principais diagramas empregados nas diversas metodologias estruturadas são:

- ◇ dicionários de dados, modelagem do fluxo de dados (DFD);
- ◇ modelos de dados: diagrama entidade e relacionamento (DER) e modelo entidade-relacionamento (MER).

### **Análise e Projeto Orientados a Objetos**

Análise, projeto e programação orientados a objetos: Coad e Yourdon (1979), Rumbaugh (1991), Grady Booch (1991), Jacobson(1992). Diferentemente da análise e projeto estruturados, na orientação a objetos o processo a ser informatizado é visto como um conjunto de objetos que interagem para realizar as funções. As vantagens do modelo OO são:

- ◇ maior grau de abstração;
- ◇ maior encapsulamento;
- ◇ modelos apoiados em conceitos do mundo real;
- ◇ reutilização (reusabilidade).

Neste curso, não é abordado o ciclo de vida de desenvolvimento de software que são inúmeros: cascata, iterativo, incremental, ágil, extremo e outros. No entanto, as fases clássicas do ciclo de vida são utilizadas (engenharia de requisitos, análise, projeto, implementação, testes, manutenção e operação).

### Objeto e Classe

Apresenta-se uma breve revisão de objeto e classes de objeto assim como a notação UML de ambos.

#### Objeto

É uma abstração que representa uma entidade do mundo real pode ser algo concreto (computador, carro) ou abstrato (transação bancária, histórico, taxa de juros). Um objeto num sistema possui três propriedades: estado, comportamento e identidade.

◇ Estado: definido pelo conjunto de propriedades do objeto (os atributos) e de suas relações com os outros objetos. É algo que muda com o tempo, por exemplo, um objeto turma pode estar no estado aberto ou fechado. Inicia no estado aberto e fecha quando 10 alunos fazem inscrição.

◇ Comportamento: como um objeto responde às solicitações dos outros e tudo mais o que um objeto é capaz de fazer. É implementado por um conjunto de operações. Ex. objeto turma pode ter operações acrescentar aluno ou suprimir aluno.

◇ Identidade: significa que cada objeto é único no sistema. Por exemplo, o objeto turma Tecno-OO manhã é diferente do objeto Tecno-OO tarde.

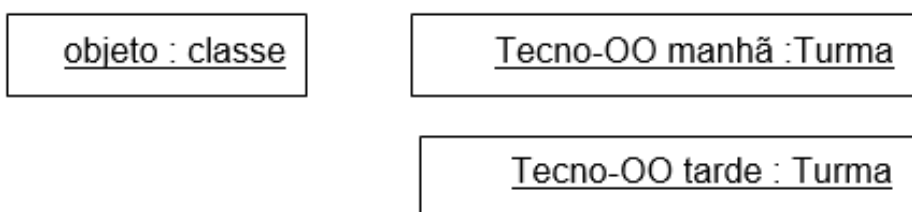


Figura 1. Notação de objeto em UML

#### Classe

Uma classe é uma descrição de um conjunto de objetos com propriedades, comportamento, relacionamentos e semântica comuns. Uma classe pode ser vista como um esqueleto/modelo para criar objetos.

Exemplo: classe turma

◇ Atributos: sala, horário

◇ Operações: obter local, adicionar estudante, obter horário

#### Dicas

◇ Classes devem encerrar uma só abstração do mundo real. Por exemplo, uma classe estudante contendo também o histórico do estudante não é uma boa solução. Melhor é dividi-la em duas classes: estudante e histórico.

◇ Utilizar substantivos para nomear as classes.

◇ Nas fases iniciais de desenvolvimento, pode-se suprimir os atributos e os métodos deixando somente os compartimentos.

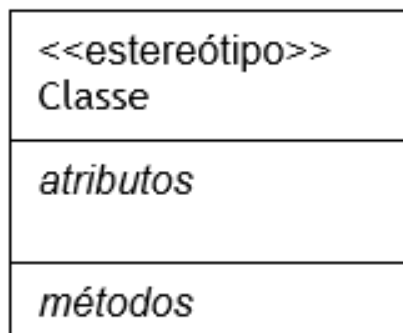


Figura 2. Notação UML para classe.

### Noção Geral de Análise e Projeto Oo

Objetivo deste capítulo é apresentar de maneira geral o método de análise e projeto de sistemas orientados a objetos utilizado durante o curso.

São descritas as fases principais do método e os diagramas UML mais indicados para cada uma delas. Este método é uma simplificação do RUP (Rational Unified Process).

#### Visão Geral

No curso, seguiremos o seguinte método:

Análise de requisitos: lista de requisitos funcionais e não-funcionais e diagrama de Casos de Uso.

Levantamento das classes candidatas: montar o diagrama de classes com um levantamento preliminar das classes, com atributos, métodos e relações (quando possível).

Estudo da interação entre objetos: diagramas de interação

Refinamento do diagrama de classes

Definição do comportamento de classes com estado através de máquinas de estados e diagrama de atividades

Modelo de implantação

Modelo de implementação

Codificação

Os passos de 3 a 5 se repetem até que se chegue próximo da implementação. Eventualmente é preciso revisar os requisitos e verificar as implicações das mudanças no projeto.

#### Análise De Requisitos

Consiste em determinar os serviços que o usuário espera do sistema e as condições (restrições) sob as quais o sistema será desenvolvido e operar. As necessidades do usuário podem ser muito variadas, o analista deve ser capaz de retirar os requisitos funcionais e não-funcionais destas necessidades:

◇ Funcionais: lista de serviços que o sistema deve oferecer ao usuário.

◇ Não funcionais: propriedades e características desejadas do sistema relativas à capacidade de armazenamento, tempo de resposta, configuração, uso (ex. uso intuitivo), confiabilidade, etc.

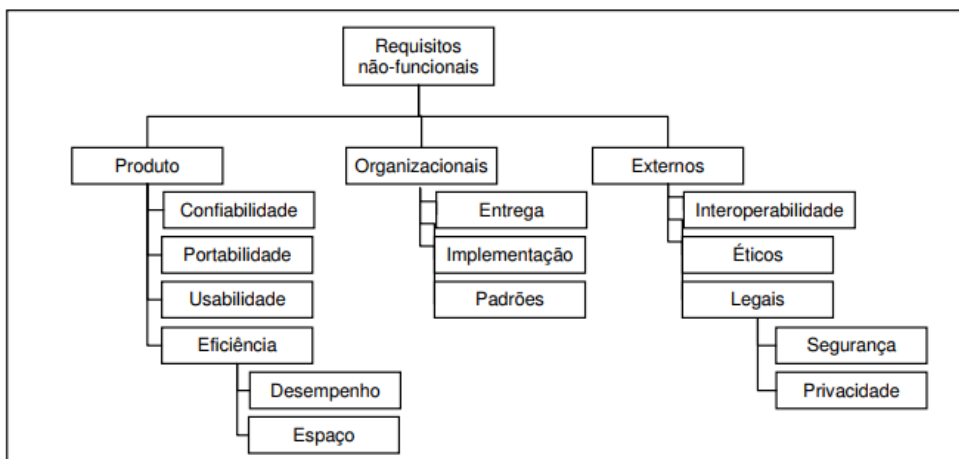


Figura 3: Taxonomia de requisitos não-funcionais (extraída de

**Papel dos Casos de Uso na Análise de Requisitos**

Casos de uso representam funcionalidades completas para o usuário e não, funcionalidades internas do sistema. Outro ponto importante é que o diagrama de casos de uso é um artefato de comunicação entre cliente, usuários e desenvolvedores. Por ser extremamente simples e, conseqüentemente, de fácil compreensão, incentiva a participação do cliente e usuários no processo de desenvolvimento. Também serve como um contrato entre a equipe/empresa desenvolvedora e o cliente.

**Casos de Uso**

A coleção de casos de uso representa todos os modos pelos quais o sistema pode ser utilizado pelos atores envolvidos. Um caso de uso é uma seqüência de ações realizadas colaborativamente pelos atores envolvidos e pelo sistema que produz um resultado significativo (com valor) para os atores. Um ator pode ser um usuário ou outro sistema.

Para uma calculadora de linha de comando cujo objetivo é executar expressões aritméticas (ex.  $-2 + 3*5$ ), o diagrama de casos da figura 4 pode ser considerado adequado.



Figura 4. Diagrama de casos de uso para a calculadora.

Diagrama de casos de uso é apenas um panorama visual das funcionalidades do sistema, é necessária uma descrição textual para detalhar os casos de uso. A tabela 1 ilustra esta documentação para o caso de uso resolver expressões aritméticas.

Tabela 1: documentação para o caso de uso resolver expressões aritméticas básicas.

Nome do caso de uso	Efetuar expressões aritméticas básicas
Descrição	Permite resolver expressões envolvendo números inteiros e reais e as operações básicas de soma, subtração, multiplicação e divisão sem parênteses.
Ator Envolvido	Usuário
Pré-condições	Sistema deve estar em execução aguardando por uma expressão
Pós-condições	Expressão aritmética resolvida ou abandono da expressão pelo usuário.

Fluxo básico	
Usuário	Sistema
	{Solicita expressão}
	Solicita a expressão. (A2)
Fornece a expressão	
	{Valida expressão}
	Avalia se a expressão é sintaticamente correta (A1)
	{Calcula valor}
	Calcula o valor da expressão.
	{Mostra o resultado}
	Informa o resultado da expressão.
	{Fim} Fim do caso de uso.
Fluxos alternativos	
A1: em {Valida expressão}	Se o usuário fornecer uma expressão sintaticamente incorreta, informá-lo sobre o erro e retomar o fluxo básico em {Solicita expressão}
A2: em {Solicita expressão}	usuário pode decidir encerrar o caso de uso sem fornecer uma expressão. Nesse caso retomar o fluxo básico em {Fim}

Portanto, a saída da fase de análise de requisitos é composta por:

- ◇ lista de requisitos funcionais e não-funcionais;
- ◇ diagrama de casos de uso e definições textuais dos casos.

### Análise e Projeto

Análise é a solução conceitual dada ao problema. Marca o início da definição informática, mas sem levar em conta detalhes da implementação tais como a linguagem a ser utilizada e o sistema gerenciador de banco de dados. Preocupa-se principalmente com as classes do domínio do problema e suas relações e também com os casos de uso.

Projeto é a solução informática dada ao problema. A separação entre análise e projeto é tênue, pois o projeto acaba sendo o resultado de sucessivos refinamentos do modelo conceitual de análise.

### Diagramas de Interação

Há vários tipos de diagramas de interação na UML. Exemplifica-se o uso do diagrama de seqüência cuja utilidade é estudar as interações entre os objetos com o objetivo de refinar o diagrama de classes, identificando relações entre classes, seus métodos e atributos. A figura 5 mostra um cenário onde o usuário fornece uma expressão aritmética sintaticamente correta.

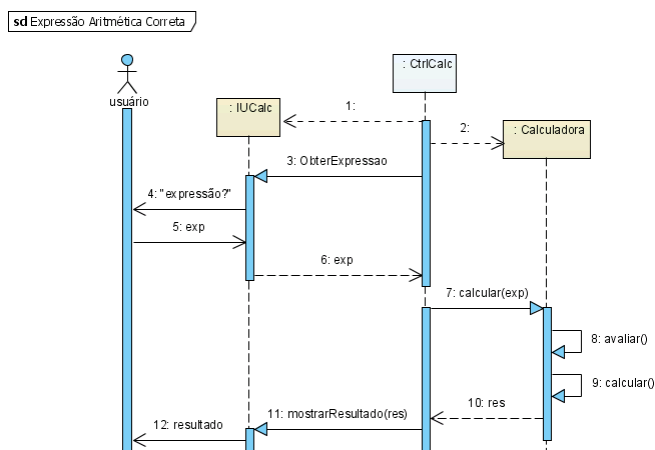


Figura 5. Diagrama de seqüência para a calculadora.

Refinamento do Diagrama de Classes

A partir das informações do diagrama de seqüência, é possível:

- ◇ identificar métodos associados às classes. Por exemplo, a classe IUCalculadora deve ter um método mostrarResultado(<resultado>).
- ◇ identificar as relações entre classes. Pelo diagrama de seqüência, fica clara a existência de uma relação de dependência entre a classe de controle e as duas outras como ilustra a figura 6.

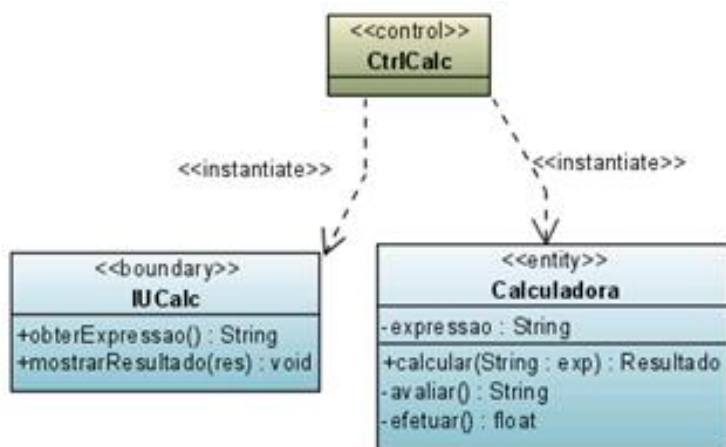


Figura 6. Diagrama de classes.

Definir o Comportamento das Classes

Nem todas as classes de um sistema possuem mais de um estado. Para as classes mais complexas, podemos especificar seus comportamentos utilizando máquinas de estado. A figura 7 mostra uma possível máquina de estados para a calculadora.

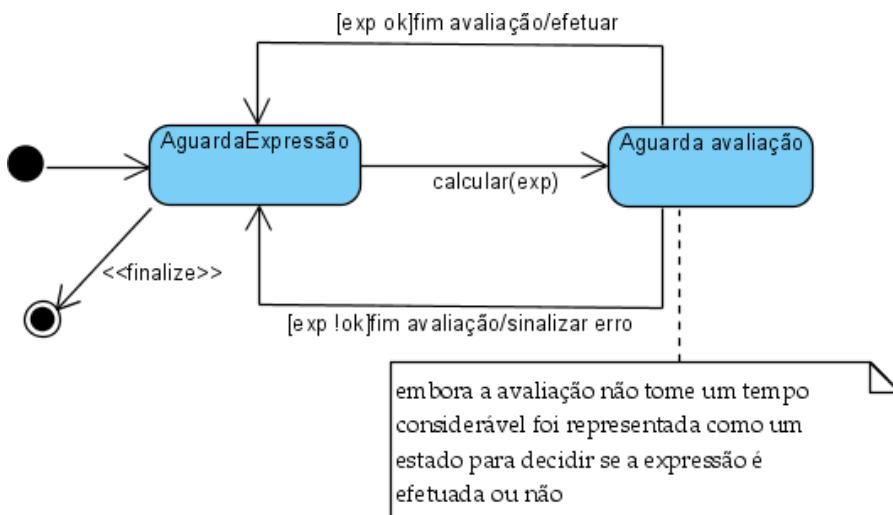


Figura 7. Máquina de estados para calculadora.

Os métodos de uma classe podem ainda ser detalhados por meio de um diagrama de atividades como ilustra figura 8.

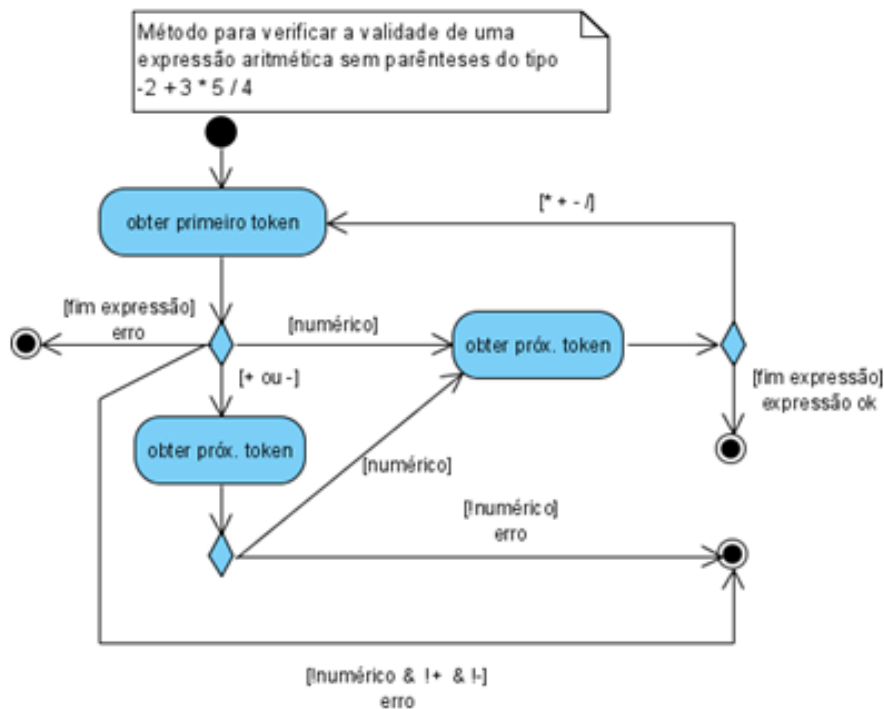


Figura 8: diagrama de atividades - detalhe de um método para verificar a sintaxe de expressão aritmética.

### Implantação

diagrama de implantação representa as necessidades de hardware e software básico (ex. servidores). Para tornar o diagrama mais realista, a figura 9 supõe que a calculadora é um serviço ofertado por um servidor de aplicações Web.

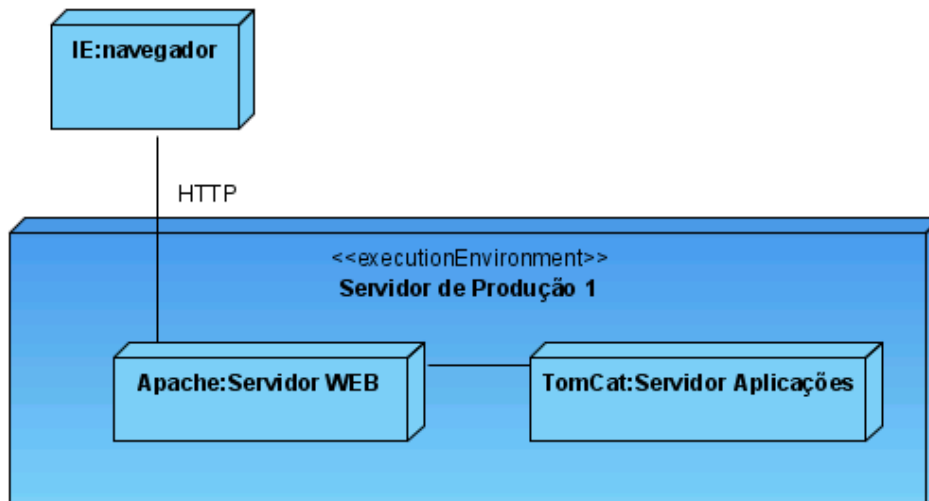


Figura 9: diagrama de implantação (deployment).

### Componentes do Sistema

objetivo é documentar os componentes do sistema (fontes, bibliotecas) e suas relações. A figura 10 ilustra o diagrama de componentes para a calculadora e mostra a dependência entre seus componentes.

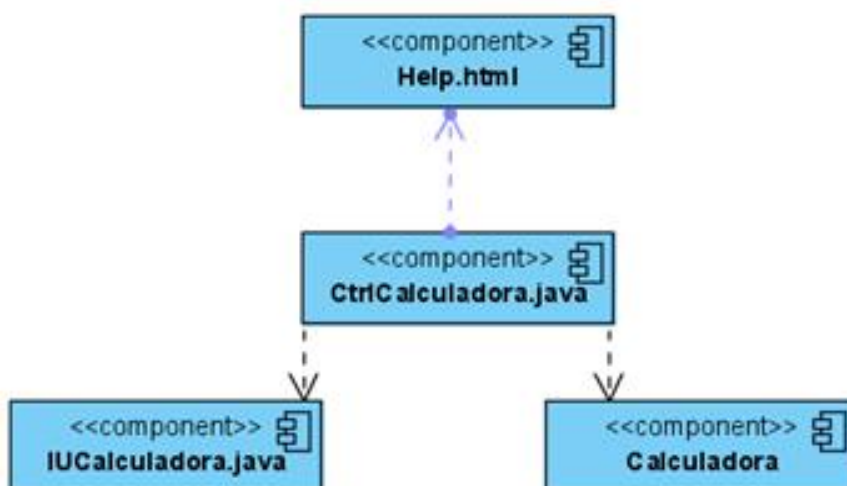


Figura 10: diagrama de componentes para a calculadora.

### Modelagem Estrutural E Comportamental

Com esta rápida introdução à UML, é possível observar que alguns diagramas são mais indicados para modelar a estrutura do sistema e outros, o comportamento. A figura 11 mostra esta divisão.

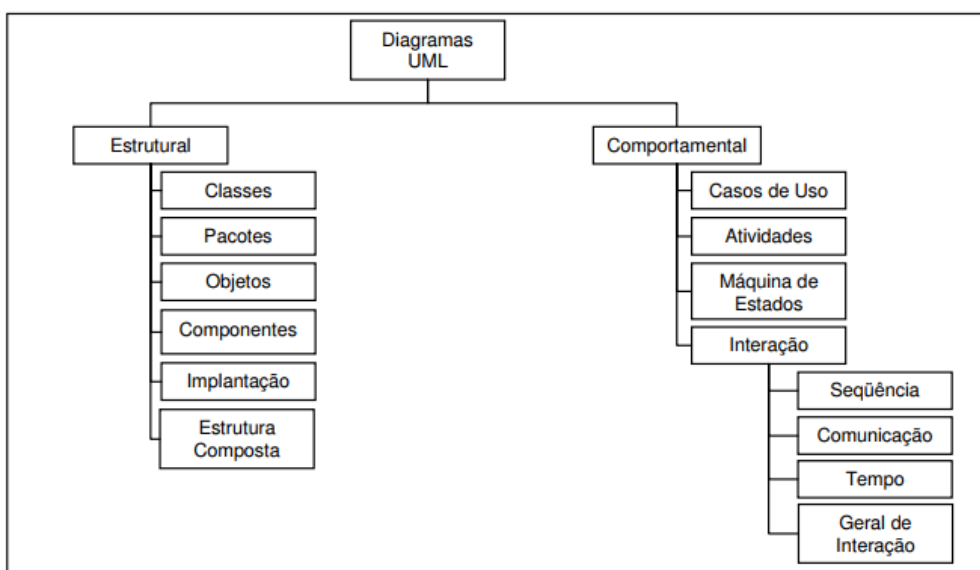


Figura 11. Diagramas estruturais e comportamentais da UML.

Segue uma breve descrição dos diagramas UML ainda não descritos neste documento:

**Pacotes:** representa uma coleção de classes que juntas formam uma unidade. Também pode servir para agrupar um conjunto de casos de uso com similaridades funcionais. Os pacotes podem apresentar relações, por exemplo, um pacote de classes pode depender de outro para executar suas funções.

**Objetos:** É um instantâneo da execução do sistema, retrata os objetos instanciados e suas relações em um dado momento.

**Componentes:** segundo a definição de (OMG, 2007, pg. 146), um componente é um módulo ou parte de um sistema que encapsula seu conteúdo (comportamento e dados). Um componente exhibe seu comportamento através de interfaces bem definidas e pode depender de outros componentes.

**Deployment (Implantação ou Distribuição):** para representar a arquitetura física do sistema, ou seja, para representar as relações entre os componentes (artefatos) e os locais de execução (nodos: máquinas ou sistemas servidores).



Estrutura Composta: “descreve a estrutura interna de uma classe ou componente, detalhando as partes internas que o compõe como estas se comunicam e colaboram entre si” (Guedes, 2004).

Atividades: pode ser utilizado para diversos fins, um deles é a especificação mais detalhada de métodos complexos ou do encadeamento dos casos de uso.

Interação.Comunicação: mostra as interações entre uma coleção de objetos sem a linha do tempo (pode ser obtido do diagrama de seqüência e vice-versa).

Interação.Tempo: mostra o estado de um objeto ao longo do tempo.

Interação.Geral: é a fusão do diagrama de atividades com o de seqüência. Permite fazer referência a diagramas de seqüência e combiná-los com controle de fluxo (ex. pontos de decisão, forks e joins).

### Modelo de Casos de Uso

Modelo de casos de uso (que é mais do que o diagrama) é o principal resultado da fase de análise de requisitos. Diagramas de casos de uso são utilizados para representar de forma panorâmica os requisitos funcionais de um sistema do ponto de vista do usuário. Cabe salientar que há outras utilizações possíveis para o diagrama de casos de uso, tal como modelagem do negócio, porém, o foco nesta seção está na representação de requisitos funcionais do usuário.

#### Definição

É um diagrama utilizado na análise de requisitos com objetivos claros:

Compreender o problema.

Delimitar o sistema (quem está no entorno).

Definir as funcionalidades oferecidas ao usuário (não há preocupação com a implementação).

Os elementos básicos de um diagrama de casos de uso são:

- ◇ atores,
- ◇ casos de uso e
- ◇ relações entre os mesmos.

#### ATORES

◇ Representam papéis desempenhados por usuários ou qualquer outra entidade externa ao sistema (ex. hardware, outros sistemas)

◇ Podem iniciar casos de uso

◇ Podem prover e/ou receber informações dos casos de uso



Figura 12: Notação UML para ator.

#### Como encontrar atores de um sistema

- ◇ Examinar o problema procurando por pessoas ou sistemas do entorno.
- ◇ Quais as pessoas ou departamentos interessados num determinado requisito funcional?

- ◇ Quem irá suprir o sistema com informações e quem irá receber informações do sistema?
- ◇ Quais os recursos externos utilizados pelo sistema?
- ◇ Uma pessoa desempenha diferentes papéis?
- ◇ O sistema interage com outros sistemas já existentes?

Há fluxos primários ou básicos (fluxo normal de eventos) e alternativos (o que fazer se...). Para descrevê-los, é possível se inspirar na situação em que uma pessoa explica um caminho à outra. Primeiro, o fluxo básico é explicado, depois, as alternativas.

Para ir ao churrasco, pegue a BR116 na direção São Paulo. Logo após o clube Santa Mônica, tem um retorno por baixo da pista. Faça o retorno e continue reto (não retorne à BR). Continue nesta estradinha asfaltada por 1 km, no entroncamento pegue a estrada de terra à direita, ande cerca de 500m, você verá um grande eucalipto e uma araucária. A entrada da chácara é entre os dois. Não se esqueça de trazer o pinhão. // *primário*

Se estiver chovendo muito, os 500m na terra podem ser bem difíceis porque o barro é mole. Neste caso, siga reto no entroncamento (ao invés de virar à direita) e na próxima a direita pegue a rua de paralelepípedos. Ande cerca de 1 km e depois vire na segunda a direita que vai desembocar na frente da chácara. // *alternativo 1*

Se você for comprar o pinhão no caminho, logo depois de fazer o retorno da BR tem uma venda. Se estiver fechada, um pouco mais a frente, tem um senhor da chácara Pinhais que também vende. Se não encontrar pinhão, não tem problema. // *alternativo 2*

Fluxos documentam as responsabilidades, ou seja, como as responsabilidades especificadas nos casos de uso são divididas entre sistema e atores. No desenrolar do projeto, as responsabilidades atribuídas ao sistema devem ser distribuídas entre os objetos que compõem o sistema.

Nas fases iniciais de análise é bom concentrar-se nos fluxos básicos (cerca de 80% do tempo de execução de um sistema é ocupado pelos casos primários) e somente identificar os casos secundários.

### Fluxo Básico

Um fluxo básico representa o que ocorre normalmente quando o caso de uso é executado. A descrição do fluxo básico deve conter (Bittner e Spencer, 2003):

- ◇ ator e o evento que o mesmo dispara para iniciar o caso;
- ◇ a interação normal (sem tratamento de exceções) entre ator e sistema;
- ◇ descrição de como o caso termina.

Exemplo: considere um sistema onde o cliente realiza compras on-line num site utilizando um carrinho de compras virtual. O projetista do sistema previu um caso chamado buscar produtos e fazer pedido especificado pelo fluxo básico seguinte - extraído de (Bittner e Spencer, 2003):

NOME: Buscar produtos e fazer pedido  
DESCRIÇÃO: Este caso descreve como um cliente usa o sistema para visualizar e comprar produtos disponíveis. Para encontrar um produto, o cliente pode pesquisar o catálogo por tipo de produto, fabricante ou por palavras-chaves.  
PRÉ-CONDIÇÕES: o cliente está logado no sistema.  
PÓS-CONDIÇÕES: o cliente realiza uma compra ou não.  
FLUXO BÁSICO DE EVENTOS

O caso de uso inicia quando o ator cliente escolhe a opção de consultar o catálogo de produtos.

{Mostrar catálogo de produtos}

O sistema mostra os produtos oferecidos, ressaltando os produtos cujas categorias constam no perfil do cliente.

{Escolher produto}

O cliente escolhe um produto a ser comprado e define a quantidade desejada.

Para cada produto selecionado disponível em estoque, o sistema registra o código do produto e a quantidade solicitada reservando-a no estoque e adiciona-o ao carrinho de compras.

{Produto esgotado}

Os passos 3 e 4 se repetem até que o cliente decida efetuar a compra dos produtos.

{Processar pedido}

10.O sistema pergunta ao cliente se deseja fornecer informações sobre o pagamento.

11.O sistema utiliza um protocolo seguro para obter as informações de pagamento do cliente.

12.Executar subfluxo validar informações de pagamento

13.{Informações de pagamento não válidas}

14.O sistema pergunta ao cliente se deseja fornecer informações sobre o envio das mercadorias.

15.O sistema utiliza um protocolo seguro para obter as informações de envio.

16.Executar subfluxo validar informações de envio.

17.{Informações de envio não válidas}

18.Executar subfluxo efetuar transação financeira.

19.O sistema pergunta ao cliente se deseja comprar mais produtos.

20.Se o cliente desejar comprar mais produtos, retomar o caso no ponto {Mostrar catálogo de produtos}, se não o caso termina.

No fluxo básico acima, pode-se notar a existência de vários elementos que serão descritos a seguir: subfluxo, pontos de extensão e fluxos alternativos.

### Subfluxo

Um fluxo de eventos pode ser decomposto em subfluxos para melhorar a legibilidade. No entanto, é interessante evitar muitas decomposições, pois o fluxo ficará muito fragmentado e seu entendimento dificultado. Um subfluxo deve ser atômico, isto é, ou é executado na sua totalidade ou não é executado. Para referenciar um subfluxo a partir de outro fluxo usar a notação: executar <nome subfluxo>. No exemplo buscar produtos e fazer pedido, os subfluxos seguintes são encontrados:

◇ S1 Validar informações de pagamento;

◇ S2 Validar informações de envio;

◇ S3 Efetuar transação financeira.

Estes subfluxos podem ser detalhados da mesma maneira que um fluxo básico, porém deve-se evitar muitas decomposições sob o risco de perder a visão geral do caso de uso.

Pontos de extensão

São pontos precisos num fluxo de eventos que servem para inserir comportamentos adicionais. Pontos de extensão podem ser privados ou públicos. São privados se visíveis somente dentro do caso onde foram definidos ou públicos se visíveis nos casos que estendem o caso onde foram definidos.

No exemplo Buscar produtos e fazer pedido, os pontos de extensão seguintes são encontrados:

- ◇ {Mostrar catálogo de produtos}
- ◇ {Escolher produto}
- ◇ {Produto esgotado}
- ◇ {Processar pedido}
- ◇ {Informações de pagamento não válidas}
- ◇ {Informações de envio não válidas}

Um ponto de extensão pode definir

- ◇ uma localização único dentro do fluxo, por exemplo, {Mostrar catálogo de produtos}, {Escolher produtos} e {Processar pedido};
- ◇ um conjunto de localizações que representam um certo estado do caso de uso, por exemplo, {Produto esgotado} que poderia aparecer em vários pontos do fluxo de eventos;
- ◇ uma região entre dois pontos de extensão, por exemplo, {Escolher produtos} e {Processar pedido}.

#### Fluxo Alternativo

Um fluxo alternativo apresenta um comportamento opcional, de outra forma, que não é parte do comportamento normal de um caso de uso. Fluxos alternativos são utilizados para representar tratamento de exceções ou um comportamento alternativo complexo que tornaria o fluxo básico muito longo ou de difícil compreensão.

Fluxos alternativos sempre são dependentes da existência de uma condição que ocorre em um ponto de extensão de outro fluxo de eventos. Há três tipos de fluxos alternativos:

Específico: iniciam num ponto de extensão.

Regional: podem ocorrer entre dois pontos de extensão.

Geral: podem ocorrer em qualquer ponto do caso de uso.

No exemplo Buscar produtos e fazer pedido, os fluxos alternativos seguintes são encontrados: Tratar produto esgotado (específico) e pesquisar por palavras-chaves (regional).

#### *A2 Tratar produto esgotado*

Em {**Produto esgotado**} quando não há a quantidade requisitada do produto em estoque.

1. O sistema informa que o pedido não pode ser completamente satisfeito.
2. // a descrição deste fluxo continua com oferta de quantidades e produtos alternativos ao cliente
3. O fluxo de eventos básico é retomado no ponto onde foi interrompido.

#### *A1 Pesquisar por palavras-chaves*

Entre {**Mostrar catálogo de produtos**} e {**Escolher produto**} quando o cliente escolhe realizar uma pesquisa por palavras-chaves.

1. O sistema pergunta ao cliente pelos critérios de busca do produto.
2. O cliente fornece os critérios de busca de produto.
3. // a descrição deste fluxo continua
4. O fluxo de eventos básico é retomado em {**Escolher produto**}.

Não há fluxo geral para o exemplo, mas poderia ser definido da seguinte maneira: em qualquer ponto do caso de uso Buscar produtos e fazer pedido...

Por que representar um fluxo alternativo separadamente do fluxo básico se é possível representá-lo com um se (if) no fluxo básico?

◇ Fluxos alternativos são opcionais e descrevem comportamentos que estão fora do comportamento normal esperado.

◇ Nem todos os fluxos alternativos representam funcionalidades essenciais, muitos deles podem não ser necessários, podem ser muito caros ou não prover funcionalidades importantes o suficiente para dispêndio de esforços de desenvolvimento.

◇ Fluxos alternativos permitem adicionar funcionalidade ao fluxo básico de maneira incremental ou remover funcionalidade à medida que tempo e dinheiro se esgotam.

Por exemplo, qual a importância de realizar pesquisas por palavras-chaves no exemplo em uso? Se for apenas uma das alternativas de busca não inviabiliza a funcionalidade do fluxo básico como um todo.

Agora se alguém perguntar qual a importância do fluxo básico buscar produtos e realizar pedido, é fácil ver que não pode ser deixado de fora.

### Diagrama de Atividade

Um diagrama de atividade pode ser empregado para representar os fluxos de eventos de um caso de uso. Sua utilização não suprime a descrição textual, pelo contrário, ele deve ser visto como uma ilustração simplificada da descrição textual. Se todos os detalhes da descrição textual forem colocados no diagrama, este ficará extremamente poluído e perderá sua utilidade: tornar o caso de uso mais compreensível aos leitores.

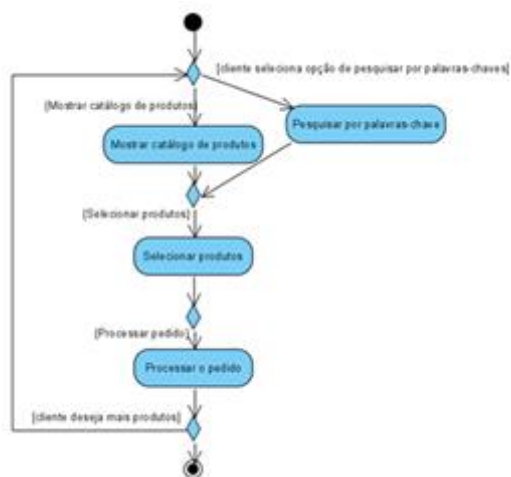


Figura 13: Exemplo de diagrama de atividades.

### Cenários

Cenários são instâncias de execução dos casos de uso. Os fluxos alternativos representam as possibilidades de execução de um caso de uso.

No exemplo buscar produtos e realizar pedido, o fluxo alternativo pesquisar produtos por palavras-chaves é uma alternativa à simples visualização do catálogo de produtos, logo há pelo menos dois caminhos possíveis de execução. Um cenário representa um desses caminhos (figura 14).

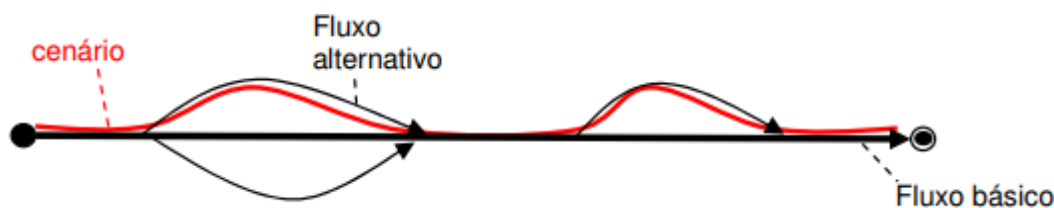


Figura 14: Representação esquemática de um cenário.

Cenários são importantes para definir casos de teste e para desenvolvedores pensarem sobre como o sistema será utilizado. Podem ser documentada adicionando-se informação às descrições dos casos de uso ou como parte da descrição dos testes. Não há necessidade de descrevê-los detalhadamente, basta nomeá-los e descrever o caminho a ser percorrido (por exemplo, fluxo básico, fluxo alternativo a1, fluxo básico).

### Realizações de Casos de Uso

Um caso de uso pode ser realizado (projetado e implementado) de diferentes modos. Em UML há uma representação para realização de caso de uso como ilustra a figura 15. O intuito dessa representação é fazer uma ponte entre as descrições do sistema utilizadas pelas pessoas envolvidas na sua construção, mas que não participam do desenvolvimento em si, e as descrições do sistema utilizadas pela equipe de desenvolvimento.

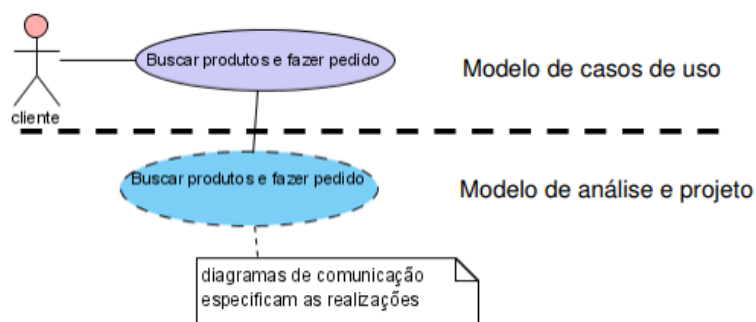


Figura 15: Realização de um caso de uso.

Diagramas de interação podem ser associados às realizações de casos de uso para especificar o fluxo de informações entre objetos que concretizam o caso. Porém, a representação de realização de caso não é muito utilizada.

### Relações

Há vários tipos de relações possíveis num diagrama de casos de uso, porém é importante salientar que as relações:

não representam a ordem de execução dos casos;

devem melhorar a compreensão do que o sistema deve fazer (e não como projetá-lo). Em seguida, apresentam-se as relações mais comuns.

### Associação

Associação é o tipo mais comum de relação. Pode ser utilizada entre dois atores ou entre um ator e um caso de uso. São representadas por uma linha cheia, com ou sem direção.

### Ator x Ator

Relações associativas podem conectar atores para representar comunicação entre atores. A relação pode receber um nome que identifica o conteúdo da mensagem, documento ou objeto que trafega entre os atores. A figura 16 mostra uma associação entre o ator usuário de biblioteca que passa o

livro ao atendente que realiza o empréstimo ou a devolução. Como não há flechas, assume-se que o atendente devolve algo ao usuário da biblioteca, provavelmente um comprovante não representado no diagrama. Não é recomendável colocar este tipo de relação no diagrama de casos de uso.

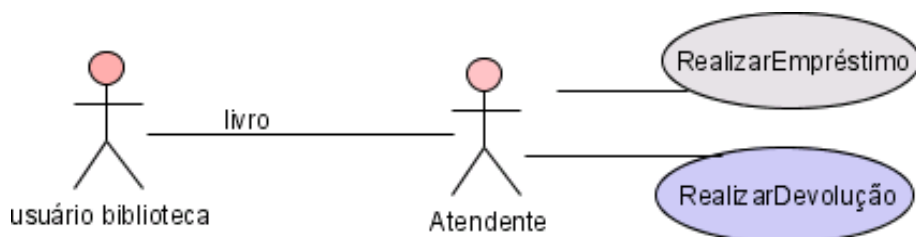


Figura 16: Exemplo de associação entre atores.

#### Ator x Caso

Há vários usos para associações entre atores e casos de uso:

Indica quem inicia a comunicação, o ator ou o caso de uso (sistema);

Indica o fluxo de informações, ou seja, quem fornece informações a quem.

Para documentar a escolha, pode-se atribuir um nome à associação. Na figura 16, há uma associação entre o atendente e o caso de uso realizar empréstimo. Observar que é bidirecional, portanto o atendente inicia a execução do caso, fornece e recebe informações do mesmo.

Associações unidirecionais deixam os diagramas mais claros, embora não sejam obrigatórias. Por exemplo, a figura 17 ilustra um mesmo diagrama que representa os requisitos de um sistema de telefonia com relações bidirecionais e unidirecionais. No diagrama superior, pode-se deduzir que o emissor inicia a chamada telefônica e, no inferior, esta informação está explícita.

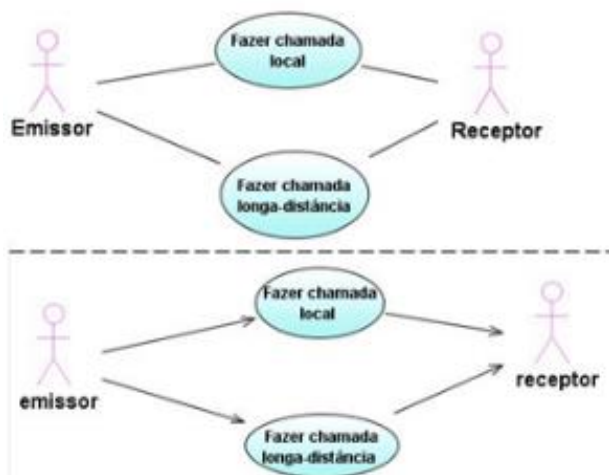


Figura 17: Associações bidirecionais e unidirecionais.

#### Inclusão

A relação de inclusão é utilizada entre dois casos, quando um deles inclui o outro na sua execução (subcaso). Um subcaso representa parte de um fluxo de eventos básico, isto é, um subfluxo que foi separado e representa um conjunto de ações atômico.

Ressalta-se que a existência de um subfluxo na descrição textual de um caso não implica sua representação no diagrama de casos de uso. A relação de inclusão (<<include>>) deve ser utilizada somente quando dois ou mais casos de uso apresentam partes idênticas nos seus fluxos de evento, caso contrário (se somente um caso de uso utilizar o subfluxo) não deve ser representado. Isto requer que os fluxos de evento sejam escritos antes de se colocar relações de inclusão no diagrama.

A figura 18 mostra um caso efetuar matrícula que possui subfluxos escolher disciplinas, alocar alunos às turmas e emitir boleto. Este último é compartilhado com o caso Efetuar inscrição curso opcional e, portanto, deve ser representado no diagrama. Um erro comum que adiciona complexidade ao diagrama é incluir os subfluxos escolher disciplinas e alocar alunos às turmas no diagrama.

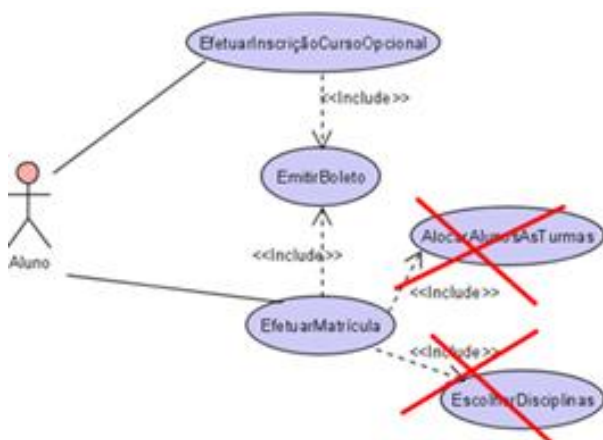


Figura 18: exemplo de inclusão de casos.

É importante ressaltar que:

- ◇ um caso de uso nunca deve ser incluído apenas por um caso, ou seja, não utilizar <<include>> para decompor o diagrama em partes;
- ◇ um caso de uso que é incluído por vários outros não tem conhecimento sobre quem o inclui, portanto, podem ser incluídos por qualquer caso sem sofrer modificações;
- ◇ não utilizar a relação de inclusão para representar opções de menu, pois o caso que faz a inclusão seria um simples despachante, todo o comportamento estaria fragmentado nos casos incluídos.

Enfim, inclusão deve ser utilizada para administrar comportamentos comuns e não para estruturar funcionalmente o diagrama.

#### Extensão

Um caso pode estender outro quando se deseja inserir um comportamento opcional ou excepcional disparado por alguma condição (ex. um alarme ou condição especial de algum objeto). Situações que podem levar ao uso da relação de extensão (Bittner e Spencer, 2003):

- ◇ Descrições opcionais ao comportamento normal do sistema: por exemplo, módulos que podem ser comprados do desenvolvedor ou de terceiros.
- ◇ Descrições de tratamento de erros e exceções complexos: estes tratamentos podem ser extremamente longos e ofuscar o fluxo básico.
- ◇ Customização: fluxos alternativos que especificam como diferentes clientes tratam certas situações no dentro do mesmo caso de uso base.
- ◇ Administração de escopo e de release: comportamentos que serão incluídos futuramente. É importante ressaltar que:
  - ◇ Um caso de uso de extensão não requer modificações no caso base (aquele que é estendido). O comportamento básico do caso base permanece intacto.
  - ◇ Um caso de uso que estende um caso base conhece este último (não é muito comum um caso de uso estender mais de um caso base).
  - ◇ Uma extensão nasce como um fluxo alternativo, mas nem todo fluxo alternativo vira uma extensão.



◇ Casos de uso que estendem assumem o controle no ponto de extensão e quando terminam devolvem o controle no mesmo ponto.

Aqui cabe uma distinção entre fluxos alternativos e casos de uso que estendem outros. Fluxos alternativos são parte do caso de uso base e têm acesso ao seu estado, pré-condições, outros fluxos existentes e pontos de extensão além daquele onde se inserem. Casos de uso que estendem conhecem apenas o ponto de extensão onde se inserem no caso estendido. Para saber se um fluxo alternativo é candidato a ser uma extensão, deve responder positivamente à questão: o sistema pode ser entregue sem a extensão?

Na figura 19, a emissão de histórico escolar é estendida pelo caso imprimir comprovante de término quando o aluno solicitante for formado. Observa-se que é um comportamento opcional que pode não ser oferecido sem prejuízo ao comportamento básico emitir histórico escolar.

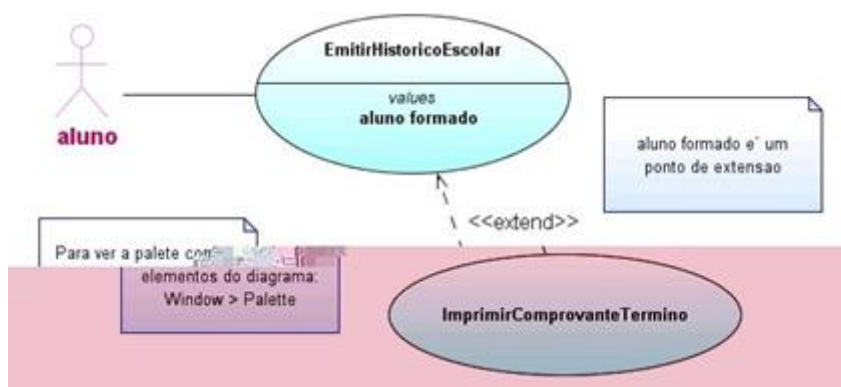


Figura 19: exemplo de relação de extensão entre casos de uso.

Nota-se na o ponto de extensão público denominado {aluno formado} onde o comportamento opcional imprimir comprovante término é inserido. É provável que existam outros pontos de extensão privados definidos nos fluxos de emitir histórico escolar, porém, no diagrama só os usados pelas extensões são listados. A figura 20 ilustra o diagrama de casos de uso para o exemplo buscar produto e fazer pedido.

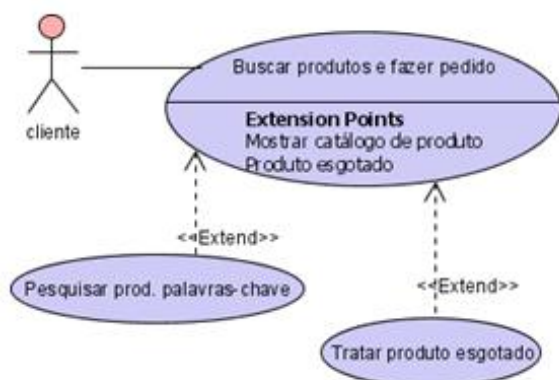


Figura 20: pontos de extensão para o caso buscar produtos e fazer pedido.

### Generalização/Especialização

A relação de generalização/especialização pode ocorrer entre casos de uso ou entre atores.

#### Caso x Caso

Generalização permite especificar comportamentos genéricos que podem ser especializados para atenderem necessidades específicas. Normalmente é utilizado quando se quer descrever famílias de sistemas.

Por exemplo, uma empresa que desenvolve software para terminais bancários de auto-atendimento quer expandir seus negócios para outras áreas, tais como pagamento direto em bombas de gasolina.

**NOME:** Realizar transação (caso abstrato)

**DESCRIÇÃO:** Permite ao usuário comprar mercadorias de um terminal automático sendo que o valor das mercadorias descontado de uma conta bancária.

**PRÉ-CONDIÇÕES:** (e) o cliente possui um cartão bancário; a conexão com o banco está ativa; o terminal deve ter mercadoria.

**PÓS-CONDIÇÕES:** (ou) O terminal retornou o cartão bancário ao cliente, entregou a mercadoria ao cliente e debitou o valor de sua conta. O terminal retornou o cartão bancário ao cliente, não entregou nenhuma mercadoria e nenhum valor foi debitado da sua conta.

**FLUXO BÁSICO DE EVENTOS**

1. O ator cliente insere o cartão bancário no terminal.
2. O sistema lê as informações da conta do cliente no cartão bancário
3. O sistema solicita ao cliente a senha
4. O cliente fornece a senha
5. O sistema verifica se a senha fornecida pelo cliente é idêntica à lida do cartão bancário
6. O sistema contata com o Sistema Bancário para verificar se as informações da conta do cliente são válidas
7. O sistema solicita o valor da transação
8. O sistema contata o Sistema Bancário para verificar se o cliente tem saldo para cobrir a solicitação.

**{Cliente realiza a transação}**

9. O sistema registra o valor da transação.
10. O sistema comunica ao Sistema Bancário que a transação foi efetuada.
11. O sistema grava no log os dados da transação: data, hora, valor e conta.
12. Término do caso de uso.

**NOME:** Sacar (caso concreto)

**DESCRIÇÃO:** Especializa o caso de uso realizar transação para permitir ao cliente retirar dinheiro de um terminal de auto-atendimento bancário.

**PRÉ-CONDIÇÕES:** (e) o cliente possui um cartão bancário; a conexão com o banco está ativa; o terminal deve ter dinheiro.

**PÓS-CONDIÇÕES:** (ou) O terminal retornou o cartão bancário ao cliente, entregou o dinheiro ao cliente e debitou o valor de sua conta. O terminal retornou o cartão bancário ao cliente, não entregou dinheiro e nenhum valor foi debitado da sua conta.

**FLUXO BÁSICO DE EVENTOS**

**Em {Cliente realiza transação}**

1. O sistema verifica se tem dinheiro suficiente em relação ao montante solicitado pelo cliente.
2. O sistema entrega o montante solicitado.
3. O sistema solicita que retire o dinheiro do terminal.
4. O cliente pega o dinheiro.

## 5. Retoma-se o caso de uso abstrato em {Cliente realiza transação}

**NOME:** Abastecer veículo (caso concreto)

**DESCRIÇÃO:** Especializa o caso de uso *realizar transação* para permitir ao cliente obter combustível de uma bomba debitando o valor de sua conta.

**PRÉ-CONDIÇÕES:** (e) o cliente possui um cartão bancário; a conexão com o banco está ativa; a bomba tem combustível.

**PÓS-CONDIÇÕES:** (ou) O terminal retornou o cartão bancário ao cliente, liberou o combustível ao cliente e debitou o valor de sua conta. O terminal retornou o cartão bancário ao cliente, não liberou combustível e nenhum valor foi debitado da sua conta.

**FLUXO BÁSICO DE EVENTOS**

**Em {Cliente realiza transação}**

1. O sistema solicita ao cliente para tirar o bico de abastecimento e libera o fornecimento de combustível.
2. O cliente enche o tanque até atingir o valor informado ou até que o tanque esteja cheio.
3. O cliente recoloca o bico na bomba.
4. Retoma-se o caso de uso abstrato em {Cliente realiza transação}

Diagrama de casos de uso correspondente aos casos acima é ilustrado na figura 21.



Figura 21: generalização/especialização de casos de uso.

Ressalta-se que nesta situação, são executados os casos de usos especializados. Eles apenas reusam partes do caso geral. A tabela 3 mostra as diferenças entre especialização e extensão de casos de uso.

Tabela 3: comparativo entre especialização e extensão de casos de uso.

Especialização	Extensão
O caso de uso especializado é executado	O caso de uso base é executado
caso de uso base não precisa ser completo e com sentido. Há várias lacunas preenchidas somente nas especializações.	O caso de uso base deve ser completo e com sentido.
comportamento de uma execução depende unicamente do caso específico.	comportamento de uma execução depende do caso de uso base e de todas as
extensões que são executadas.	

**Ator x Ator**

Especialização de atores representa que um conjunto deles possui responsabilidades ou características em comum. Algumas dicas para evitar modelagens desnecessárias:

- ◇ Não utilizar atores para representar permissões de acesso.
- ◇ Não utilizar atores para representar organogramas (hierarquias) de cargos de uma empresa.
- ◇ Utilizar atores somente para definir papéis em relação ao sistema.

Por exemplo, se num sistema de matrículas de uma universidade há casos de uso especiais para alunos de ciências exatas e para alunos de humanas, então é sinal que estes alunos são especializações de um ator genérico aluno. A figura 22 ilustra a notação UML para este caso. Observar que alunos de ciências exatas e de humanas herdam todas as associações do ator aluno.

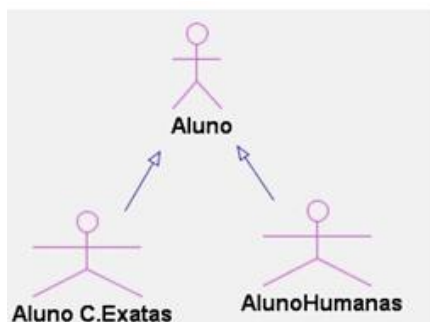


Figura 22: Especialização de atores.

## Modelagem

### Dicas

#### Casos de uso auxiliares

- ◇ Casos de uso auxiliares são frequentemente esquecidos, pois não são essenciais à funcionalidade do sistema. Porém, esquecê-los completamente pode conduzir a um sistema difícil de ser utilizado.
- ◇ Lembrar de colocar casos de uso para executar, manter e configurar o sistema, tais como: lançar e parar o sistema, incluir novos usuários, fazer backup das informações, incluir novos relatórios e realizar configurações.

#### Decomposição funcional

- ◇ Não é necessário detalhar em excesso os casos de uso. Muitos detalhes levam a decomposição dos casos em funções. O objetivo é compreender o sistema através de cenários de utilização.
- ◇ Casos de uso não são feitos para analisar (no sentido de decompor) os requisitos em requisitos menores. É um processo de síntese ou elaboração (e não de análise) no qual o problema não é totalmente conhecido. Quebrá-lo em partes menores (análise) dificulta a obtenção de uma visão geral.
- ◇ Em equipes com forte competência em análise estruturada, há tendência em encontrar funções ao invés de casos de uso. Por exemplo, fazer pedido, revisar pedido, cancelar pedido e atender pedido podem parecer bons casos de uso. No fundo, todas estas funções estão relacionadas ao caso de uso realizar pedido.
- ◇ Decomposição funcional pode levar a um número intratável de casos, mesmo para pequenos sistemas, e à perda de foco no que realmente é importante no sistema (o que traz valor aos atores).
- ◇ Casos de uso não chamam outros casos de uso ou se comunicam com outros casos.

#### Estrutura e detalhamento

- ◇ Não estruturar demais o diagrama de casos de uso, isto é, não inclua relações entre casos de uso a não ser que sejam extremamente necessárias. O uso em excesso destas relações pode fragmentar diagrama, diminuindo a compreensão global.
- ◇ O modelo deve ser o mais simples e direto possível.
- ◇ Não descrever o que ocorre fora do sistema. Por exemplo, interação entre atores pode ser importante para o negócio, mas se o sistema não facilita esta interação, deixe-a fora. Se for necessário esclarecer estes pontos faça um modelo do negócio e não um modelo de casos de uso.
- ◇ Não fazer casos tais como incluir, consultar, alterar e excluir (ICAE). Casos de uso que descreve comportamentos ICAE não adicionam muito valor à descrição da funcionalidade do sistema. Para estes tipos de comportamentos, os requisitos são bastante claros e não se deve perder tempo especificando-os. A maior parte destes comportamentos tem um padrão mostrar campos, usuário entra com os dados, sistema valida, usuário corrige erros,... A validação, a parte mais importante dos comportamentos ICAE, pode ser capturada no domínio do modelo (por meio de relações, cardinalidade e restrições) ou textualmente no glossário de dados.

#### Modelo de casos de uso é mais que um diagrama

- ◇ O diagrama de casos de uso é uma visão panorâmica dos casos de uso e, portanto, apenas um dos componentes do modelo de casos de uso. A descrição textual dos mesmos é a parte mais importante do modelo. São elementos do modelo de casos de uso: glossário, modelo do domínio e diagramas de atividades.
- ◇ Um glossário e um modelo do domínio podem evitar o excesso de detalhes nas descrições dos casos de uso. Por exemplo, ao invés de descrever a validação da entrada de dados, os campos podem ser definidos no glossário com os respectivos valores possíveis.



## **Conhecimentos em HTML 5, CSS 3, Metodologia CSS (OOCss, BEM, Smacss)**

### **HTML 5**

Ao ser anunciada a versão 5 do padrão HTML, inicialmente parte do mercado não recebeu com grande entusiasmo, acostumado a receber poucos recursos novos de uma versão para outra. A verdade é que, desde que a versão 4.0 foi lançada em 1997, poucos avanços aconteceram nos dez anos seguintes. O padrão foi atualizado para 4.01 (praticamente uma errata) e o padrão XHTML foi criado e posteriormente atualizado para 1.1 – padrão este que se resume na HTML 4.01 com algumas variações em XML. A World Wide Web permaneceu praticamente estática neste período por várias razões.

Uma destas razões é que o W3C (órgão que regulamenta os padrões Web, entre eles a HTML) tem um ciclo de versão demorado, que exige versões de rascunho e períodos de contribuição da comunidade, sugerindo novos conceitos e posteriormente validando, concordando e discordando dos mesmos – processos estes que levam tempo.

Em segundo lugar, os fabricantes de navegadores web demoravam muito a adotar o novo padrão, por muitas vezes adotando-o parcialmente – isso quando não fugiam dele, criando elementos e conceitos particulares, tornando-o um website aderente apenas a este ou aquele navegador web. Alguns fabricantes atualizavam seus navegadores web e padrões suportados apenas na troca do sistema operacional, o que poderia levar até três anos para acontecer. Isso sem falar nos navegadores do mundo mobile.

E, finalmente, como se os dois primeiros obstáculos não fossem o bastante, os internautas não atualizavam suas versões de navegador web – seja por inexperiência ou descaso – permanecendo com versões antigas que não davam suporte ao novo padrão. O absurdo chega ao ponto que o Microsoft Internet Explorer em sua versão 6.0 era o navegador web mais utilizado na Internet Brasileira, mais de 10 anos após seu lançamento.

Como resposta à lentidão do W3C, uma comunidade paralela formada por profissionais da Mozilla Foundation (fabricante do Firefox), Opera Software (navegador Opera) e Apple (Safari) foi criada em 2004 com o nome de WHATWG (Web Hypertext Application Technology Working Group), com o intuito de discutir novos padrões e recursos para a Web.

Recebendo posteriormente contribuições de outras empresas como Google e Microsoft rapidamente o novo padrão foi ganhando forma de tal maneira que, três anos depois, foi submetido ao W3C que por sua vez decidiu adotá-lo batizando-o de HTML5. O novo padrão ajudou a enterrar o XHTML 2.0, padrão que o W3C estava trabalhando em 2007, considerado por muitos um equívoco – o padrão não era sequer compatível com a versão 1.1.

À primeira vista, a HTML5 parece se tratar meramente de um conjunto de novas tags para renderização de texto e formulários, impressão essa reforçada ao se folhear a maioria dos livros disponíveis hoje no mercado. Não se engane. Estas tags mencionadas são apenas a ponta do iceberg.

Embora seja chamada de HTML5 e a sigla signifique HyperText Markup Language, ou seja, linguagem de marcação de hipertexto, as novidades vão além disso. A HTML5 é um grande guarda-chuva tecnológico, pois pendurados a ele estão o novo padrão CSS na versão 3 e uma imensa gama de novas APIs na linguagem JavaScript estendendo as funcionalidades e possibilidades da WWW para patamar absolutamente impressionantes.

### **Diferenças da HTML4**

Este artigo não estaria completo se não discorrêssemos acerca das diferenças entre a nova HTML5 e a sua antecessora, a HTML4, esta que ocupou lugar de destaque no mundo web por muitos anos. Ainda teremos um bom tempo de espera até que a HTML5 se torne de fato um padrão na web e que todos os browsers e tecnologias afins assumam a mesma como centro de implementação. Como programador web, é muito interessante que saiba quais as principais diferenças entre ambas as versões, justamente para que possa salvar tempo e aumentar a produtividade em situações como essa.

Uma das características mais marcantes dessa nova versão da linguagem é o fato de que a mesma não é uma versão final, isto é, estará sempre e continuamente mudando ao longo do tempo. Isso inclui dizer que os desenvolvedores da linguagem estarão sempre adicionando e removendo atributos, tags e o que considerarem interessantes à mesma. Ao mesmo tempo, constitui um risco se você estiver usando a mesma como algo definitivo no seu projeto. Isso significa que se optar pela HTML5 terá de seguir suas atualizações e estar constantemente evoluindo seu código também.

A HTML5 foi feita para ser simples, isso implica em uma sintaxe extremamente mais simples e limpa. A simples declaração do doctype foi apenas mais uma das facilidades incluídas na nova versão. Agora, você precisa inserir apenas um `<!doctype html>` no início do seu documento e tudo estará pronto. Além disso, a sintaxe da HTML5 é compatível também com a HTML4 e XHTML1.

A linguagem apresenta também um elemento novo, que veremos aqui no artigo, o `<canvas>`, responsável por substituir muitas das implementações antes feitas em Flash, o que faz muitos desenvolvedores considerar que este já se encontra obsoleto e futuramente morto.

A extensão de tags a um tool de novos e interessantes recursos fez uma grande diferença na linguagem. Tags como: `<header>` e `<footer>`, que estendem a funcionalidade de tabelas agora para a página como um todo, `<section>` e `<article>`, que permitem marcar áreas específicas dos layouts, `<video>` e `<audio>` para uma inclusão melhorada de conteúdos multimídia nas páginas, e `<menu>` e `<figure>` para bem arranjar textos, imagens e menus, trazem todo um conjunto de implementações e funcionalidades bem pertinentes para a web de hoje.

Além disso tudo, a remoção de alguns outros recursos como as tags `<center>`, `<big>`, `<font>`, etc fazem com a responsabilidade do CSS aliado à nova linguagem só aumente, otimizando o desenvolvimento front-end.

#### Convertendo de versões antigas

Para converter códigos políglotas e obsoletos para a nova versão da HTML5, basicamente temos de seguir três passos:

1. Remova os identificadores PUBLIC FPI e SYSTEM da declaração do DOCTYPE;
2. Substitua quaisquer tags depreciadas da HTML ou construa sua implementação baseada num código HTML que seja de acordo com os padrões da HTML5. Por exemplo, tenha certeza de assegurar que qualquer conjunto de tags `<col>` para colunas de tabelas HTML sempre tenham um elemento "colgroup" como seu parente de configuração;
3. Comece a tirar vantagem dos novos recursos da HTML5, tais como convertendo suas div's para as tags de seção da HTML5.

Mas e se necessitarmos realizar conversões de versões mais antigas ainda da HTML? Esse tipo de conversão requer um pouco mais de trabalho e atenção, uma vez que ocorreram notórias mudanças nas versões da HTML entre os anos 1997 e 2000 com o objetivo de suportar conversores baseados em XML, dispositivos móveis com regras de conversão bem restritas, templates client-side cacheáveis além de agregação com outros tipos de conteúdo como os velhos feeds RSS. Vejamos mais alguns passos para tal:

1. Confira sempre se o documento inicia com uma declaração XML e uma declaração de DOCTYPE;
2. Verifique se o elemento de top `<html>` inclui o atributo `"xmlns="http://www.w3.org/1999/xhtml"`. (A URI para XHTML e versões mais recentes da HTML incluem o ano "1999" porque foi definido aquele ano enquanto a W3C HTML Recommendation lançada em 2000 estava ainda em fase de desenvolvimento.);
3. Verifique se todas as tags estão marcadas com a tag de fim, ou se estão todas "self-closed" com o sinal de `</>`. Não se esqueça de averiguar se os nomes dos elementos iniciam e terminam com a marcação em minúscula.
4. Verifique se todos os valores de atributos estão cercados pelas aspas. Verifique também se os atributos booleanos estão codificados na sua forma full, usando o atributo name entre aspas como o

valor (attribute="attribute") quando o mesmo valor for true e o omitindo completamente quando o valor for false.

5. A forma full será apropriadamente entendida pelos web browsers que convertem documentos com a ainda sintaxe HTML ou a sintaxe XML do HTML5. Evite usar formas minimizadas para selected="", a qual o XPath trata como false ao invés de true.

Observe que a especificação HTML5 explicita estado como:

Os valores "true" e "false" não são permitidos em atributos booleanos

Isso acontece porque os browsers olham para o código com o valor booleano para os atributos e tratarão a string "false" como um valor false enquanto browsers que somente olham para a presença ou ausência do atributo tratarão esse código como true, resultando em comportamentos inconsistentes e confusos.

Atributos booleanos que precisam ser mudados incluem alguns dos tipos ilustrados na Tabela 1.

Tabela 1. Lista de atributos e respectivas mudanças de tipo booleano

Atributo	Mudança
Async	Mudar para async="async"
Checked	Mudar para checked="checked"
Compact	Mudar para compact="compact"
Declare	Mudar para declare="declare"
Defer	Mudar para defer="defer"
Disabled	Mudar para disabled="disabled"
Ismap	Mudar para ismap="ismap"
Multiple	Mudar para multiple="multiple"
Noresize	Mudar para noresize="noresize"
Noshade	Mudar para noshade="noshade"
Nowrap	Mudar para nowrap="nowrap"
Open	Mudar para open="open"
Readonly	Mudar para readonly="readonly"
Required	Mudar para required="required"
Reversed	Mudar para reversed="reversed"
Scoped	Mudar para scoped="scoped"
Selected	Mudar para selected="selected"

Observe que "true" e "false" são valores válidos para alguns atributos "não booleanos", em particular atributos enumerados que o atributo draggable.

Outra forma de se trabalhar migrando esse tipo de documento é verificando qual o tipo de versão que foi usada para construir o documento HTML. Uma boa forma de se fazer isso é submeter a URL do website ao site do W3C: Markup Validation Service. Ao executar, os possíveis resultados são:



- **HTML5:** Indica que o site já foi convertido para o padrão HTML5. Por exemplo, o próprio site do Google é um bom exemplo de teste que usa o padrão.
- **XHTML 1.0 Strict:** Indica que o site está usando a versão padrão 2000 W3C da HTML. Por exemplo, o próprio site do W3C adere à essa versão.
- **XHTML 1.0 Transitional or HTML 4.01 Transitional:** Indica que o site está usando o formato transicional entre a versão padrão da HTML4 de 1997 e a versão padrão da HTML de 2000. Por exemplo, o site AltaVista.com usa o formato da HTML 4.01 Transitional, já o site da Microsoft usa XHTML 1.0 Transitional.
- **HTML 4.01 Strict:** Indica que o site está usando a versão antiga da HTML4 de 1997. Por exemplo, o site do Yahoo usa o antigo padrão.

### **As Oito Áreas da HTML5**

O novo padrão é vasto. Por esta razão, O W3C dividiu nestas novidades em oito áreas tecnológicas. A divisão ajuda na homologação do padrão e suporte dos fabricantes, que estão sendo feitos em porções. Estas áreas são:

- **Semantics (Semântica):** A ponta do iceberg. Nesta área, estão as novas tags que auxiliam na análise semântica dos textos presentes nas páginas. Muito úteis especialmente para melhorar a eficiência dos mecanismos de busca, como o Google. Adicionalmente, novas tags para renderização de formulários, em destaque caixas de texto com validação nativa.
- **Offline & Storage (Fora do Ar e Armazenamento):** Aborda todas as funcionalidades referentes ao tratamento do website quanto o visitante estiver em modo offline, além das novidades em armazenamento de informações no navegador como o LocalStorage e o Banco de Dados IndexedDB, indo mais além dos famigerados cookies.
- **Device Access (Acesso por Dispositivos):** Compreende todas as APIs que estendem a experiência de visitantes de dispositivos móveis como tablets e smartphones, como a Geolocalização (usando GPS ou outros recursos para determinar onde está o usuário), o uso do acelerômetro (utilizado para determinar a orientação do dispositivo no espaço), o uso do microfone e câmera destes dispositivos, além de eventos específicos para telas sensíveis a toque.
- **Connectivity (Conectividade):** Avanços incríveis na conectividade de aplicações web, como a comunicação usando WebSockets e SSE (Server Side Events), essenciais especialmente em jogos eletrônicos.
- **Multimedia:** Novas tags para a publicação de áudio e vídeo na Internet, além de novos formatos multimedia suportados.
- **3D, Graphics & Effects (3D, Gráficos e Efeitos):** Engloba aqui o Canvas que permite desenhar elementos gráficos em uma página, como uma tela de desenho. Adicionalmente, o suporte a WebGL para renderizações em 3D, o suporte ao formato SVG e efeitos 3D obtidos através de CSS3.
- **Performance & Integration (Performance & Integração):** Melhorias significativas na manipulação e submissão de formulários pela Internet. Possibilidades de processamento paralelo utilizando os novíssimos Web Workers, além de melhorias integração com o visitante (interface) com o excelente recurso de Drag 'n Drop (arrastar-e-soltar).
- **CSS3:** Novíssimos estilos e efeitos sem sacrificar performance ou indexação dos mecanismos de busca.

### **CSS3**

CSS é chamado de linguagem Cascading Style Sheet e é usado para estilizar elementos escritos em uma linguagem de marcação como HTML. O CSS separa o conteúdo da representação visual do site. Pense na decoração da sua página. Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode criar tabelas, usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante.

CSS foi desenvolvido pelo W3C (World Wide Web Consortium) em 1996, por uma razão bem simples. O HTML não foi projetado para ter tags que ajudariam a formatar a página. Você deveria apenas escrever a marcação para o site.

Tags como <font> foram introduzidas na versão 3.2 do HTML e causaram muitos problemas para os desenvolvedores. Como os sites tinham diferentes fontes, cores e estilos, era um processo longo, doloroso e caro para reescrever o código. Assim, o CSS foi criado pelo W3C para resolver este problema.

A relação entre HTML e CSS é bem forte. Como o HTML é uma linguagem de marcação (o alicerce de um site) e o CSS é focado no estilo (toda a estética de um site), eles andam juntos.

CSS não é tecnicamente uma necessidade, mas provavelmente você não gostaria de olhar para um site que usa apenas HTML, pois isso pareceria completamente abandonado.

### Vantagens do CSS

A diferença entre um site que implementa CSS e outro que não o usa é gigantesca e notável.

Você já deve ter visto um site que não carrega completamente ou tem um plano de fundo branco com texto azul e preto. Isso significa que a parte CSS do site não foi carregada corretamente ou não existe.

E é assim que um site somente com HTML se parece. Acredito que você vai concordar comigo de que isso não é muito bonito, certo?

Antes de usar CSS, toda a estilização tinha que ser incluída na marcação HTML. Isso significa que você deveria descrever separadamente todo o plano de fundo, as cores das fontes, os alinhamentos, etc.

Mas o CSS permite que você estilize tudo em um arquivo diferente, criando assim o estilo separadamente. E, mais tarde, faça integração do arquivo CSS na parte superior da marcação HTML. Isso mantém a marcação HTML limpa e fácil de manter.

Resumindo, com o CSS você não precisa mais escrever repetidamente como os elementos individuais se parecem. Isso economiza tempo, encurta o código e diminui a chance de erros.

O CSS permite que você tenha vários estilos em uma página HTML, tornando as possibilidades de personalização quase infinitas. Hoje em dia, isso está se tornando mais uma necessidade do que um simples recurso.

### Como CSS Funciona

O CSS é uma ferramenta muito potente que possibilita criar diversas funcionalidades ao invés de usar JavaScript ou outra linguagem mais pesada. Se usado com moderação, CSS pode viabilizar uma ótima experiência ao desenvolvedor e usuários das páginas web.

Com o Cascading Style Sheets é possível criar animações complexas, criar efeitos com uso de parallax, que faz parecer que a imagem de fundo tem uma profundidade diferente um dos outros, criar sites interativos e também jogos com HTML5 e CSS3.

O CSS usa uma sintaxe simples baseada em inglês com um conjunto de regras que o governam. Como mencionamos anteriormente, o HTML nunca teve a intenção de usar elementos de estilo, apenas a marcação da página. Foi criado para descrever apenas o conteúdo. Por exemplo: <p>Este é um parágrafo.</p>.

Mas como você estiliza o parágrafo? A estrutura da sintaxe CSS é bem simples. Tem um seletor e um bloco de declaração. Você seleciona um elemento e depois declara o que deseja fazer com ele. Bastante simples, certo?

Mas tem algumas regras que você precisa saber. Isso também é simples, não se preocupe.

O seletor aponta para o elemento HTML que você deseja estilizar. O bloco de declaração contém uma ou mais declarações separadas por ponto e vírgula.

Cada declaração inclui um nome de propriedade CSS e um valor, separados por dois pontos. Uma declaração CSS sempre termina com um ponto-e-vírgula e os blocos de declaração são cercados por chaves.

Vamos ver um exemplo:

Todos os elementos <p> serão estilizados e serão coloridos de azul e negrito.

```
1. <style>
2.
3. p {
4.   color: blue;
5.   text-weight: bold;
6. }
7.
8. </style>
```

Em outro exemplo, todos os elementos <p> serão centralizados, com tamanho 16x e de cor pink.

```
1. <style>
2. p {
3.
4.   text-align: center;
5.   font-size: 16px;
6.   color: pink;
7.
8. }
9. </style>
```

#### Anatomia de um comando CSS

O CSS estipula regras para o arquivo em html. Com cada regra é possível estilizar o conteúdo todo ou somente determinados elementos. Por isso entenda, um comando básico é composto por seletor e declarações, que contém propriedade e valor.

Seletor {Propriedade: Valor}

A sintaxe do CSS é muito simples de aprender. O seletor seleciona quais elementos em html receberão a propriedade. Pode ser p (parágrafo) ou o body (corpo da sua página). Já a propriedade pode ser a cor ou algo mais específico como cor do fundo (background). E por último o valor, que determina o valor da propriedade.

Vamos simular um exemplo. Digamos que o objetivo é mudar a fonte de uma tag h1. Para isso podemos usar h1 {font-size: 20px;}

- h1 – é o seletor. Neste caso selecionamos o h1.
- font-size – é a declaração que contém a propriedade (font-size) e o valor é (20px).

Metodologia CSS

### **SMACSS**

O sistema estabelece e é bastante baseado em cinco categorias de regras de CSS: base, layout, module, state e a pouco importante theme. As regras de base são as do tipo que não utilizam seletores com classes ou ids, as encontramos em um CSS Reset ou normalize.css.

O sistema alerta sobre a agressividade dos CSS Resets mas não alerta sobre as regras deste tipo definidas no próprio projeto, ainda mais quando aplicadas a divs, spans ou headings. Regras cujos seletores não utilizam classes são globais e qualquer decisão tomada neste nível irá perpetuar por todo o projeto, cuidado.

As categorias de layout e module são bastante semelhantes. Pense no layout como elementos agregadores e geralmente únicos como header, footer e sidebar. O sistema propõe que regras de layout tenham ids ou classes com o prefixo l- como seletores.

As regras da categoria module englobam os demais componentes da página. O sistema não encoraja o uso de elementos nos seletores, preferindo .box .title ao invés de .box h2. Ainda, os seletores como .box-title são defendidos para facilitar a leitura do HTML.

Assim como o OOCSS, o sistema repudia regras do tipo #sidebar .media onde a localização do elemento passa a ser relevante para sua apresentação. O SMACSS reforça que seja adicionada uma classe para abrigar as variações. O elemento da sidebar passa a ter a classe do módulo e também a do sub-módulo: <div class="media media-sidebar">.

A categoria de state engloba regras responsáveis por gerenciar estado de componentes enquanto o usuário estiver navegando. Regras desta categoria são as únicas que podem e talvez precisem utilizar !important. O padrão indica que as classes possuam o prefixo is-. Com certeza, algum dos seus projetos já precisou de uma classe como .is-active, .is-collapsed ou is-current.

O SMACSS é mais uma série de tutoriais de como escrever um bom código que propriamente um sistema de CSS. Contra os padrões, não concordo com qualquer aparição de #id em folhas de estilos por ir contra os preceitos de reuso.

O atributo id na realidade serve mais como um destino de navegação, por isto a necessidade de ser único. Considero um pouco estranho também o prefixo l- para as classes de layout, o que me atreve a desconsiderar totalmente a categoria e a gerir suas regras como se pertencessem a categoria module. Não existe necessidade desta distinção nos seletores, apenas separar as regras em um arquivo layout.css já é suficiente.

A categoria de state é a mais interessante, o padrão fica muito conveniente para ser utilizado em código JavaScript. Quando o estado de um componente demanda regras muito específicas, o SMACSS sugere o seletor .is-tab-active. Desta maneira, o JavaScript perde um tanto da sua modularidade, seletores aninhados como .tab.is-active podem ser uma melhor jogada.

Por fim, mesmo sendo bastante válido, o SMACSS não soluciona alguns desafios típicos do design de componentes de médio porte pois em nenhum momento endereça como nomear adequadamente elementos descendentes.

### **Bem**

O BEM – sigla para block, element, modifier – é uma metodologia com várias versões cujo o preceito de esclarecer o desenvolvedor mais sobre o markup através de suas classes. Este sistema permite escrever sites de maneira rápida, auto-explicativa e com manutenção descomplicada. Esqueça seus preconceitos com os caracteres duplos de hífen, eu já deixei o meu de lado, e reflita a seguir o quanto mais de informação a classe .report-graph\_\_bar\_size\_bigoferrece em relação as tradicionais .bar, .report-graph-bar ou .graph-bar.

O block é uma entidade independente da aplicação, podendo ser o mais alto nível de abstração (header, footer) ou componente (graph, tabs). O element é um descendente dependente de um block que possui uma certa função. Para permitir nomes compostos e evitar ambiguidades, o padrão estabelece o controverso `__` como separador. Desmembrando a classe `report-graph__bar`, identificamos `bar` como element e sabemos da existência do elemento pai `report-graph`, que é o block.

O estilo define o modifier como uma propriedade de um block ou element que altera sua aparência. Desta maneira, o block `.menu` poderia ser acrescido da classe `.menu_size_big`, note o uso de `_`.

A fim de deixar o padrão um pouco mais claro, veja um exemplo:

```
<div class="report-graph">
  <div class="report-graph__bar">...</div>
  <div class="report-graph__bar report-graph__bar_size_big">...</div>
</div>
```

As classes `.report-graph`, `report-graph__bar` e `report-graph__bar_size_big` são, respectivamente, block, element e modifier. Neste ponto você já deve ter refletido e concluído que sim, há muito mais informação numa classes nomeada neste padrão.

Uma das falhas do estilo é não possuir categorias como as do SMACSS. Segundo o padrão, o estado de um componente deve ser endereçado como um modifier, não há uma categoria de state. Conforme citei anteriormente, classes modifier como `.menu__item_state_current` tiram a modularidade do JavaScript fazendo com que o código dependa do componente. Alguns desenvolvedores também podem sentir falta do estilo não versar nada sobre regras aplicadas diretamente a elementos (categoria base do SMACSS) e boas práticas de CSS.

O grande trunfo do padrão é mesmo sugerir uma nomenclatura adequada para elementos descendentes. Outro aspecto interessante é o padrão de organização de arquivos.

## DRY CSS

O princípio consiste em não repetir propriedades com mesmos valores em seu código. De maneira simples, a todo momento que isto for necessário, estas propriedades devem ser agrupadas e endereçadas por vários seletores. O sistema define que o agrupamento seja nomeado através de um primeiro seletor, algo como `#MEDIUM-WHITE-BACKGROUND` para um agrupamento de background e border brancas. As custas deste primeiro seletor, muitos desenvolvedores confundem que o pattern sugere que este seletor seja usado no HTML, o que não é verdade.

A técnica, assim como outros sistemas, sugere que seu código seja pensando em termos de padrões de aparência. O problema está no fato da técnica não sugerir melhor uso das classes de CSS. Segundo o sistema, sempre que a aparência de um elemento mudar, sua classe precisará ser movida para outros agrupamentos nas folhas de estilo. Sistemas como SMACSS e BEM endereçam mudanças de aparência com a criação de um submódulo ou modificador que será adicionado ao HTML evidenciando a alteração e permitindo manter a antiga aparência.

Não entenda mal, os conceitos por trás do DRY CSS são válidos. Inclusive, já sugeri uma variação da técnica em conjunto com placeholders do SASS para nomear propriedades que definem um elemento da aparência do seu projeto. Os placeholders carregam os conceitos de DRY CSS e ainda permitem basear sua arquitetura em outros sistemas mais poderosos.

## Modelagem de Dados

A modelagem de dados é uma etapa importante e essencial em qualquer projeto de desenvolvimento ou manutenção de software. Assim, estar atualizado neste assunto é importante para qualquer profissional da área. Serve como material introdutório ao tema de modelagem de dados, apresentando um breve resumo sobre o assunto.

O objetivo deste artigo é apresentar uma visão geral sobre as habilidades de modelagem de dados fundamentais que todos os desenvolvedores devem ter, habilidades que podem ser aplicadas em projetos tradicionais que seguem uma abordagem em série e aplicadas em projetos ágeis que seguem uma abordagem evolucionária. Idealmente, todos profissionais de TI deveriam ter um entendimento básico sobre modelagem de dados. Eles não precisam ser especialistas neste assunto, mas deveriam estar preparados para se envolverem na criação de um modelo, estar apto a ler um modelo de dados existente, entender quando criar um modelo de dados e quando não criar e conhecer as técnicas de projeto de dados fundamentais.

Neste contexto, este artigo apresenta uma breve introdução sobre estas habilidades. Seu público principal são os desenvolvedores de aplicações que precisam obter um entendimento de alguma das atividades críticas realizadas por um DBA. Este entendimento deve levar ao conhecimento do que um DBA faz e por que faz, ajudando a estabelecer uma ponte de comunicação reduzindo a distância existente entre esses dois papéis fundamentais no desenvolvimento de software (desenvolvedor e DBA).

O que é modelagem de dados?

Modelagem de dados é o ato de explorar estruturas orientadas a dados. Como outros artefatos de modelagem, modelos de dados podem ser usados para uma variedade de propósitos, desde modelos conceituais de alto nível até modelos físicos de dados. Do ponto de vista de um desenvolvedor atuando no paradigma orientado a objetos, modelagem de dados é conceitualmente similar à modelagem de classes. Com a modelagem de dados identificamos tipos de entidades da mesma forma que na modelagem de classes identificamos classes. Atributos de dados são associados a tipos de entidades exatamente como associados atributos e operações às classes. Existem associações entre entidades, similar às associações entre classes – relacionamento, herança, composição e agregação são todos conceitos aplicáveis em modelagem de dados.

Modelagem de dados tradicional é diferente da modelagem de classes porque o seu foco é totalmente nos dados – modelos de classes permitem explorar os aspectos comportamentais e de dados em um domínio de aplicação, já com o modelo de dados podemos apenas explorar o aspecto dado. Por causa deste foco, projetistas de dados tendem a serem melhores em identificar os dados “corretos” em uma aplicação do que modeladores de objetos. No entanto, algumas pessoas modelam métodos de banco de dados (stored procedures, stored functions e triggers) quando estão realizando a modelagem física dos dados.

Apesar de o foco deste artigo ser modelagem de dados, existem normalmente alternativas para artefatos orientados a dados. Por exemplo, quando estamos na modelagem conceitual, os diagramas ORM (Object Role Model) não são a única opção. Além do Modelo Lógico de Dados, é comum a criação de diagramas de classes da UML.

Como modelos de dados são usados na prática?

Embora as questões de metodologias sejam abordadas depois, precisamos discutir como modelos de dados podem ser usados na prática para melhor entendê-los. Provavelmente, iremos nos deparar a três estilos básicos de modelos de dados:

Modelos de dados conceituais: esses modelos, algumas vezes chamados modelos de domínio, são tipicamente usados para explorar conceitos do domínio com os envolvidos no projeto. Em equipes ágeis, modelos conceituais de alto nível são normalmente criados como parte do esforço inicial do entendimento dos requisitos do sistema, pois eles são usados para explorar as estruturas e conceitos de negócio estáticos de alto nível. Em equipes tradicionais (não ágeis), modelos de dados conceituais são normalmente criados como precursores aos modelos lógicos de dados (MLD) ou suas alternativas.

Modelos Lógico de Dados (MLDs): MLDs são usados para explorar os conceitos do domínio e seus relacionados. Isso pode ser feito para o escopo de um simples projeto ou para uma empresa inteira.

MLDs descrevem os tipos de entidades lógicas, tipicamente referenciadas simplesmente como tipos de entidades, os atributos de dados que descrevem essas entidades e os relacionamentos entre as entidades. MLDs são raramente usados em projetos ágeis apesar de normalmente estarem presentes em projetos tradicionais (onde eles raramente adicionam muito valor na prática).

Modelos Físicos de Dados (MFDs): MFDs são usados para projetar o esquema interno de um banco de dados, descrevendo as tabelas de dados, as colunas de dados das tabelas e o relacionamento entre as tabelas. MFDs normalmente são bastante úteis em projetos ágeis e tradicionais, por isso este será o foco deste artigo: modelagem física dos dados.

Embora MLDs e MFDs parecerem similares, e eles de fato são, o nível de detalhes que eles modelam pode ser significativamente diferente. Isso porque o objetivo de cada diagrama é diferente – podemos usar um MLD para explorar conceitos do domínio com os envolvidos no projeto e MFD para definir o projeto do banco de dados.

A Figura 1 apresenta um simples MLD e a Figura 2 um simples MFD, ambos modelando o conceito de clientes e endereços, assim como o relacionamento entre eles. Ambos os diagramas seguem a notação de Barker, que será descrita a seguir.

Note como o MFD mostra mais detalhes, incluindo uma tabela associativa necessária para implementar a associação, assim como as chaves necessárias para manter os relacionamentos. Mais detalhes sobre esses conceitos serão descritos a seguir.

MFDs devem também refletir os padrões de nomenclatura de banco de dados da organização. Neste caso, uma abreviação do nome da entidade é colocado para cada nome de coluna e uma abreviação para “número” foi consistentemente introduzida. Um MFD deve também indicar os tipos de dados das colunas, tais como integer e char. Apesar de a Figura 2 não mostrá-las, tabelas de referência como para o endereço é usado, assim como para estados e países estão implícitos pelos atributos END\_USADO\_CODIGO, END\_ESTADO\_CODIGO, END\_PAIS\_CODIGO.

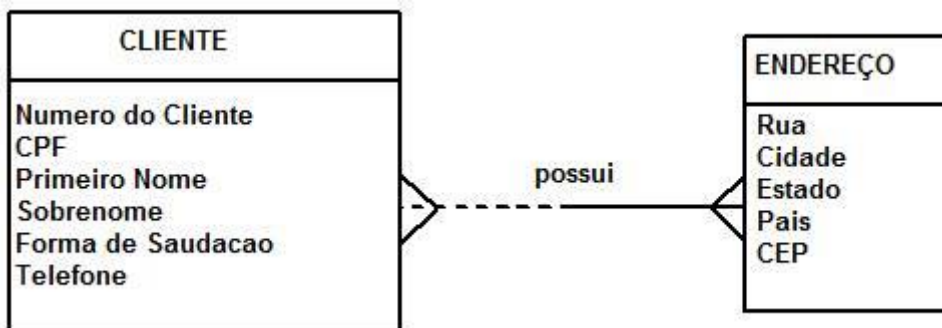


Figura 1. Um simples modelo lógico de dados



Figura 2. Um simples modelo físico de dados

Modelos de dados podem ser usados efetivamente tanto no nível da empresa como de projetos. Os arquitetos da empresa normalmente criarão um ou mais MLDs de alto nível que descreve as estruturas de dados que apoiam toda a empresa, normalmente chamados de modelos de dados da empresa ou modelos de informação da empresa.

Um modelo de dados da empresa é uma das várias visões que os arquitetos da empresa podem escolher para manter e apoiar – outras visões podem explorar a infraestrutura de rede/hardware, a estrutura da organização, infraestrutura de softwares o processo de negócios, dentre outros. Esses modelos provêm informações que uma equipe de projeto pode usar como conjunto de restrições e também como descrição da estrutura do sistema.

Equipes de projeto tipicamente criam MLDs como um dos principais artefatos de análise quando seu ambiente de implementação é predominantemente procedural por natureza, por exemplo quando estão usando COBOL estruturado como linguagem de implementação. MLDs são também boas escolhas quando um projeto é orientado a dados, como um data warehouse ou sistema de relatório.

No entanto, MLDs são normalmente escolhas ruins quando uma equipe de projeto está usando tecnologias orientadas a objeto ou baseadas em componentes porque os desenvolvedores trabalhariam melhor com diagramas UML ou quando o projeto não é orientado a dados. Como uma dica de modelagem, aplique os artefatos corretos para aquele trabalho a ser desenvolvido.

Quando um banco de dados relacional é usado para armazenar dados, equipes de projeto são aconselhadas a criar um MFD para modelar um esquema interno. MFD normalmente é apenas um dos artefatos de projeto críticos para projetos de desenvolvimento de aplicações de negócio.

O que dizer sobre modelos conceituais?

Muitos profissionais de dados preferem criar um ORM (Object-Role Model), como o apresentado no exemplo da Figura 3, em vez de um MLD para um modelo conceitual.

A vantagem é que a notação é muito simples, algo que os envolvidos no projeto podem rapidamente interpretar, apesar da desvantagem que seria o fato de os modelos se tornarem grandes rapidamente.

ORMs nos permite primeiramente explorar os exemplos de dados reais em vez de simplesmente saltar para uma abstração potencialmente incorreta – por exemplo, a Figura 3 examina o relacionamento entre clientes e um endereço em detalhe. Para mais informações visite o site da ORM.



Figura 3. Um simples ORM (Object-Role Model)

Normalmente ORMs são usados para explorar o domínio da aplicação com os envolvidos no projeto, mas depois ele é substituído por um artefato mais tradicional, como um MLD, um diagrama de classes ou até um MFD.

Notações comuns de modelagem de dados

A Figura 4 apresenta um resumo da sintaxe das quatro notações mais comuns para modelagem de dados: Engenharia da Informação (EI), Notação de Barker, IDEF1X e UML (Unified Modeling Language). Este diagrama não tem a pretensão de ser altamente compreensivo, mas sim prover uma visão geral básica sobre as notações.

Além disso, para não se estender muito no texto, não foi possível descrever a abordagem altamente detalhada para nomenclatura de relacionamento como sugerido por Barker. Apesar disso, foi provida uma breve discussão de cada notação na Tabela 1.



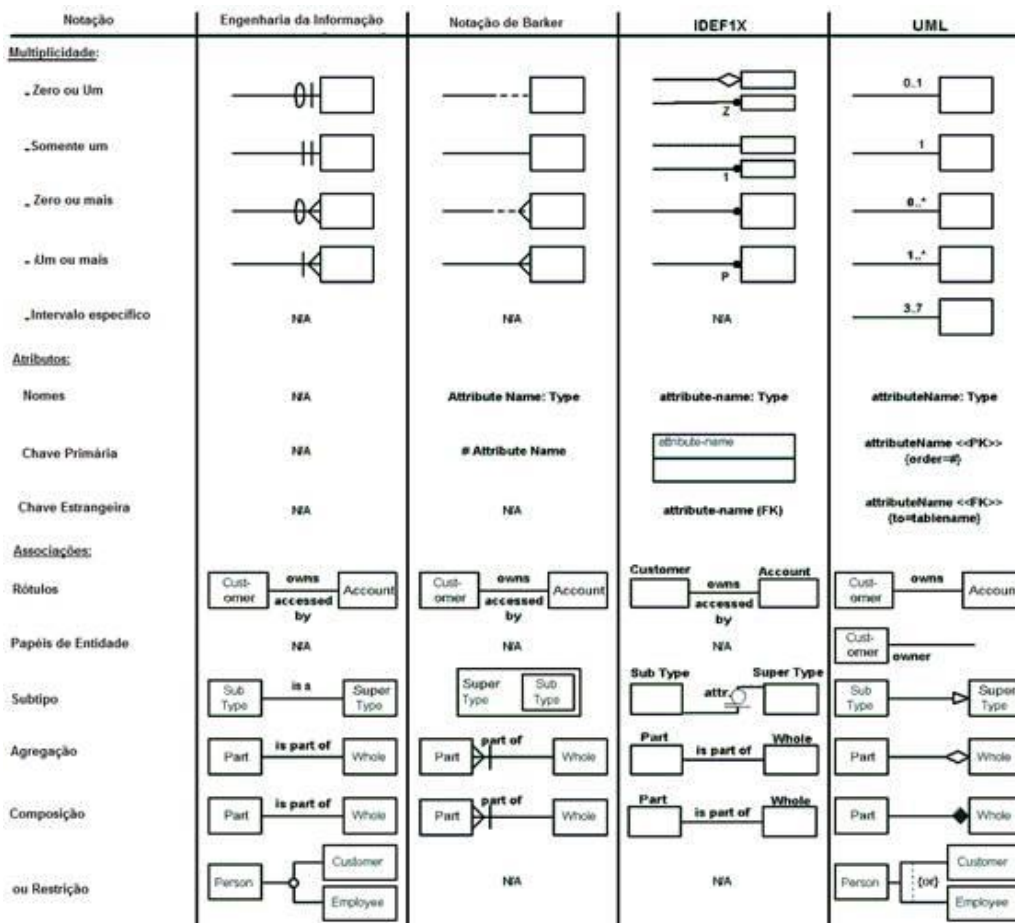


Figura 4. Comparando a sintaxe das notações comuns para modelagem de dados

Notação	Comentários
EI	A notação EI é simples e fácil de ser lida, e é bem abrangente para modelagem de dados de negócio e modelagem lógica de alto nível. O único ponto negativo desta notação é que ela não suporta a identificação de atributos de uma entidade. Assume-se que os atributos serão modelados com outro diagrama ou simplesmente descrito em uma documentação de apoio.
Barker	A notação de Barker é uma das mais populares, sendo apoiadas por várias ferramentas (ex: Oracle toolset), é bem abrangente para todos os tipos de modelos de dados. Esta abordagem pode se tornar complicada com hierarquias que possuem vários níveis de profundidade.
IDEF1X	Esta notação é a mais complexa, e foi originalmente intencionada para modelagem física, mas foi mal aplicada para modelagem lógica. Esta notação tem sido abandonada por todos, então evite-a se possível.
UML	Esta não chega a ser uma notação de modelagem de dados oficial. Apesar de várias sugestões para um perfil UML de modelagem de dados existirem, nenhum é completo e não são oficializados pela UML.

Tabela 1. Discutindo notações comuns de modelagem de dados

### Como Modelar Dados

É crucial para um desenvolvedor de aplicação ter uma noção dos fundamentos de modelagem de dados não apenas para ler os modelos de dados, mas também para trabalhar efetivamente com os DBAs responsáveis pelos aspectos relacionados aos dados do projeto. O objetivo ao ler esta seção não é aprender como se tornar um modelador de dados, mas sim obter uma apreciação a respeito do que é envolvido nesta tarefa.

As seguintes tarefas são realizadas de forma iterativa:

- Identificar os tipos de entidade;
- Identificar atributos;
- Aplicar convenção de nomes;
- Identificar relacionamentos;
- Associar chaves;
- Normalizar para reduzir a redundância dos dados;
- Diversificar para melhorar o desempenho.

### **Identificar Os Tipos de Entidade**

Um tipo de entidade, ou simplesmente entidade, é conceitualmente similar ao conceito de orientação a objeto de uma classe – um tipo de entidade representa uma coleção de objetos similares. Um tipo de entidade pode representar uma coleção de pessoas, lugares, coisas, eventos ou conceitos. Exemplos de entidades em um sistema de vendas incluiria: Cliente, Endereço, Venda, Item e Taxa. Se estivéssemos modelando classes, esperaríamos descobrir classes exatamente com esses nomes. No entanto, a diferença entre uma classe e um tipo de entidade é que classes possuem dados e comportamentos, enquanto que tipos de entidade possuem apenas dados.

Idealmente, uma entidade deveria ser normal, descrevendo de forma coesa uma informação do mundo real. Uma entidade normalmente descreve um conceito, tal como uma classe coesa modela um conceito. Por exemplo, cliente e venda são claramente dois conceitos diferentes, portanto, faz sentido modelá-los como entidades diferentes.

### **Identificar Atributos**

Cada tipo de entidade terá um ou mais atributos de dados. Por exemplo, na Figura 1 podemos ver que a entidade Cliente possui atributos como Primeiro Nome e Sobrenome e na Figura 2 que a tabela TCLIENTE possui colunas de dados correspondentes CLI\_PRIMEIRO\_NOME e CLI\_SOBRE-NOME (uma coluna é a implementação de um atributo de dados em um banco de dados relacional).

Atributos devem ser coesos do ponto de vista do domínio da aplicação. Na Figura 1 decidimos que queríamos modelar o fato de pessoas possuírem primeiro nome e sobrenome em vez de apenas um nome (ex: “Cláudio” e “Dias” VS. “Cláudio Dias”). Usar o nível de detalhe correto pode ter um impacto significativo no esforço de desenvolvimento e manutenção. Refatorar uma simples coluna de dados em várias colunas pode ser difícil, o que pode resultar em construir o sistema com elementos desnecessários e, portanto, provoca um maior custo de desenvolvimento e de manutenção do que realmente necessário.

### **Aplicar Convenções de Nome**

Sua organização deve dispor de normas e diretrizes aplicáveis à modelagem de dados, algo que você deve ser capaz de obter dos administradores da empresa (se não existir você deve fazer algum lobby para incluí-lo). Essas diretrizes devem incluir as convenções de nomenclatura para a modelagem lógica e física, as convenções de nomenclatura lógica devem ser focadas na capacidade de leitura de humanos, enquanto as convenções de nomenclatura física refletirão considerações técnicas. Você pode ver claramente que diferentes convenções de nomenclatura foram aplicadas nas Figuras 1 e 2.

A ideia básica é que desenvolvedores sigam um conjunto comum de padrões de modelagem em um projeto de software. Tal como é importante seguir convenções comuns de codificação, um código limpo que segue as diretrizes escolhidas é mais fácil de ser compreendido. Isso funciona da mesma forma para as convenções de modelagem de dados.

### **Identificar Relacionamentos**

No mundo real, entidades possuem relacionamentos entre elas. Por exemplo, clientes FAZEM compras, clientes MORAM EM endereços e itens de venda SÃO PARTE DAS vendas. Todos esses termos

em maiúsculo definem relacionamentos entre entidades. Os relacionamentos entre entidades são conceitualmente idênticos aos relacionamentos (associações) entre objetos.

A Figura 5 descreve um MLD parcial para um sistema de compra online. A primeira coisa a se notar são os vários estilos aplicados aos nomes dos relacionamentos e papéis – diferentes relacionamentos requerem diferentes abordagens. Por exemplo, o relacionamento entre Cliente e Venda possui dois nomes, compra e é comprado por, mesmo o relacionamento entre essas entidades sendo apenas um. Neste exemplo, tendo um segundo nome no relacionamento, a ideia seria especificar como ler o relacionamento em cada direção. O ideal seria colocar apenas um nome por relacionamento.

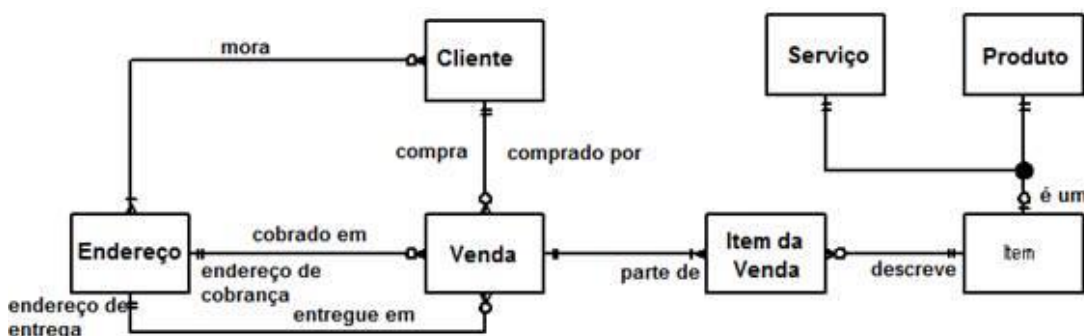


Figura 5. Um modelo lógico de dados (notações Engenharia da Informação)

Precisamos também identificar a cardinalidade e opcionalmente de um relacionamento (a UML combina os conceitos de cardinalidade e opcionalmente de um relacionamento em um conceito único de multiplicidade). Cardinalidade representa o conceito de “quantos” enquanto opcionalmente representa o conceito de “se é obrigatória a existência da entidade”. Por exemplo, não é suficiente saber que clientes fazem vendas. Quantas vendas um cliente pode realizar? Nenhuma, uma ou várias? Além disso, os relacionamentos existem nos dois sentidos: não apenas clientes fazem vendas, mas vendas são realizadas por clientes. Isso nos leva a questões como: quantos clientes podem ser envolvidos em uma dada venda e é possível ter uma venda com nenhum cliente envolvido? A Figura 5 mostra que clientes fazem uma ou mais vendas e que qualquer venda é realizada por um e somente um cliente. Ela também mostra que um cliente possui um ou mais endereços e que qualquer endereço possui zero ou mais clientes associados a ele.

Apesar de a UML distinguir entre diferentes tipos de relacionamentos – associações, hierarquia, agregação, composição e dependência – modeladores de dados normalmente não estão por dentro dessa questão. Subtipo, uma aplicação de hierarquia, é normalmente encontrada em modelos de dados. Agregação e composição são muito menos comuns, assim como dependências, que são tipicamente uma construção de software e, portanto, não aparecem no modelo de dados, a menos que tenhamos um modelo físico de dados bastante detalhado que mostre como views, triggers ou stored procedures dependem de outros aspectos do esquema do banco de dados.

### Associar Chaves

Existem duas estratégias fundamentais para associar chaves às tabelas. Primeiro, podemos associar uma chave natural que é um ou mais atributos de dados existentes que são únicos para o conceito do negócio. Imaginemos uma tabela Cliente, por exemplo. Ela possui de imediato duas chaves candidatas, as colunas NumeroCliente e CPF. A segunda forma é introduzindo uma nova coluna, chamada chave substituta, que é uma chave que não possui qualquer significado para o negócio. Um exemplo disso seria uma coluna idEndereco de uma tabela Endereco. Endereços não possuem uma chave natural “trivial” porque seria necessário usar todas as colunas da tabela Endereco para formar uma chave. Assim, introduzir uma chave substituta é uma opção muito melhor neste caso.

O debate entre “natural vs. substituta” é um das grandes questões religiosas na comunidade de banco de dados. O fato é que não existe estratégia perfeita, e com o tempo perceberemos que na prática algumas vezes fazem sentido usar chaves naturais e em outras situações é mais adequado o uso de chaves substitutas.

Normalizar para reduzir redundância de dados

Normalização de dados é um processo no qual atributos de dados em um modelos de dados são organizados para aumentar a coesão dos tipos de entidade. Em outras palavras, o objetivo da normalização de dados é reduzir e até eliminar redundância de dados, uma questão importante para desenvolvedores, pois é incrivelmente difícil armazenar objetos em um banco de dados relacional que mantém a mesma informação em vários lugares. A Tabela 2 resume as três principais regras de normalização descrevendo como aumentar os níveis de normalização em tipos de entidade.

Com respeito à terminologia, um esquema de dados é considerado estar em um nível de normalização do seu tipo de entidade menos normalizado. Por exemplo, se todos os tipos de entidade estão na segunda forma normal (2NF) ou maior, então dizemos que o esquema de dados está na 2NF.

Nível	Regra
Primeira Forma Normal (1NF)	Uma entidade está na 1NF quando ela não contém grupos de dados repetidos.
Segunda Forma Normal (2NF)	Uma entidade está na 2NF quando ela está na 1NF e quando todos seus atributos que não são chaves primárias são completamente dependentes de sua chave primária.
Terceira Forma Normal (3NF)	Uma entidade está na 3NF quando ele está na 2NF e quando todos seus atributos são diretamente dependentes da chave primária.

Tabela 2. Regras de normalização de dados

A Figura 6 descreve um esquema de banco de dados na ONF enquanto que a Figura 7 descreve um esquema normalizado na 3NF.

Por que normalização de dados? A vantagem de ter do esquema de dados altamente normalizado é que a informação é armazenada em um lugar apenas, reduzindo a possibilidade de dados inconsistentes. Além disso, esquemas de dados altamente normalizados em geral são conceitualmente mais próximos dos esquemas orientados a objeto, pois os objetivos da orientação a objetos de promover alta coesão e pouco acoplamento entre as classes resulta em soluções similares (ao menos do ponto de vista de dados). Isso geralmente torna mais simples mapear os objetos para o esquema de dados. Infelizmente, a normalização normalmente traz um custo para o desempenho. Com o esquema de dados da Figura 6 todos os dados para uma venda estão armazenados em uma linha (assumindo que vendas poderão ter até dois itens), simplificando o acesso. Com o esquema de dados da Figura 6 podemos rapidamente determinar a quantidade total de uma venda lendo uma única linha da tabela. Para fazer o mesmo com o esquema de dados da Figura 7, precisamos ler dados a partir de uma linha na tabela Venda, dados a partir de linhas na tabela ItemVenda para aquela venda e dados a partir das linhas correspondentes na tabela Item. Para esta consulta, o esquema de dados da Figura 6 provavelmente obtém melhor resultado.

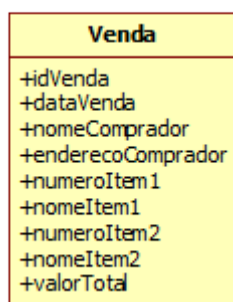


Figura 6. Um esquema de dados inicial para Venda (notação UML)

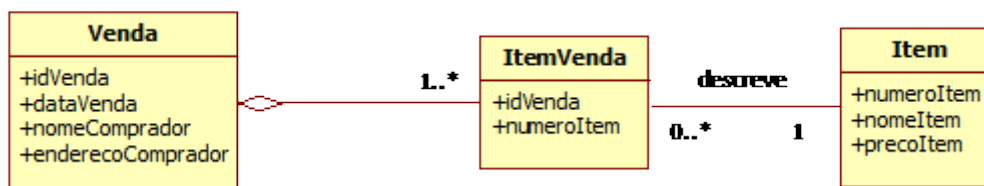


Figura 7. Um esquema normalizado em 3NF (notação UML)



## Modelo de Entidades e Relacionamentos

O **Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER)**, como o nome sugere, é um modelo conceitual utilizado na **Engenharia de Software** para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).

Em geral, este modelo representa de forma abstrata a estrutura que possuirá o **banco de dados** da aplicação. Obviamente, o banco de dados poderá conter várias outras entidades, tais como chaves e tabelas intermediárias, que podem só fazer sentido no contexto de **bases de dados relacionais**.

**Observação:** Nem sempre criaremos modelos para um sistema completo, pois isso poderia resultar em um modelo muito extenso e difícil de interpretar. Dependendo da magnitude do que estaremos desenvolvendo, podemos criar modelos apenas para uma parte do sistema, um módulo, ou mesmo uma funcionalidade. Imagine, por exemplo, um sistema ERP de grande porte que contemple vendas, finanças, recursos humanos, etc. Várias entidades estão presentes em mais de uma parte do sistema, mas não seria muito interessante, e provavelmente nem mesmo necessário, criar um único modelo para todo o sistema, por isso pode-se dividir a modelagem em várias partes menores.

### Entidades

Os objetos ou partes envolvidas em um domínio, também chamados de entidades, podem ser classificados como físicos ou lógicos, de acordo com sua existência no mundo real. Entidades físicas: são aquelas realmente tangíveis, existentes e visíveis no mundo real, como um cliente (uma pessoa, uma empresa) ou um produto (um carro, um computador, uma roupa). Já as entidades lógicas são aquelas que existem geralmente em decorrência da interação entre ou com entidades físicas, que fazem sentido dentro de um certo domínio de negócios, mas que no mundo externo/real não são objetos físicos (que ocupam lugar no espaço). São exemplos disso uma venda ou uma classificação de um objeto (modelo, espécie, função de um usuário do sistema).

As entidades são nomeadas com substantivos concretos ou abstratos que representem de forma clara sua função dentro do domínio. Exemplos práticos de entidades comuns em vários sistemas são Cliente, Produto, Venda, Turma, Função, entre outros.

Podemos classificar as entidades segundo o motivo de sua existência:

**Entidades fortes:** são aquelas cuja existência independe de outras entidades, ou seja, por si só elas já possuem total sentido de existir. Em um sistema de vendas, a entidade produto, por exemplo, independe de quaisquer outras para existir.

**Entidades fracas:** ao contrário das entidades fortes, as fracas são aquelas que dependem de outras entidades para existirem, pois individualmente elas não fazem sentido. Mantendo o mesmo exemplo, a entidade venda depende da entidade produto, pois uma venda sem itens não tem sentido.

**Entidades associativas:** esse tipo de entidade surge quando há a necessidade de associar uma entidade a um relacionamento existente. Na modelagem Entidade-Relacionamento não é possível que um relacionamento seja associado a uma entidade, então tornamos esse relacionamento uma entidade associativa, que a partir daí poderá se relacionar com outras entidades. Para melhor compreender esse conceito, tomemos como exemplo uma aplicação de vendas em que existem as entidades Produto e Venda, que se relacionam na forma muitos-para-muitos, uma vez que em uma venda pode haver vários produtos e um produto pode ser vendido várias vezes (no caso, unidades diferentes do mesmo produto). Em determinado momento, a empresa passou a entregar brindes para os clientes que comprassem um determinado produto. A entidade Brinde, então, está relacionada não apenas com a Venda, nem com o Produto, mas sim com o item da venda, ou seja, com o relacionamento entre as duas entidades citadas anteriormente. Como não podemos associar a entidade Brinde com um

relacionamento, criamos então a entidade associativa "Item da Venda", que contém os atributos identificadores das entidades Venda e Produto, além de informações como quantidade e número de série, para casos específicos. A partir daí, podemos relacionar o Brinde com o Item da Venda, indicando que aquele prêmio foi dado ao cliente por comprar aquele produto especificamente.

Mais adiante veremos um exemplo prático onde poderemos observar a existência dessas entidades de forma mais clara.

#### Relacionamentos

Uma vez que as entidades são identificadas, deve-se então definir como se dá o relacionamento entre elas. De acordo com a quantidade de objetos envolvidos em cada lado do relacionamento, podemos classificá-los de três formas:

**Relacionamento 1.1 (um para um):** cada uma das duas entidades envolvidas referencia obrigatoriamente apenas uma unidade da outra. Por exemplo, em um banco de dados de currículos, cada usuário cadastrado pode possuir apenas um currículo na base, ao mesmo tempo em que cada currículo só pertence a um único usuário cadastrado.

**Relacionamento 1.n ou 1..\* (um para muitos):** uma das entidades envolvidas pode referenciar várias unidades da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada a uma unidade da outra entidade. Por exemplo, em um sistema de plano de saúde, um usuário pode ter vários dependentes, mas cada dependente só pode estar ligado a um usuário principal. Note que temos apenas duas entidades envolvidas: usuário e dependente. O que muda é a quantidade de unidades/exemplares envolvidas de cada lado.

**Relacionamento n..n ou \*.\* (muitos para muitos):** neste tipo de relacionamento cada entidade, de ambos os lados, podem referenciar múltiplas unidades da outra. Por exemplo, em um sistema de biblioteca, um título pode ser escrito por vários autores, ao mesmo tempo em que um autor pode escrever vários títulos. Assim, um objeto do tipo autor pode referenciar múltiplos objetos do tipo título, e vice-versa.

Os relacionamentos em geral são nomeados com verbos ou expressões que representam a forma como as entidades interagem, ou a ação que uma exerce sobre a outra. Essa nomenclatura pode variar de acordo com a direção em que se lê o relacionamento. Por exemplo: um autor escreve vários livros, enquanto um livro é escrito por vários autores.

#### Atributos

Atributos são as características que descrevem cada entidade dentro do domínio. Por exemplo, um cliente possui nome, endereço e telefone. Durante a análise de requisitos, são identificados os atributos relevantes de cada entidade naquele contexto, de forma a manter o modelo o mais simples possível e consequentemente armazenar apenas as informações que serão úteis futuramente. Uma pessoa possui atributos pessoais como cor dos olhos, altura e peso, mas para um sistema que funcionará em um supermercado, por exemplo, estas informações dificilmente serão relevantes.

Os atributos podem ser classificados quanto à sua função da seguinte forma:

**Descritivos:** representam característica intrínsecas de uma entidade, tais como nome ou cor.

**Nominativos:** além de serem também descritivos, estes têm a função de definir e identificar um objeto. Nome, código, número são exemplos de atributos nominativos.

**Referenciais:** representam a ligação de uma entidade com outra em um relacionamento. Por exemplo, uma venda possui o CPF do cliente, que a relaciona com a entidade cliente.

Quanto à sua estrutura, podemos ainda classificá-los como:

**Simples:** um único atributo define uma característica da entidade. Exemplos: nome, peso.

**Compostos:** para definir uma informação da entidade, são usados vários atributos. Por exemplo, o endereço pode ser composto por rua, número, bairro, etc.

Alguns atributos representam valores únicos que identificam a entidade dentro do domínio e não podem se repetir. Em um cadastro de clientes, por exemplo, esse atributo poderia ser o CPF. A estes chamamos de Chave Primária.

Série: **MVC e Regras de negócio.**

Já os atributos referenciais são chamados de Chave Estrangeira e geralmente estão ligados à chave primária da outra entidade. Estes termos são bastante comuns no contexto de bancos de dados. Mantendo o exemplo anterior, a entidade cliente tem como chave primária seu CPF, assim, a venda possui também um campo “CPF do cliente” que se relaciona com o campo CPF da entidade cliente.

Diagrama Entidade Relacionamento

Enquanto o MER é um modelo conceitual, o Diagrama Entidade Relacionamento (Diagrama ER ou ainda DER) é a sua representação gráfica e principal ferramenta. Em situações práticas, o diagrama é tido muitas vezes como sinônimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstrato demais para auxiliar no desenvolvimento do sistema. Dessa forma, quando se está modelando um domínio, o mais comum é já criar sua representação gráfica, seguindo algumas regras.

O diagrama facilita ainda a comunicação entre os integrantes da equipe, pois oferece uma linguagem comum utilizada tanto pelo analista, responsável por levantar os requisitos, e os desenvolvedores, responsáveis por implementar aquilo que foi modelado.

Em sua notação original, proposta por Peter Chen (idealizador do modelo e do diagrama), as entidades deveriam ser representadas por retângulos, seus atributos por elipses e os relacionamentos por losangos, ligados às entidades por linhas, contendo também sua cardinalidade (1..1, 1..n ou n..n). Porém, notações mais modernas abandonaram o uso de elipses para atributos e passaram a utilizar o formato mais utilizado na UML, em que os atributos já aparecem listados na própria entidade. Essa forma torna o diagrama mais limpo e fácil de ser lido.

Observe na **Figura 1** um exemplo simples de um diagrama para um sistema de imobiliárias.

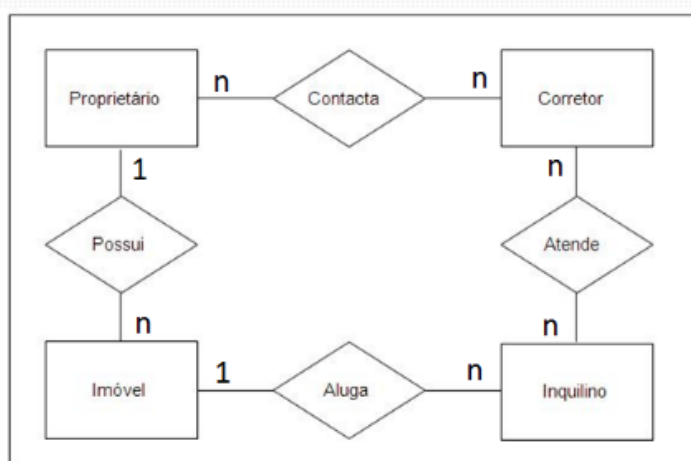


Figura 1. Diagrama Entidade Relacionamento de sistema de imobiliária



No domínio representado pelo diagrama acima temos as seguintes entidades e relacionamentos:

Proprietário contata Corretor (um proprietário pode contatar vários corretores e um corretor pode ser contactado por vários proprietários).

Corretor atende Inquilino (um corretor pode atender vários inquilinos e um inquilino pode ser atendido por vários corretores).

Inquilino aluga Imóvel (um inquilino aluga um imóvel e um imóvel pode ser alugado por vários inquilinos).

Proprietário possui Imóvel (um proprietário possui vários imóveis e um imóvel pertence a apenas um proprietário).

Uma variante da **Figura 1** pode ser vista na **Figura 2**, onde a cardinalidade do relacionamento é exibida junto do losango.

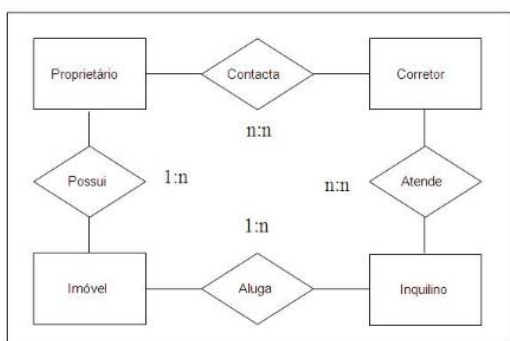


Figura 2. Diagrama de Entidade Relacionamento (variação)

Uma outra variação já mostra a cardinalidade de uma forma mais completa, deixando claro as possibilidades de números de objetos envolvidos em cada relacionamento. Nesse modelo, em cada lado do relacionamento os números aparecem no formato (X, Y) ao invés de um único número como vemos nas figuras anteriores. A **Figura 3** ilustra um exemplo desse tipo.



Figura 3. Diagrama Entidade Relacionamento (variação 2)

Neste diagrama, lemos os relacionamentos da seguinte forma:

1 ou 1 grupo possui 0 ou muitos produtos. Como de um lado temos “1 ou 1”, isso equivale a apenas “1”, pois não temos várias possibilidades. Já do lado do produto, indicamos que um grupo pode possuir nenhum produto, mas também pode possuir vários.

0 ou várias vendas contém 1 ou muitos produtos. Ou seja, um produto pode nunca ser vendido (0 vendas) como também pode ser vendido várias vezes (n vendas). Já uma venda deve conter 1 ou vários produtos, pois uma venda não pode estar vazia (0 produtos).

Os atributos, como já foi dito, podem aparecer no diagrama na forma de elipses ligadas às entidades. Essa foi a notação original proposta, mas como podemos ver na **Figura 4**, ela deixa o diagrama com muitos itens e pode atrapalhar um pouco a organização destes

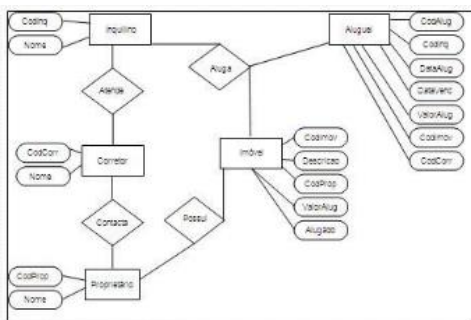


Figura 4. Atributos apresentados como elipses

Em uma notação mais atual, comumente utilizada na UML, os atributos aparecem listados dentro do próprio retângulo da entidade, enquanto o nome da entidade aparece no topo na forma de título. Na Figura 5 temos um exemplo.

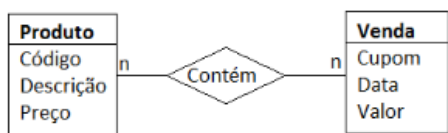


Figura 5. Diagrama com atributos nas entidades

Ferramentas CASE

Do inglês Computer-Aided Software Engineering, as chamadas **ferramentas CASE** são aquelas baseadas em computadores (softwares) utilizadas na Engenharia de Software para auxílio nas atividades desde análise de requisitos até, **modelagem de dados**.

No contexto desse artigo, as **ferramentas CASE** permitem a criação de diagramas de forma simples em um ambiente de fácil utilização e com recursos para incluir as principais regras de composição dos diagramas. Exemplos comuns desse tipo de ferramenta são: **Star UML**, **Astah** e **ERwin Data Modeler**. Na Figura 6 vemos um exemplo de diagrama sendo construído no Astah.

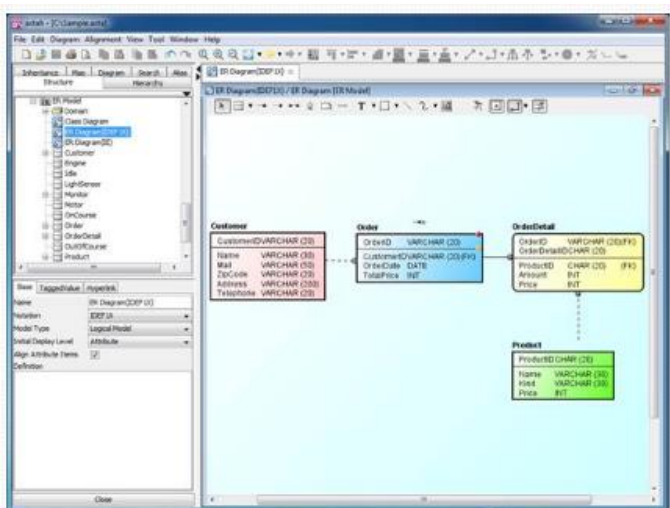


Figura 6. Diagrama no Astah Community

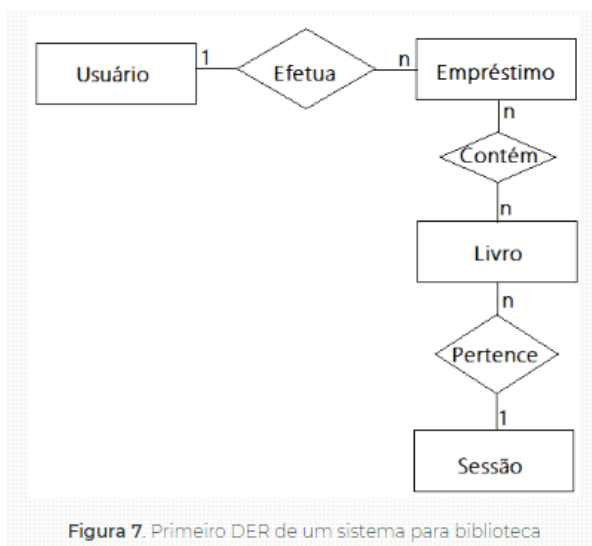
Além dessas ferramentas específicas, alguns IDEs (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado) como o Visual Studio e ferramentas de **gerenciamento de ban-**

cos de dados como **SQL Server Management Studio** possuem funcionalidades para criar diagramas facilmente e já gerar o código equivalente (SQL para criação das tabelas, chaves e relacionamentos, por exemplo).

Para fixar tudo que foi visto ao longo deste artigo, vamos agora desenvolver um pequeno exemplo prático em que modelaremos um sistema de bibliotecas, focando especificamente no empréstimo de livros.

Primeiramente precisamos identificar as entidades envolvidas nesse contexto. Sabemos que as entidades físicas existentes são o Usuário da biblioteca e o Livro que será emprestado. Além disso, consideraremos aqui que o livro pertence a uma Sessão, que ajuda na organização das obras do acervo. Em um sistema real pode haver outras informações sobre o livro, mas para esse exemplo a sessão é o bastante. Por fim, temos a entidade lógica Empréstimo, que tanto está relacionada com o usuário, quanto com o livro.

Assim já podemos esboçar nosso primeiro diagrama, simples, contendo as principais entidades e o relacionamento entre elas (**Figura 7**).



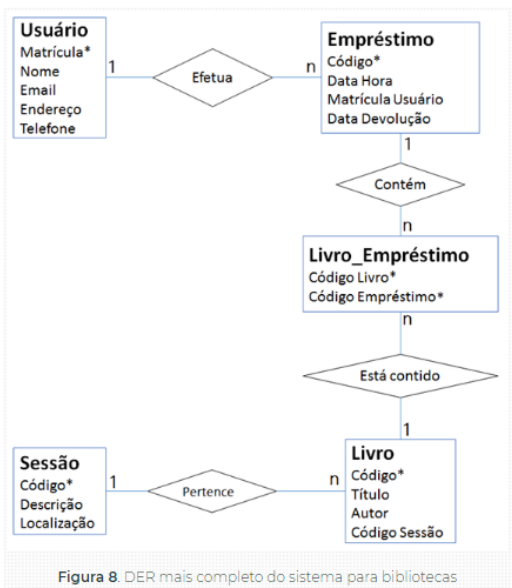
Neste primeiro diagrama podemos identificar alguns dos conceitos vistos:

Entidades fortes: Usuário, Livro e Sessão;

Entidades fracas: Empréstimo;

Relacionamentos: um Usuário efetua vários Empréstimos, vários Empréstimos contêm vários Livros, vários Livros pertencem a uma Sessão.

Agora que visualizamos o domínio no diagrama, podemos adicionar os atributos e outras entidades que se façam necessárias. Assim, passamos à **Figura 8**,



Neste ponto cabe fazer algumas observações importantes:

Especificamos os atributos de cada entidade e marcamos algumas delas com um asterisco, indicando que aquela é a chave primária da tabela, ou seja, um atributo único, que nunca poderá se repetir entre as entidades do mesmo tipo. Note que neste momento ainda não é necessário especificar o tipo de cada atributo (texto, número, data, etc.), isso só será necessário mais adiante, quando já estivermos planejando o banco de dados da aplicação.

Surgiu a entidade associativa Livro\_Empréstimo, que representa os livros contidos em um empréstimo (considerando um empréstimo contém vários livros e um livro pode estar contido em vários empréstimos). Esta entidade é composta pelas chaves das duas entidades principais. Se fosse necessário, nesta entidade também poderíamos adicionar informações complementares como quantidade (não se aplica neste caso, mas caberia em um sistema de vendas, por exemplo) e observações sobre o item.

Na entidade associativa, o relacionamento n..n foi dividido em dois relacionamentos do tipo 1..n, agora lidos da seguinte forma: um empréstimo contém vários itens, mas um item só pode estar contido em um único empréstimo (restrito pelas chaves primárias); um livro pode estar contido em vários itens de empréstimo (ser emprestado várias vezes), mas cada item refere-se a um único livro.

O **Modelo Entidade Relacionamento** (e principalmente o diagrama) é uma importante ferramenta durante o desenvolvimento de sistemas, principalmente aqueles mais complexos e difíceis de visualizar sem uma análise mais aprofundada.

A correta modelagem auxilia no correto **desenvolvimento da base de dados** e evita que várias alterações sejam necessárias para corrigir erros de concepção provenientes de falhas durante a análise, ou ainda por problemas de comunicação entre os membros da equipe.

O Modelo Entidade-Relacionamento (modelo relacional) foi definido por Peter Chen em 1976, e teve como base a teoria relacional criada por E.F.Cood (1970). Segundo Chen, a visão de uma dada realidade, baseia-se no relacionamento entre conceitos desta realidade, os quais retratam os fatos que governam esta mesma realidade, e que cada conceito (entidade ou relacionamento) pode possuir atributos (qualificadores desta realidade).

O conceito de abstração permite ao analista separar da realidade em estudo, as partes que são realmente relevantes para o desenvolvimento do sistema de informações e excluir da modelagem todos os aspectos que não exercem influência sobre o ambiente a ser modelado.

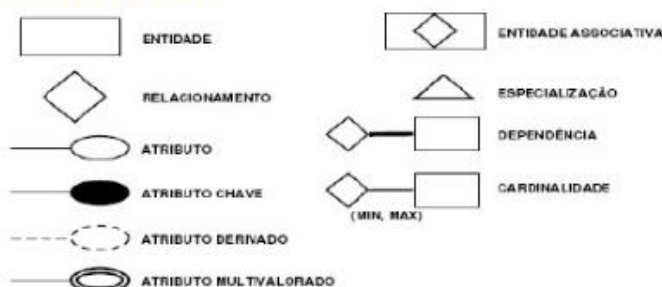
O objetivo da modelagem de dados é possibilitar a apresentação de uma visão única não redundante e resumida dos dados de um problema. Também nos ajuda a entender a estrutura e o significado destes dados.

No desenvolvimento de aplicações em banco de dados, o Modelo Entidade Relacionamento (E-R) é o mais largamente utilizado para a representação e entendimento dos dados que compõem a essência de um problema.

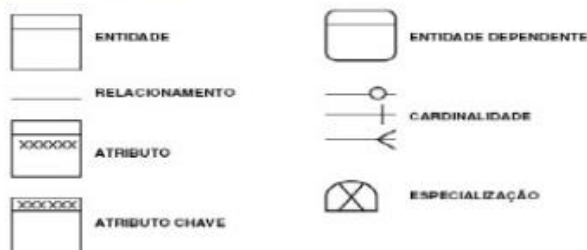
A Modelagem de Dados é amplamente utilizada como meio de conhecer os problemas organizacionais e projetar soluções.

**EXEMPLOS DAS NOTAÇÕES MAIS CONHECIDAS NO MODELO ENTIDADE RELACIONAMENTO**

**Notação Peter Chen**



**Notação James Martin**



**MODELO CONCEITUAL**

Ao se utilizar Modelagem Conceitual de dados com a técnica de Entidade e Relacionamentos, obteremos resultados e esquemas puramente conceituais sobre a essência de um problema ou negócio para o qual estamos desenvolvendo um projeto, não representando-se procedimentos ou fluxo de dados existentes.

Chen dedicou-se a destacar a importância de reconhecer os objetos que compõem um problema ou negócio, independentemente de preocupar-se com formas de tratamento das informações, procedimentos, programas, etc.

Estes objetos que desejamos conhecer e modelar para um sistema, Chen classificou em dois grupos : Entidades e Relacionamentos.

A Modelagem acontece em 3 fases:

1. Listar os Requisitos de Informação – levantamento de informações junto ao cliente.
2. Construir modelos para cada elemento

3. Escolha da tecnologia a ser utilizada – converter os modelos em sistemas

Ou seja a Modelagem Conceitual de Dados se caracteriza por ser **Independente de tipo de SGBD e Programas a serem desenvolvidos**. Ou seja, nesta fase, não nos preocupamos muito em como será feita a implementação dos programas, serviços web, componentes ou páginas do sistema.

O modelo Entidade Relacionamento propõe que a realidade seja visualizada sob três pontos de vista:

1 – Os objetos que compõe a realidade

2 - os tipos de informação ou características que se deseja conhecer sobre os objetos que compõem a

realidade e

3 - a forma como estes objetos se relacionam entre si.

Desta forma, o Modelo Entidade-Relacionamento é composto por três conceitos: Entidade, Atributo e Relacionamento. Para representação destes conceitos, existe o Diagrama Entidade Relacionamento.

### DIAGRAMA ENTIDADE RELACIONAMENTO

#### • Técnica mais difundida de modelagem conceitual

– Abordagem entidade-relacionamento (ER)

É um diagrama que descreve as entidade e relacionamentos entre as entidades em um modelo de dados.

• O Modelo conceitual é representado através de diagrama entidade-relacionamento

#### **Entidade:**

Define-se entidade como aquele objeto que existe no mundo real com uma identificação distinta e com um significado próprio.

• Representação de uma entidade.

#### **Propriedades de entidades**

• Entidade isoladamente não informa nada

• É necessário atribuir propriedades às entidades

• Propriedades especificadas na forma de: relacionamentos, atributos e generalizações/especializações

#### **ATRIBUTO**

É uma característica ou qualidade de uma entidade que tem valor para o negócio, que deve ser observada pelo usuário.

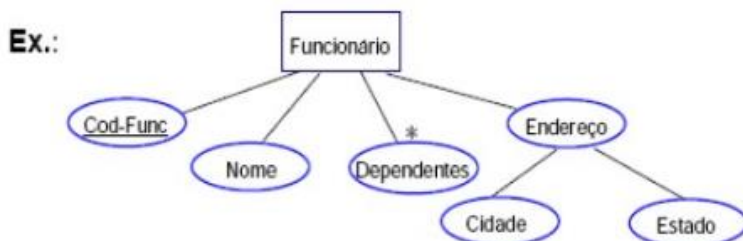
exemplo: Em um cadastro de clientes de uma empresa, podemos encontrar os seguintes atributos de cliente:

cpf, nome, endereço, cidade, bairro, estado, cep.

A definição dos atributos geralmente acontece durante várias discussões com o cliente. Todo atributo que pode vir a ser necessário em alguma tomada de decisão, no presente ou no futuro, deve ser incluído.

**Atributos (campos)**

**Atributo:** Elemento de dado que contém informação que descreve uma entidade



**Mas cuidado... evite armazenar dados desnecessários.**

Os atributos podem ser classificados de várias formas dependendo do tipo de informação que representam, obrigatoriedade de preenchimento do atributo, e composição do atributo.

**Atributo Monovalorado:** assume um único valor para cada elemento do conjunto x entidade

**Ex.:** Nome

**Atributo Composto:** formado por um ou mais sub-atributos. Quando nos deparamos com uma situação dessas, temos que analisar se deveremos fragmentar estes atributos em sub-atributos.

**Ex.:** Endereço (composto de rua, número, complemento, bairro, cep)

**Atributo Multivalorado:** uma única entidade tem diversos valores para este atributo (seu nome é sempre representado no plural)

**Ex.:** Dependentes

**Atributo Determinante:** identifica cada entidade de um conjunto-entidade (também conhecido como atributo chave)

**Ex.:** Cod\_Func

**Domínio de um Atributo:** conjunto de valores permitidos para o atributo.

**Ex.:** Sexo {M, F}, Estado Civil ( C, S, D )

**Atributo Obrigatório e Opcional**

Alguns atributos de uma entidade são obrigatórios outros são opcionais.

Nome é obrigatório pois toda pessoa deve ter um nome.

Telefone é opcional pois nem toda pessoa possui um telefone.

Para chegarmos a essa conclusão, temos que fazer uma análise minuciosa de campo a campo da(s) entidade(s) que estamos criando.

Então podemos fazer as seguintes definições:

**Atributo obrigatório** - é aquele que para uma instância de uma entidade ou relacionamento **deve** possuir um valor. (NOT NULL)

**Atributo opcional** - É aquele que para uma instância da entidade ou relacionamento **pode** possuir um valor. É o que chamamos de valor nulo. (NULL)

### Como escolher o nome, tipo de dado e tamanho de um atributo?

Os nomes dos atributos devem sempre ser significativos e descrever claramente o tipo de informação que o atributo representa.

O tipo de dado de um atributo depende da informação que deseja ser armazenada no atributo. O tipo de dado se for armazenar um dado que não fará parte de cálculos, um nome por exemplo, o tipo pode ser **texto**, se for armazenar valores numéricos que farão parte de cálculos o tipo será **numérico (inteiro,simples...)**, se for armazenar datas o tipo será **data** e assim por diante.

O tamanho do atributo define a quantidade de informação que você pode armazenar, assim: para um campo do tipo texto com tamanho 30 você poderá armazenar textos com no máximo 30 caracteres, um campo numérico de tamanho 2 armazena números com até dois dígitos.

### Tipos de dados:

#### Número

Smallint - Valor Inteiro de 15 bits com sinal

integer - Valor Inteiro de 31 bits com sinal

float(p) - formato científico com precisão

decimal(p,q) - formato decimal

#### Texto

char(n) - Cadeia de tamanho fixo de n caracteres

varchar(n) - cadeia de tamanho variável de n caracteres

long varchar - cadeia de caracteres de tamanho variável

#### Data/Hora

Date

time

timestamp

### Dicas para identificação de Entidades, Relacionamentos e Atributos.

Dado um texto descrevendo o banco de dados a ser projetado:

A presença de um substantivo usualmente indica uma entidade,

A presença de um verbo é uma forte indicação de um relacionamento,

Um adjetivo, que é uma qualidade, é uma forte indicação de um atributo,

Um advérbio temporal, qualificando o verbo, é uma indicação de um atributo do relacionamento.



## **Modelagem Da Informação Com Idef1x**

Os sistemas de informações estão presentes, cada vez mais, na vida de organizações e indivíduos. A qualidade destes sistemas depende dos sistemas de bancos de dados que os suportam. O projeto de um banco de dados de alta qualidade depende, por sua vez, da modelagem da informação, que trata da concepção da estrutura da informação em determinado universo de discurso (ambiente, empresa, negócio, etc.).

Reingruber & Gregory (1994) sustentam que a importância da modelagem da informação tem sido subestimada. Desenvolvedores de sistemas dedicam, tradicionalmente, maior atenção à modelagem dos processos que usam informações do que à estrutura subjacente.

Esta mentalidade vem mudando à medida em que os sistemas de informações crescem em tamanho e complexidade, evidenciando os problemas causados por modelagem da informação deficiente. Várias abordagens têm sido propostas para a modelagem da informação. Todas, de forma geral, têm base no modelo Entidade-Relacionamento de Peter Chen (1976).

Dentre as mais difundidas encontram-se NIAM (Nijssen & Halpin 1989), método CASE de Barker (1990), Engenharia da Informação (Martin & McClure 1985) e IDEF1X (NIST 1993). As abordagens citadas têm sido bastante utilizadas no projeto de bancos de dados relacionais.

Mais recentemente, técnicas baseadas na orientação a objeto têm sido propostas para a modelagem da informação para projeto de bancos de dados orientados a objeto: UML (Muller 1999) e IDEFobject (IEEE 1998), uma nova versão de IDEF1X.

Os bancos de dados relacionais têm um embasamento teórico sólido (Codd 1970), testado e aprovado durante quase três décadas.

Embora a orientação a objetos seja uma tecnologia mais moderna, a maior parte dos sistemas de informações em desenvolvimento ainda é baseada em bancos de dados relacionais, garantindo uma sobrevida longa a técnicas de modelagem da informação voltadas ao mundo relacional, como IDEF1X.

O PADRÃO IDEF1X IDEF1X é uma linguagem e método para modelagem da informação publicada como padrão norte-americano pelo NIST (1993). A linguagem já era usada, entretanto, antes da existência do padrão (Bruce 1992).

As seções a seguir apresentam as construções da linguagem: entidades, relacionamentos, chaves, atributos, e notas. As cinco fases do método IDEF1X para modelagem da informação são sumariamente descritas. A linguagem IDEF1X As construções sintáticas básicas da linguagem IDEF1X são: entidade, relacionamento e atributo.

Entidades, representadas graficamente como retângulos com uma linha divisória horizontal, são conjuntos de objetos de mesma natureza. Na figura 1, a entidade Produto representa um conjunto de instâncias ou ocorrências de produtos. Relacionamentos são associações significativas entre duas ocorrências de entidades.

São representados por linhas rotuladas com um verbo ou frase verbal, como em Nota Fiscal tem Item. Atributos são características das entidades, como ilustra a figura 1, onde endereço é um atributo de Cliente.

As entidades IDEF1X podem ser dependentes, representadas por retângulos com os cantos arredondados, ou independentes, representadas por retângulos com os cantos vivos. Uma entidade é dependente sempre que sua chave (também conhecida como chave primária) contém a chave completa de pelo menos uma outra entidade.

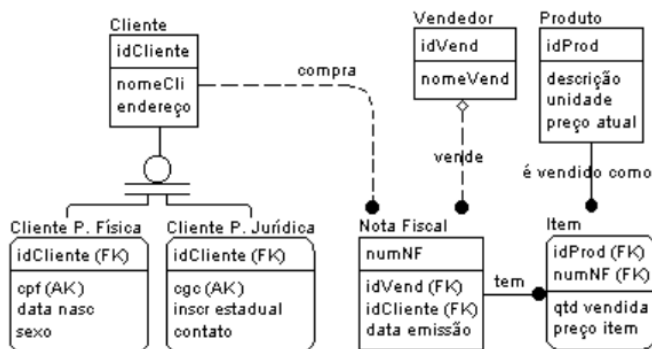


Figura 1 – Modelo IDEF1X ilustrando as construções básicas da linguagem

A chave de uma entidade é formada por um ou mais atributos cujo valor identifica uma ocorrência da entidade. Os atributos chaves são inscritos acima da divisão horizontal do retângulo que representa a entidade. Além da chave primária, IDEF1X descreve outros dois tipos de chave: estrangeira e alternada. Uma chave estrangeira é anotada pelo símbolo (FK) (foreign key), o qual denota a origem estrangeira do atributo, que é migrado a partir de outra entidade, onde é chave primária.

Na figura 1, idCliente é um atributo chave estrangeira em Nota Fiscal, migrado desde Cliente. Chave alternada é o atributo (ou conjunto de atributos) cujo valor identifica uma ocorrência da entidade, mas não foi escolhido como chave. Um atributo chave alternada é anotado pelo símbolo (AK) (alternate key), como no exemplo da figura 1, onde cgc é um identificador de Cliente P. Jurídica.

A migração do atributo chave de uma entidade e sua implementação como chave estrangeira em outra entidade respeita a regra de migração de chaves: todo relacionamento implica na migração da chave completa, e toda migração de chave é devida à existência de relacionamento. A forma como a migração ocorre varia segundo o tipo de relacionamento.

A figura 2 apresenta esquematicamente os tipos de relacionamento IDEF1X e suas representações gráficas. Relacionamentos podem ser específicos ou não-específicos.

Um relacionamento específico é aquele que, do ponto de vista de pelo menos uma das ocorrências de entidade envolvidas, tem uma outra ocorrência específica associada segundo o relacionamento. Relacionamentos não-específicos, por outro lado, são associações várias-paravárias, nas quais não é possível especificar uma (pois são várias) ocorrência associada.

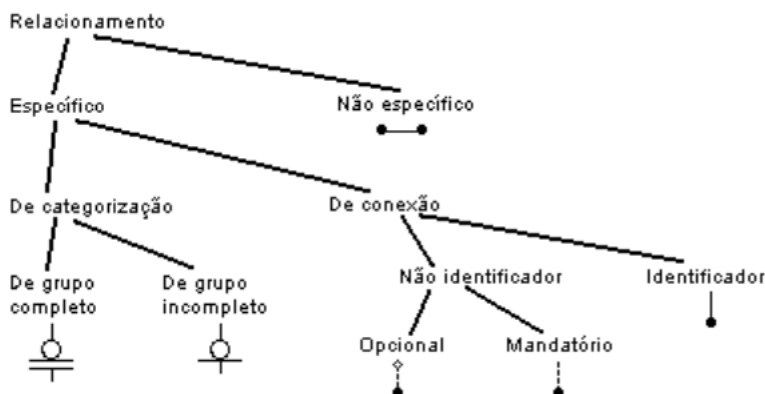


Figura 2 - Tipos de relacionamento na linguagem IDEF1X

Os relacionamentos específicos podem ser de categorização ou de conexão. Em um relacionamento de categorização, ocorrências de ambas entidades associadas representam o mesmo tipo de objeto real ou abstrato, sendo que a entidade genérica representa os atributos genéricos do objeto, enquanto cada entidade categoria reúne características de uma determinada especialização.

Nos relacionamentos de conexão, também chamados de conexão específica, qualquer número de ocorrências de uma entidade filha (child entity) associa-se a zero ou uma ocorrência de uma entidade genitora (parent entity).

Vários relacionamentos de categorização podem reunir entidades categorias em um grupo de categorização (category cluster), representado por uma circunferência sublinhada por um ou dois traços. Uma ocorrência da entidade genérica pode estar associada a apenas uma ocorrência de uma das entidades categorias do grupo (isto é, as categorias são excludentes).

Um grupo completo é anotado por uma linha dupla sob a circunferência que representa o grupo. Todas as categorias estão presentes, ou seja, uma ocorrência da entidade genérica está necessariamente associada a uma das categorias, como no caso da entidade genérica Cliente, na figura 1, onde qualquer ocorrência está associada a uma ocorrência da entidade categoria Cliente P.

Física ou (exclusivamente) Cliente P. Jurídica. Em um grupo incompleto, nem todas as categorias possíveis estão presentes. Uma ocorrência da entidade genérica pode ou não estar associada a uma das categorias. Os relacionamentos de conexão específica nos quais toda a chave da entidade genitora faz parte da chave da entidade filha são ditos identificadores.

O relacionamento Nota Fiscal tem Item, na figura 1, é um exemplo de relacionamento identificador, pois a chave de Nota Fiscal faz parte da chave de Item. Relacionamentos de conexão onde pelo menos um atributo chave da entidade genitora não faz parte da chave da entidade filha chamam-se não-identificadores, e são representados por linha pontilhada.

Podem ser mandatórios, quando toda ocorrência da entidade filha deve estar associada a uma ocorrência da entidade genitora, ou opcionais (representados por um pequeno losango junto à entidade genitora), nos casos nos quais é possível que uma ocorrência da entidade filha não esteja associada a uma ocorrência da entidade genitora.

Na figura 1, Cliente compra Nota Fiscal é um relacionamento mandatório, enquanto Vendedor vende Nota Fiscal é opcional – permite que haja nota fiscal sem vendedor associado. IDEF1X permite representar precisamente a cardinalidade dos relacionamentos, que diz respeito a quantas ocorrências podem estar associadas a uma ocorrência de entidade envolvida em determinado relacionamento.

Todo relacionamento deve ser considerado nos dois sentidos possíveis: da entidade genitora (ou genérica) para a filha (ou categoria), e vice-versa. No sentido da entidade filha (ou categoria) para a genitora (ou genérica), a cardinalidade é “exatamente uma”, com a exceção dos relacionamentos não-identificadores opcionais, onde a cardinalidade é “zero ou uma”.

No sentido inverso, a variedade é maior. A figura 3 apresenta a representação gráfica das várias cardinalidades possíveis para relacionamentos de conexão e seus significados. Em relacionamentos de categorização, a cardinalidade no sentido da entidade genérica para a categoria é sempre “zero ou uma”.

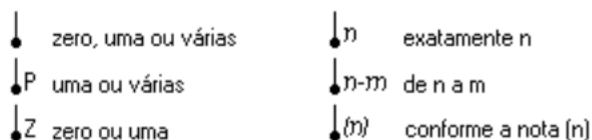


Figura 3 - Cardinalidade de relacionamentos de conexão

Restrições sobre a estrutura da informação que não são representáveis através dos elementos da linguagem recém descritos são representados através de notas.

Um exemplo é ilustrado na figura 3, onde um índice (n) junto à entidade filha referencia uma nota (que deve ser apresentada em português, indexada, em outro lugar no diagrama) que descreve uma cardinalidade não convencional. A notação de IDEF1X permite construir modelos precisos e detalhados.

Às qualidades da linguagem unem-se as virtudes do método IDEF1X, que reúne a experiência de muitos anos da indústria e forças armadas norte-americanas. O método IDEF1X é apresentado a seguir.

**O Método IDEF1X**

O padrão IDEF1X (NIST 1993) apresenta um guia de projeto para o desenvolvimento de um modelo da informação.

Este guia, ou método IDEF1X, propõe o desenvolvimento de um modelo da informação em cinco fases, com detalhamento iterativo e progressivo do modelo: • Fase 0 – Iniciação de projeto.

- Fase 1 – Definição de entidades.
- Fase 2 – Definição de relacionamentos.
- Fase 3 – Definição de chaves. •

Fase 4 – Definição de atributos. Na Fase 0 estabelecem-se os objetivos da modelagem (documentados na forma de uma declaração de propósito e escopo), desenvolve-se um plano (incluindo cronograma) de modelagem, organiza-se a equipe de projeto e coleta-se o material a ser usado como fonte de informação.

O material-fonte deve ser indexado e arquivado para futura referência. As figuras 4 e 5 representam o material-fonte coletado em um projeto-exemplo a ser ilustrado nesta seção: a modelagem da informação referente a notas fiscais de venda da Empresa Comercial Fictícia (ECF).

A figura 4 representa o material número 1 (nota fiscal de venda de mercadorias); a figura 5 ilustra o material número 2 (ficha de produto). A figura 6 apresenta os papéis existentes numa equipe de projeto IDEF1X: o gerente de projeto administra o projeto, contrata o modelador, identifica os peritos e fontes de informação adequadas, e preside o comitê de revisão de aceitação do modelo.

O modelador é responsável pela coleta de informações e concepção do modelo da informação, bem como pelo treinamento de revisores e documentação do modelo.

Fontes do modelo são pessoas e documentos que podem oferecer esclarecimento sobre o ambiente modelado. Peritos são pessoas cujo conhecimento e experiência são usados na validação do modelo concebido. O comitê de revisão de aceitação dá a palavra final sobre a qualidade e aceitabilidade do modelo. Para evitar conflito de interesses, o comitê não deve ter a participação de modeladores.

EMPRESA COMERCIAL FICTÍCIA (ECF)			
Rodovia BR 101 s/n - São José			
DATA:	31/08/99	Nº	123456
DESTINATÁRIO/REMETENTE			
Loja Sertão Ltda.			
CGC			
88122000/0001-11			
ENDEREÇO			
Rodovia SC 407 Km 4 São José			
DADOS DO PRODUTO			
CÓDIGO	DESCRIÇÃO	QTDE UN	PREÇO
1233	POLTR GIRAT X	2,00 un	98,00
1344	VENT TETO VT3	1,00 un	38,95
TOTAL			234,95

Figura 4 – Nota fiscal, material-fonte nº 1

Código:	1344
Descrição:	VENT TETO VT3
Unidade:	un
Preço:	41,00
Qtd estoque:	22 23 16

Figura 5 – Ficha de produto, material-fonte nº 2

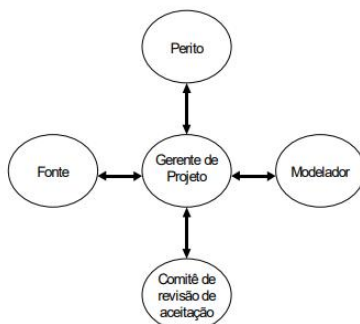


Figura 6 - Papéis numa equipe de projeto IDEF1X (NIST 1993)

As fases 1 e 2 tratam da estrutura da informação num nível de abstração alto. Na Fase 1, as entidades são identificadas e definidas.

Na Fase 2 as associações relevantes entre as entidades concebidas na fase anterior são identificadas e definidas, e um modelo gráfico de entidades e relacionamentos é desenhado.

As definições formam um glossário – são sentenças em português que definem o que é cada ocorrência de cada entidade ou relacionamento. A figura 7 ilustra, na forma de um conjunto de documentos conhecido como kit IDEF1X, os resultados das fases 0, 1 e 2 do projeto-exemplo.

Os documentos apresentados são: declaração de propósito e escopo, organização da equipe de projeto e lista do material-fonte (da Fase 0), lista e glossário de entidades (da Fase 1), e glossário de relacionamentos e diagrama de entidades e relacionamentos (da Fase 2).

O diagrama da Fase 2, ilustrado em separado na figura 8, representa apenas entidades e relacionamentos, sem especificar o tipo, uma vez que este detalhamento depende da especificação das chaves, assunto da Fase 3. A notação de relacionamento identificador é usada indistintamente. Relacionamentos não-específicos podem ser usados na Fase 2, como é o caso do existente entre as entidades Nota Fiscal e Produto, na figura 8.

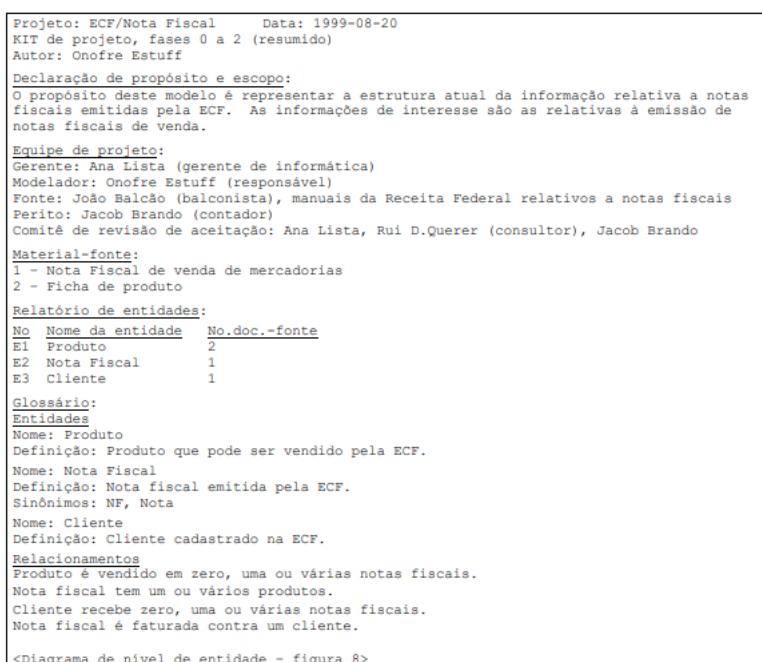


Figura 7 - Kit IDEF1X para o projeto-exemplo (fases 0 a 2)

Em um diagrama IDEF1X, cada relacionamento tem registrados seu nome e cardinalidade. O nome esclarece qual o caráter do relacionamento entre as ocorrências de entidade envolvidas e é representado por uma frase verbal anotada junto ao relacionamento. Nos casos de relacionamentos específicos, pode-se representar apenas a frase verbal no sentido da entidade genitora (lado específico do relacionamento) para a entidade filha. Na figura 8, por exemplo, representa-se a frase verbal recebe, do relacionamento Cliente – Nota Fiscal, e omitese a frase verbal é faturada contra um, do relacionamento Nota Fiscal – Cliente.

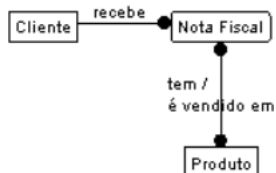


Figura 8 - Diagrama de entidades e relacionamentos do projeto-exemplo

Na Fase 3, os atributos chaves de cada entidade são identificados e definidos. Os tipos dos relacionamentos são detalhados, a migração de chaves e a resolução de relacionamentos não-específicos são executadas.

A resolução de relacionamentos não específicos gera novas entidades e relacionamentos, o que demanda uma atualização do projeto nas fases 1 e 2. Um glossário de atributos chaves é anexado à documentação.

Na Fase 4, os atributos não-chaves são identificados e definidos, e as definições são anexadas à documentação. Cada atributo deve ser único no modelo, isto é, deve ter nome único e ser próprio de apenas uma entidade. A figura 9 apresenta o detalhamento do projeto-exemplo iniciado nesta seção. Neste kit IDEF1X da Fase 4 há uma nova entidade, ItemNF, com dois novos relacionamentos específicos (com Nota Fiscal e Produto), derivados do relacionamento não-específico entre Nota Fiscal e Produto.

A figura 10 apresenta o diagrama completo para o projeto-exemplo, que faz parte do kit da Fase 4. Os kits são usados pelo comitê de revisão de aceitação para verificar e validar o modelo. Opcionalmente, ou adicionalmente, é conduzida uma revisão final, conhecida como procedimento walkthrough.

O modelo recebe um status que determina sua aceitação plena ou condicional, sua reconstrução, ou mesmo sua rejeição. A razão de se determinar a validade do modelo a partir da avaliação de uma equipe de peritos é que não existe um modelo absolutamente “correto”.

Uma concepção de modelagem pode ser considerada adequada quando atende ao que o comitê de revisão considera correto ou satisfatório. Um modelo da informação é desenvolvido para atender a um propósito (registrado na declaração de propósito e escopo da Fase 0), e não para representar “a Verdade”. Esta noção é explorada em maior profundidade na próxima seção.

```

Projeto: ECF/Nota Fiscal   Data: 1999-10-01
KIT de projeto, fases 3 e 4 (resumido)
Autor: Onofre Estuff

Glossário:
Entidades e atributos
Nome: Produto
Definição: Produto que pode ser vendido pela ECF.
Atributo id Produto: Código identificador de produto
Atributo descr: Descrição ou nome de produto
Atributo unidade: Unidade na qual é vendido o produto
Atributo qtde estoque: Quantidade em estoque do produto
Atributo preço atual: Preço atual do produto

Nome: Nota Fiscal
Definição: Nota fiscal emitida pela ECF.
Sinônimos: NF, Nota
Atributo num NF: Número identificador da nota
Atributo id Cliente: Código do cliente destinatário da nota
Atributo data emissão: Data da emissão da nota

Nome: Cliente
Definição: Cliente cadastrado na ECF.
Atributo id Cliente: Código identificador do cliente
Atributo nome Cliente: Nome do cliente
Atributo end Cliente: Endereço do cliente
Atributo cgc Cliente: Código identificador do cliente no CGC

Nome: ItemNF
Definição: Item de nota; produto vendido em uma nota.
Atributo num NF: Número da nota à qual se refere o item
Atributo id Produto: Código do produto vendido no item
Atributo qtde vendida: Quantidade vendida do produto no item
Atributo preço venda: Preço praticado na venda do produto neste item

Relacionamentos
<Legíveis no diagrama...>

<Diagrama completo - figura 10>
    
```

Figura 9 - Kit IDEF1X para um projeto-exemplo (completo)

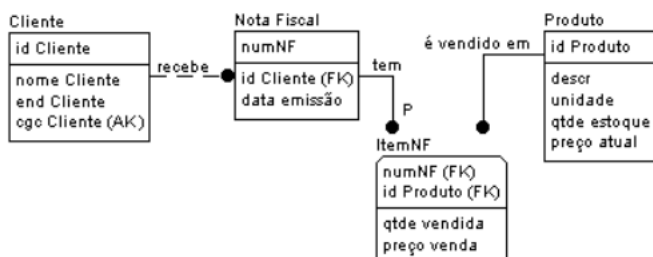


Figura 10 - Diagrama completo do projeto-exemplo

### O Melhor De Idef1x: O Princípio Do Consenso

Hay (1995) compara várias linguagens de modelagem, incluindo IDEF1X, quanto à completude técnica e legibilidade. Conclui declarando que não há a linguagem “correta”, e que o sucesso de um projeto de banco de dados depende mais da habilidade do projetista do que da notação escolhida.

A habilidade do modelador é, com efeito, mais importante do que a escolha da linguagem.

Mesmo assim, os estudos comparativos entre as diversas linguagens para modelagem da informação têm mérito, já que detalhes da sintaxe e semântica de cada linguagem podem favorecer ou prejudicar a qualidade de um modelo da informação. Entretanto, há um outro fator crítico para o sucesso de um projeto de banco de dados: a forma como é conduzida a avaliação e aprovação de um modelo.

IDEF1X, ao contrário das demais abordagens, define um método detalhado para ser usado em conjunto com a linguagem. O método é top-down, iterativo, progressivo, e tem como critério para a aprovação de um modelo o consenso de peritos na área de domínio, ou universo de discurso, do modelo. Um modelo não pode ser mais “correto” do que o consenso dos peritos.

Caso contrário, existiria uma verdade absoluta cuja modelagem seria de responsabilidade do modelador. Portanto, o resultado que se pode alcançar em um esforço de modelagem é a captura da estrutura da informação que representa as regras do negócio “corretas” segundo a avaliação dos peritos. Este princípio do consenso foi incorporado a IDEF1X a partir da experiência acumulada pela indústria e forças armadas norte-americanas em projetos de bancos de dados, muitos dos quais de grande tamanho, complexidade, e importância econômica e estratégica.

Ainda assim, nada assegura que o consenso de peritos na avaliação de um modelo representa garantia permanente de qualidade. É possível que mudem as regras do negócio, ou o consenso dos peritos. Portanto, é importante criar condições para a manutenção dos modelos. O glossário com as definições de entidades, relacionamentos e atributos, juntamente com a lista indexada de material-fonte e as declarações de propósito e escopo, são os documentos que permitem a qualquer modelador, participante ou não do esforço de modelagem original, compreender e manter um modelo.

O caráter relativo do consenso de peritos sobre a concepção de modelos de informação pode ser observado em outras modalidades de modelagem, não apenas de bancos de dados. Por exemplo, pode-se observar o mapa-múndi da figura 11 e fazer a pergunta de modelagem: “Quais são os continentes existentes?”

O dicionário Aurélio (Ferreira 1986) apresenta várias definições de continente. As acepções que nos interessam são: “continente. [...] 4. Grande massa de terra cercada pelas águas oceânicas. 5. Cada uma das grandes divisões da terra: Europa, Ásia, África, América e Oceania. 6. O corpo principal da Europa em relação às Ilhas Britânicas. 7. Bras., RS. Designação dada popularmente ao Rio Grande do Sul, desde os tempos coloniais até a revolução de 1835.”

Ao contrário de um glossário IDEF1X, um dicionário permite definições ambíguas e até conflitantes. A acepção 4 é genérica e, certamente, a menos tendenciosa. A acepção 5 impõe a existência de um “continente” europeu e ignora a existência da Antártida, que é maior do que a Europa e a Oceania. As acepções 6 e 7 são muito particulares, válidas em contextos específicos. De acordo com a acepção 4, a Europa não é um continente. Embora as definições 4 e 5 sejam contraditórias, costumam ser ensinadas nos bancos escolares como se fossem compatíveis, sem grandes protestos de geógrafos ou educadores.



Figura 11 – “Quais são os continentes?”





**Projeto Lógico, Projeto Físico e Particionamento de Tabelas, Normalização de Dados**

Neste momento o projetista descreve COMO DESENVOLVER os mecanismos de cada função e o relacionamento entre as funções. Nesta fase especificam-se integralmente os itens a serem construídos, as características, funcionalidades, regras de negócios e restrições. Neste ponto do desenvolvimento é possível ter uma visão final do projeto, embora ainda no papel. Esta visão deve ser validada junto ao cliente.

São Atividades/Sub Fases do Projeto Lógico

Revisar análise do sistema atual

Complementar e refinar.

Elaborar macro propostas

Levantar alternativas de macro solução, escolher a solução oportuna e revisar o planejamento.

Detalhar a lógica

Refinar requisitos funcionais reais das informações, elaborar os diagramas, descrever a lógica dos processos, elaborar dicionário de dados, esboçar documentos de entrada e saída, analisar as integrações com outros sistemas.

Definir estratégia de projeto físico

Definir tecnologia de base, rever impactos, elaborar planos logísticos, infra-estrutura e contingência, determinar planejamento e rever equipe, analisar custos, benefícios e viabilidades.

Aprovar o projeto lógico

Avaliar a qualidade, organizar informações, revisar fases anteriores, elaborar parecer e termo de compromisso, reunir e apresentar.

Mudanças em requisitos ocorrem em todas as fases do desenvolvimento do sistema.

As alterações nos requisitos são inevitáveis e não implicam em práticas pobres no processo de construção do sistema.

Embora a mudança seja inevitável, é usual o caso em que alguns requisitos são mais estáveis que outros.

Requisitos Estáveis

São concebidos com a essência de um sistema e seu domínio da aplicação, e mudam mais lentamente que requisitos voláteis.

Requisitos Voláteis

São específicos para a instanciação de um sistema em um ambiente particular e para um cliente particular e podem ser classificados em quatro tipos:

**Requisitos Mutáveis**

São os requisitos que mudam em função de mudanças no ambiente no qual o sistema opera. Por exemplo, os requisitos para um sistema que calcula taxas de dedução que evolui conforme as leis de taxação mudam.

**Requisitos Emergentes**

São os requisitos que não podem ser completamente definidos quando o sistema é especificado, mas que emergem quando o sistema está projetado e implementado. Por exemplo, pode não ser possível especificar de antemão os detalhes de como a informação será exibida. Conforme os stakeholders

veem exemplos de apresentações possíveis, eles podem pensar em novas maneiras de exibição da informação que seria útil para eles.

### **Requisitos Consequentes**

São os requisitos baseados em suposições de como o sistema será utilizado. Quando o sistema é posto em uso, algumas destas suposições podem estar erradas. Usuários irão adaptar-se ao sistema e encontrar novas maneiras de usar suas funcionalidades, o que irá resultar em demandas dos usuários para mudanças no sistema.

### **Requisitos de Compatibilidade**

São os requisitos que dependem de outro equipamento ou processo. Conforme muda esse equipamento, os requisitos também mudam.

### **Validação de Requisitos**

Os produtos de trabalho criados como consequência da engenharia de requisitos (uma especificação dos requisitos do sistema e informações relacionadas) devem ser validados quanto à qualidade durante o passo de validação de requisitos. Esta validação examina a especificação para garantir que todos os requisitos do sistema foram estruturados de maneira não ambígua, que as inconsistências, omissões e erros foram apagados e corrigidos, e que os produtos de trabalho estão em conformidade com os padrões estabelecidos para o processo, para o projeto e para o produto.

O mecanismo primário de validação de requisitos é a revisão técnica formal. O time de revisão inclui os engenheiros de sistema, clientes, usuários e outros stakeholders que examinam a especificação do sistema à procura de erros de conteúdo ou interpretação, pontos onde pode ser necessário esclarecimento, perda de informações, inconsistências (um dos maiores problemas da engenharia de grandes produtos), requisitos conflitantes, ou requisitos irreais (de desenvolvimento impossível).

Embora a revisão para validação dos requisitos possa ser conduzida de qualquer maneira desde que possibilite a descoberta de erros nos requisitos

### **Projeto Físico**

É a implementação física do projeto no ambiente empresarial. Execução ou realização do projeto de sistemas de informação. Contempla a confecção de programas (em software e manuais) e seus respectivos controles de testes e qualidade, bem como o layout final das entradas e saídas.

Subfases:

Especificar o modelo de dados: reestruturar os dados, eliminar redundâncias, analisar as dependências funcionais;

Especificar o modelo de processos: identificar e reestruturar os processos (tarefas) do sistema;

Definir arquitetura e plano de segurança:

Definir os arquivos físicos e métodos de acesso do software.

Definir procedimentos do plano de segurança das informações (Backup).

Construir o sistema de informação: analisar as linguagens de programação e executar o sistema ou programar o software.

Finalizar o sistema de informação: testes dos módulos, dos programas e arquivar os resultados. Aprovar o projeto físico.

### **Projeto de Software Lógico x Físico**

- Projeto lógico (independente de implementação) é executado para produzir um projeto que poderia ser implementado em diferentes plataformas (hardware, linguagem de programação, SGBD)
- Projeto físico (implementação específica) é executado para produzir um projeto que é específico para a plataforma escolhida

- Algumas vezes se a plataforma é conhecida quando começa o projeto, não haverá o estágio de projeto lógico.

Entendemos que cada empresa possui necessidades e características específicas, por isso disponibilizamos o que há de mais moderno em tecnologia da informação e consultores especializados para desenvolver de forma ágil sistemas customizados pra cada cliente.

Contamos com uma equipe de desenvolvimento de sistemas altamente qualificada em todas as fases de um projeto. As ferramentas de desenvolvimento, bem como os sistemas elaborados, podem ser vistos em nossos escritórios e/ou nas instalações de nossos clientes.

Nosso portfólio de projetos de grande porte em diversos segmentos gerou um forte conhecimento de regras de negócio.

### **Desenvolvimento de Sistemas**

Os projetos são iniciados através de "workshops", onde são utilizadas técnicas de JRP - Joint Requirements Planning e JAD - Joint Application Design, atingindo rapidamente os resultados esperados. Os "workshops" utilizam ainda, técnicas de modelagem de dados, análise essencial (eventos) e projeto orientado a objeto, compondo a base da Metodologia de Desenvolvimento de Projetos.

Desenvolvimento de Projetos Internet / Intranet

Elaboração de projetos de Intranet para a empresa e/ou departamentos.

Elaboração de projetos de comércio eletrônico: business to business, business to consumer.

Desenvolvimento de sites.

Alguns Projetos

Sistema Budget

Sistema GPS - Controle e gerenciamentos da solução de problemas

Sistema de Abertura de Ficha

Sistema Sales Force Automation (WEB/Mobile)

Sistema Portal do Distribuidor

Sistema Plano de Negócios

Sistema ROTULUS

Sistema Concentrador Fiscal

Metodologia de Gerenciamento de Projetos

A metodologia consiste em uma simplificação do PMBok, visando tornar o processo mais objetivo e Eficiente. As etapas componentes da metodologia são:

### **Elementos na Implantação**

Tecnologia: Recursos tecnológicos

Pessoas: Trabalho em equipe

Objetivos: Claros e realistas

A metodologia consiste em uma simplificação do PMBok, visando tornar o processo mais objetivo e eficiente. As etapas componentes da metodologia são:

Levantamento

Planejamento do Projeto

Preparação do Projeto

Treinamento e Validação de Funcionalidades

Simulações

Disponibilização e Acompanhamento

Projeto Físico

No desenvolvimento do projeto físico, o protótipo e a especificação são transformados em código de programação. Após a transformação, são feitos testes para se avaliar a qualidade das aplicações geradas nesta fase. Para cada item construído deve ser realizada uma bateria de testes com objetivo de identificar não-conformidades face às especificações (Regras de Negócios e Funcionalidades) e face aos padrões correspondentes às tecnologias utilizadas.

O objetivo principal desta fase é, ao seu final, garantir que todos os itens foram construídos conforme definidos pelo analista de sistemas, estando a aplicação apta a ser homologada pelo cliente. O sucesso desta fase está diretamente ligado à validação correta da fase anterior. Ao final desta fase, o cliente já terá a primeira versão do sistema.

São Atividades/Sub Fases do Projeto Lógico:

Revisar projeto lógico

Complementar e refinar.

Especificar o modelo de dados

Finalizar dicionário de dados, elaborar o modelo de dados, normalizar depósitos de dados.

Definir arquitetura

Definir arquivos físicos e métodos de acesso, definir procedimentos de segurança.

Construir o sistema

Finalizar as entradas e saídas do sistema, executar o sistema ou programar o software, construir programas paralelos.

Finalizar o sistema

Elaborar testes dos programas, definir fluxos e procedimentos operacionais, complementar a documentação.

Definir estratégia de projeto de implantação

Esboçar projeto de implantação, planejar treinamento, elaborar plano de conversão.

Aprovar o projeto físico

Avaliar a qualidade, organizar informações, revisar fases anteriores, elaborar parecer e termo de compromisso, reunir e apresentar.

Você pode criar uma tabela ou um índice particionado no SQL Server 2017 usando o SQL Server Management Studio ou o Transact-SQL. Os dados das tabelas e dos índices particionados são divididos horizontalmente em unidades que podem ser disseminadas por mais de um grupo de arquivos em um banco de dados. O particionamento pode tornar as tabelas e os índices grandes mais gerenciáveis e escalonáveis.

A criação de uma tabela ou um índice particionado ocorre normalmente em quatro partes:

Crie um grupo ou grupos de arquivos e os arquivos correspondentes que terão as partições especificadas pelo esquema de partição.

Cria uma função de partição que mapeia as linhas de uma tabela ou de um índice em partições com base nos valores de uma coluna especificada.

Cria um esquema de partição que mapeia as partições de uma tabela particionada ou índice para os novos grupos de arquivos.

Crie ou modifique uma tabela ou um índice e especifique o esquema de partição como local de armazenamento.

Antes de Começar

### **Limitações e Restrições**

O escopo de uma função e de um esquema de partição é limitado ao banco de dados em que foram criados. No banco de dados, as funções de partição residem em um namespace separado das outras funções.

Se alguma linha dentro de uma função de partição tiver colunas de particionamento com valores nulos, essas linhas serão alocadas para a partição à extrema esquerda. No entanto, se NULL for especificado como valor de limite e RIGHT for indicado, a partição extrema esquerda permanecerá vazia e os valores NULL serão colocados na segunda partição.

### **Segurança**

Permissões

A criação de uma tabela particionada requer a permissão CREATE TABLE no banco de dados e a permissão ALTER no esquema no qual a tabela está sendo criada. A criação de um índice particionado requer a permissão ALTER na tabela ou exibição onde o índice está sendo criado. A criação de uma tabela ou um índice particionado requer uma das permissões adicionais a seguir:

Permissão ALTER ANY DATASPACE. Essa permissão tem como padrão os membros da função de servidor fixa sysadmin e das funções de banco de dados fixas db\_owner e db\_ddladmin .

Permissão CONTROL ou ALTER no banco de dados no qual a função e o esquema de partição estão sendo criados.

Permissão CONTROL SERVER ou ALTER ANY DATABASE no servidor do banco de dados no qual a função e o esquema de partição estão sendo criados.

Usando o SQL Server Management Studio

Siga as etapas deste procedimento para criar um ou mais grupos de arquivos, arquivos correspondentes e uma tabela. Você fará referência a esses objetos no próximo procedimento quando criar a tabela particionada.

No Pesquisador de Objetos, clique com o botão direito do mouse no banco de dados no qual você deseja criar uma tabela particionada e selecione Propriedades.

Na caixa de diálogo Propriedades do Banco de dados – database\_name, em Selecionar uma página, selecione Grupos de Arquivos.

Continue adicionando linhas até que você tenha criado todos os grupos de arquivos para a tabela particionada.

Clique em OK.

Em Seleção de uma página, selecione Arquivos.

Em Linhas, clique em Adicionar. Na nova linha, digite um nome de arquivo e selecione um grupo de arquivos.

Continue adicionando linhas até que tenha criado ao menos um arquivo para cada grupo de arquivos.

Expanda a pasta Tabelas e crie uma tabela como você faz normalmente. Para obter mais informações, veja Criar tabelas (Mecanismo de Banco de Dados). Opcionalmente, você pode especificar uma tabela existente no próximo procedimento.

Para criar uma tabela particionada

Clique com o botão direito do mouse na tabela que você deseja particionar, aponte para Armazenamento e clique em Criar Partição....

No Assistente para Criar Partição, na página Bem-vindo ao Assistente para Criar Partição, clique em Avançar.

Na página Selecione uma Coluna de Particionamento, na grade Colunas de particionamento disponíveis, selecione a coluna em que deseja particionar sua tabela. Somente colunas com tipos de dados que possam ser usados para particionar dados serão exibidas na grade Colunas de particionamento disponíveis. Se você selecionar uma coluna computada como a coluna de particionamento, a coluna deverá ser designada como persistente.

As opções que você tem para as colunas de particionamento e o intervalo de valores são determinadas, basicamente, pela extensão na qual seus dados podem ser agrupados de forma lógica. Por exemplo, você pode optar por dividir seus dados em agrupamentos lógicos por meses ou trimestres de um ano. As consultas que você planeja fazer nos seus dados determinarão se o agrupamento lógico é adequado para o gerenciamento de suas partições de tabela. Todos os tipos de dados são válidos para uso como colunas de particionamento, exceto text, ntext, image, xml, timestamp, varchar(max), nvarchar(max), varbinary(max), tipos de dados de alias ou tipos de dados CLR definidos pelo usuário.

As opções adicionais a seguir estão disponíveis nessa página:

Colocar essa tabela na tabela particionada selecionada. Permite selecionar uma tabela particionada que contenha dados relacionados para unir a essa tabela na coluna de particionamento. Tabelas com partições unidas nas colunas de particionamento costumam ser consultadas de forma mais eficiente.

Alinhamento de armazenamento de índices não exclusivos e índices exclusivos com uma coluna de partição indexada

Alinha todos os índices da tabela que são particionados com o mesmo esquema de partição. Quando uma tabela e seus índices são alinhados, você pode mover partições para dentro e fora de tabelas particionadas com mais eficiência, pois seus dados são particionados com o mesmo algoritmo.

Depois de selecionar a coluna de particionamento e as outras opções, clique em Avançar.

Na página Selecione uma Função de Partição, em Selecione uma função de partição, clique em Nova função de partição ou em Função de partição existente. Se você escolher Nova função de partição, digite o nome da função.

Se você escolher Função de partição existente, selecione o nome da função que você gostaria de usar na lista. A opção Função de partição existente não estará disponível se não houver outra função de partição no banco de dados.

Na página Selecione um Esquema de Partição, em Selecione um esquema de partição, clique em Novo esquema de partição ou em Esquema de partição existente.

Se você escolher Novo esquema de partição, digite o nome do esquema. Se você escolher Esquema de partição existente, selecione o nome do esquema que você gostaria de usar na lista. A opção Esquema de partição existente não estará disponível se não houver outro esquema de partição no banco de dados.

Na página Mapear Partições, em Intervalo, selecione Limite esquerdo ou Limite direito para especificar se é para incluir o maior ou o menor valor de limite dentro de cada grupo de arquivos criado. Você deve sempre inserir um grupo de arquivos extra, além do número de grupos de arquivos especificado para os valores de limite durante a criação de partições.

Na grade Selecione os grupos de arquivos e especifique os valores de limite, em Grupo de arquivos, selecione um grupo de arquivos no qual deseja particionar seus dados. Em Limite, digite o valor de limite para cada grupo de arquivos. Se o valor de limite ficar em branco, a função de partição mapeará a tabela inteira ou o índice em uma única partição usando o nome da função de partição.

As opções adicionais a seguir estão disponíveis nessa página:

#### Definir Limites...

Abre a caixa de diálogo Definir Valores de Limite para selecionar os valores de limite e os intervalos de datas que deseja para suas partições. Essa opção estará disponível somente quando você tiver selecionado uma coluna de particionamento que contenha um dos seguintes tipos de dados: date, datetime, smalldatetime, datetime2 ou datetimeoffset.

#### Estimar armazenamento

Calcula o número de linhas, o espaço necessário e o espaço disponível para armazenamento para cada grupo de arquivos especificado para as partições. Esses valores são exibidos na grade como valores somente leitura.

A caixa de diálogo Definir Valores de Limite permite as opções adicionais a seguir:

#### Data de Início

Seleciona a data inicial para obter os valores de intervalo de suas partições.

#### Data de Término

Seleciona a data de término para obter os valores de intervalo de suas partições. Se você selecionar Limite esquerdo na página Mapear Partições, essa data será o último valor de cada grupo de arquivos/partição. Se você selecionar Limite direito na página Mapear Partições, essa data será o primeiro valor do próximo ao último grupo de arquivos.

#### Intervalo de Datas

Seleciona a granularidade de data ou incremento de valor de intervalo desejado para cada partição.

Depois de concluir essa página, clique em Avançar.

Na página Selecione uma Opção de Saída, especifique como você deseja preencher sua tabela particionada. Selecione Criar Script para criar um script SQL baseado nas páginas anteriores no assistente.

Selecione Executar imediatamente para criar a nova tabela particionada depois de concluir todas as páginas restantes no assistente. Selecione Agenda para criar uma nova tabela particionada em um momento predeterminado no futuro.

Se você selecionar Criar script, as opções a seguir estarão disponíveis em Opções de script:

#### Script para Arquivo

Gera o script como um arquivo. sql. Digite um nome de arquivo e o local na caixa Nome do arquivo ou clique em Procurar para abrir a caixa de diálogo Local do Arquivo de Script. Em Salvar Como, selecione Texto Unicode ou Texto ANSI.

#### Script para Área de Transferência

Salva o script na área de transferência.

Script para Nova Janela de Consulta

Gera o script para uma nova janela do Editor de Consultas. Essa é a seleção padrão.

Se você selecionar Agenda, clique em Alterar agenda.

Na caixa de diálogo Nova Agenda de Trabalho, na caixa Nome, digite o nome da agenda de trabalho.

Na lista Tipo de Agenda, selecione o tipo de agenda:

Iniciar automaticamente quando o SQL Server Agent for iniciado

Iniciar sempre que as CPUs estiverem ociosas

Recorrente. Selecione essa opção se sua nova tabela particionada for atualizada com novas informações regularmente.

Uma vez. Essa é a seleção padrão.

Marque ou desmarque a caixa de seleção Habilitado para habilitar ou desabilitar a agenda.

Se você selecionar Recorrente:

Em Frequência, na lista Ocorre, especifique a frequência de ocorrência:

Se você selecionar Diário, na caixa Ocorre periodicamente a cada, digite a frequência com que a agenda de trabalho se repete em dias.

Se você selecionar Semanal, na caixa Ocorre periodicamente a cada, digite a frequência com que a agenda de trabalho se repete em semanas. Selecione o dia ou os dias da semana em que a agenda de trabalho é executada.

Se você selecionar Mensalmente, selecione Dia ou O.

Se você selecionar Dia, digite o dia do mês que você deseja que a agenda de trabalho seja executada e a frequência com que a agenda de trabalho se repete em meses. Por exemplo, se você desejar que a agenda de trabalho seja executada no 15º dia do mês a cada dois meses, selecione Dia e digite "15" na primeira caixa e "2" na segunda caixa. Observe que o maior número permitido na segunda caixa é "99".

Se você selecionar O, selecione o dia específico da semana no mês que você deseja que a agenda de trabalho seja executada e a frequência com que a agenda de trabalho se repete em meses. Por exemplo, se você desejar que a agenda de trabalho seja executada no último dia da semana do mês a cada dois meses, selecione Dia, selecione último na primeira lista e dia da semana na segunda lista e depois digite "2" na última caixa. Você também pode selecionar primeiro, segundo, terceiro ou quarto, bem como dias específicos da semana (por exemplo: domingo ou quarta-feira) nas primeiras duas listas. Observe que o maior número permitido na última caixa é "99".

Em Frequência diária, especifique a frequência com que a agenda de trabalho se repete no dia da execução da agenda de trabalho:

Se você selecionar Ocorre uma vez às, digite a hora específica do dia em que a agenda de trabalho deve ser executada na caixa Ocorre uma vez às. Digite a hora, os minutos e os segundos do dia, bem como AM ou PM.

Se você selecionar Ocorre a cada, especifique a frequência com que a agenda de trabalho é executada durante o dia escolhido em Frequência. Por exemplo, se você desejar que o agendamento de trabalho se repita a cada 2 horas durante o dia em que é executado, selecione Ocorre a cada, digite "2" na primeira caixa e selecione hora(s) na lista. Nessa lista, você pode selecionar também minuto(s) e segundo(s). Observe que o maior número permitido na primeira caixa é "100".



Na caixa Iniciando às, digite a hora em que a agenda de trabalho deve começar a ser executada. Na caixa Terminando às, digite a hora em que a agenda de trabalho deve parar de se repetir. Digite a hora, os minutos e os segundos do dia, bem como AM ou PM.

Em Duração, em Data de início, digite a data que você deseja que a agenda de trabalho inicie a execução. Selecione Data de término ou Nenhuma data de término para indicar quando a execução da agenda de trabalho deve parar. Se você selecionar Data de término, digite a data em que você deseja que a execução da agenda de trabalho pare.

Se você selecionar Uma Vez, em Ocorrência única, na caixa Data, insira a data em que o agendamento de trabalho será executado. Na caixa Hora, digite a hora em que a agenda de trabalho será executada. Digite a hora, os minutos e os segundos do dia, bem como AM ou PM.

Em Resumo, em Descrição, verifique se todas as configurações da agenda de trabalho estão corretas.

Clique em OK.

Depois de concluir essa página, clique em Avançar.

Na página Resumo da Revisão, em Examinar as seleções, expanda todas as opções disponíveis para verificar se todas as configurações de partição estão corretas. Se tudo estiver como esperado, clique em Concluir.

Na página Progresso do Assistente para Criar Partição, monitore as informações de status das ações do Assistente para Criar Partição. Dependendo das opções selecionadas no assistente, a página de progresso pode conter uma ou várias ações. A caixa superior exibe o status geral do assistente e o número de mensagens de status, erro e aviso que ele recebeu.

As opções a seguir estão disponíveis na página Progresso do Assistente para Criar Partição :

#### **Detalhes**

Fornece a ação, status e qualquer mensagem retornada pela ação executada pelo assistente.

#### **Ação**

Especifica o tipo e o nome de cada ação.

#### **Status**

Indica se a ação do assistente retornou como um todo o valor de Êxito ou de Falha.

#### **Mensagem**

Fornece qualquer mensagem de aviso ou erro retornada pelo processo.

#### **Relatório**

Cria um relatório contendo os resultados do Assistente para Criar Partição. As opções são Exibir Relatório, Salvar Relatório no Arquivo, Copiar Relatório na Área de Transferência e Enviar Relatório como Email.

#### **Exibir Relatório**

Abre a caixa de diálogo Exibir Relatório, que contém um relatório de texto do progresso do Assistente para Criar Partições.

#### **Salvar Relatório no Arquivo**

Abre a caixa de diálogo Salvar Relatório Como .

#### **Copiar Relatório na Área de Transferência**

Copia os resultados do relatório de progresso do assistente na área de transferência.

### **Enviar Relatório como Email**

Copia os resultados do relatório de progresso do assistente em uma mensagem de email.

Quando terminar, clique em Fechar.

O Assistente para Criar Partição cria a função e o esquema de partição e aplica o particionamento à tabela especificada. Para verificar o particionamento da tabela, no Pesquisador de Objetos, clique com o botão direito do mouse na tabela e selecione Propriedades. Clique na página Armazenamento. A página exibe informações como o nome da função e do esquema de partição e o número de partições.

Usando Transact-SQL

Para criar uma tabela particionada

No Pesquisador de Objetos, conecte-se a uma instância do Mecanismo de Banco de Dados.

Na barra Padrão, clique em Nova Consulta.

Copie e cole o exemplo a seguir na janela de consulta e clique em Executar. O exemplo cria novos grupos de arquivos, uma função e um esquema de partição. Uma nova tabela é criada com o esquema de partição especificado no local de armazenamento

O uso de particionamento de tabelas se torna útil em grandes ambientes de banco de dados, onde existem a necessidade de grande acesso aos dados, não somente para esta causa, mas também para inúmeras funcionalidades dentro do banco, como: backup, rápido acesso a dados e dados históricos.

Para que seja necessário implementar particionamento de tabelas é aconselhável ter conhecimento sobre filegroups

### **O que é?**

O particionamento de tabelas (Oracle Partitioning) é a capacidade de dividir fisicamente tabelas de seu banco dados, sem a necessidade de reescrever o código fonte de sua aplicação, pois logicamente a tabela particionada é como uma tabela normal de seu banco de dados e é capaz de receber as mesmas instruções SQL (DDL, DML e DCL) que qualquer outra tabela, a separação dos dados acontece somente no âmbito físico, de acordo com o que for especificado no momento do particionamento pelo desenvolvedor.

### **Para que Serve?**

O particionamento é de grande utilidade quando estamos trabalhando com tabelas que tem uma massa de dados muito grande, pois a partir do momento que a tabela está particionada, os dados estão divididos fisicamente e na hora da busca das informações, um comando SELECT, por exemplo, você sempre irá buscar as informações em uma porção fisicamente menor.

Outro ponto está relacionado à segurança, se você perder partição por alguma falha (seja ela qual for) você pode fazer o restore e recover apenas da partição danificada sem afetar as demais diminuindo então o tempo de restauração e recuperação considerando também que os dados das outras partições ficam disponíveis para a aplicação durante todo esse processo, outra vantagem que se pode ter com relação à performance de seu banco de dados, é você poder colocar as partições em discos distintos colocando os dados mais acessados em discos de maior velocidade (para que seus usuários tenham uma boa performance) e os dados históricos em discos de menor velocidade visto que esse tipo de informação não é comumente acessada. Com o passar do tempo, as informações que vão se tornando “velhas” você pode mover dos discos de alta performance para os discos de menor performance sem haja a necessidade de parar a aplicação (isso faz parte das atividades do DBA).

### **Como Se Faz?**

Para particionar uma tabela, podemos fazer isso em dois momentos, primeiro no momento de criação da tabela, ou então em um segundo momento, é quando a tabela já está criada e você mesmo assim

ainda pode particioná-la utilizando uma package do Oracle chamada DBMS\_REDEFINITION (em breve escreverei um artigo sobre este tema), mas agora vamos ver como particionar uma tabela do “zero”. Neste artigo vamos considerar que teremos uma tabela que armazenará informações sobre notas fiscais (É uma tabela bem básica, apenas para entendermos o objetivo deste artigo) abaixo especificação da tabela.

#### Como Funciona

O particionamento descrito neste artigo é aquele que afeta individualmente tabelas e/ou índices, também conhecido como particionamento horizontal de tabelas. É um processo para segregar dados de tabelas específicas em várias unidades físicas menores.

Ele é diferente da segmentação de dados em datafiles diferentes, pois neste caso não temos controle de quais datafiles serão usados para armazenar dados específicos. O particionamento nos permite definir regras para que os dados sejam gravados em filegroups diferentes, com base nos valores de um ou mais campos da tabela.

Este recurso é bastante útil, mas não deve ser confundido com o sharding, comum em bancos que funcionam na nuvem, como o próprio AZURE SQL. Sharding envolve a criação de uma infraestrutura para que o banco todo funcione de forma “horizontal” e divida a carga de processamento entre máquinas diferentes.

#### Benefícios

Particionamento permite, por exemplo:

Melhorar performance de consultas que acessem dados de uma única partição;

Melhorar a performance da importação de grandes volumes de dados;

Reduzir tempo de execução de tarefas administrativas como reindexação, backup e restore, que são executadas na partição ao invés da tabela toda;

Facilitar operações de arquivamento de dados.

Uma vez definidas as regras de particionamento de cada tabela, a administração das partições é um processo simples. Portanto, é uma opção interessante a se considerar quando trabalhar com tabelas muito grandes (eu diria da ordem de 100Gb ou mais).

#### Criando Partições

Vejamos um exemplo prático. Temos uma tabela com dados históricos que vem apresentando problemas, particularmente em razão do seu tamanho.

A tabela contém histórico de transações de muitos anos, mas a grande maioria das pesquisas envolvendo esta tabela trata das transações do ano corrente. Mesmo assim, espera-se que esta tabela continue a ser usada pelo menos nos próximos 10 anos.

Tendo isso em mente, decidiu-se criar um particionamento da tabela com base na data da transação. Implementaremos estas ações do modo mais didático possível, desconsiderando o fato de que este processo eventualmente envolveria um tempo maior de parada da aplicação.

Em primeiro lugar, vamos renomear a tabela existente, visto que a tabela particionada que criaremos precisa ter o mesmo nome da tabela antiga, assim não será necessária nenhuma alteração na aplicação.

A normalização de um banco de dados é o tema do artigo de hoje. Este é um tema super importante para a criação do banco de dados, vamos colocarmos a mão-na-massa assim como fizemos no primeiro artigo de Modelagem de Dados e vamos continuar a passar os fundamentos da linguagem e os conceitos básicos de como funciona um banco de dados.

O Oracle é um SGBD relacional e isso quer dizer que ele aplica as regras definidas por Edgar Frank Codd, ele foi quem desenvolveu o modelo de banco de dados relacional. Ao todo são 12 regras, porém vou passar à vocês apenas as 3 primeiras que são as essenciais para o seu dia-a-dia.

### Normalização Banco de Dados

Para começarmos a normalização é necessário recapitularmos o modelo da primeira parte da aula de modelagem, abaixo vocês vão ver o modelo relacional de entidades que fizemos na semana passada vamos seguir a ideia de sempre darmos exemplos em cima da teoria

#### Criando Tabelas na Normalização de Dados

Toda entidade vira uma tabela;

Então seguindo esta regra teremos as seguintes tabelas:

TB\_PRODUTO

TB\_COMANDA

TB\_ESTOQUE

TB\_CLIENTE

TB\_CAIXA

Se você clicar no nome da entidade aparecerá o nome da tabela que vai ser criada.

Relacionamentos que possuem atributos viram tabelas (existe a possibilidade em relacionamentos 1:n dos atributos irem para uma das tabelas, ao invés de se criar uma nova. Entretanto, relacionamentos com atributos são mais comuns em relacionamentos n:n, gerando assim uma nova tabela);

Agora vamos criar as tabelas desses relacionamentos.

TB\_PRODUTO\_COMANDA

TB\_PRODUTO\_ESTOQUE

TB\_CAIXA\_ESTOQUE

TB\_CLIENTE\_PAGAMENTO

Toda tabela possui um ou mais campos que são os campos únicos, onde cada entidade se diferencia, por exemplo, um cliente possui um CPF único que pode ser o que diferencia todos os clientes, estes campos únicos são chamados de chaves primárias.

#### Criando as Chaves Primárias na Normalização de Dados

Abaixo seguem as chaves primárias de todas as tabelas criadas.

ID\_PRODUTO

ID\_COMANDA, DT\_INICIO, DT\_FIM

ID\_ESTOQUE

ID\_CLIENTE (Que nesse caso vai ser o CPF)

ID\_PAGAMENTO

ID\_PRODUTO, ID\_COMANDA, DT\_INICIO, DT\_FIM

ID\_PRODUTO, ID\_ESTOQUE

ID\_PAGAMENTO, ID\_ESTOQUE

ID\_CLIENTE, ID\_PAGAMENTO

Criando Chaves Estrangeiras na Normalização de Dados

Relacionamentos são representados por chaves estrangeiras (ou Foreign Key – atributo correspondente à chave primária de outra relação, base para a integridade referencial);

Temos as tabelas que possuem chaves estrangeiras que compõem os relacionamentos das tabelas do nosso banco de dados.

ID\_PRODUTO, ID\_COMANDA

ID\_PRODUTO, ID\_ESTOQUE

ID\_PAGAMENTO, ID\_ESTOQUE

ID\_CLIENTE, ID\_PAGAMENTO

Perceberam algo bem interessante, as chaves estrangeiras são os mesmos campos que formam as chaves primárias compostas dos relacionamentos, então tentem sempre achar esta conexão entre os relacionamentos, chaves estrangeiras e chaves primárias.

### **Criando Relacionamentos na Normalização**

Relacionamentos 1:1 podem ser mapeados numa única tabela (quando possuem a mesma chave primária), em duas tabelas (quando as chaves primárias são diferentes e um dos lados do relacionamento é obrigatório) ou em três tabelas (quando o relacionamento é opcional em ambos os lados)

No nosso caso existe apenas um relacionamento 1:1 na nossa modelagem que é o relacionamento da entidade Comanda x Cliente, porque um cliente pode apenas ter uma comanda para efetuar compras e uma comanda pode pertencer apenas a um cliente enquanto ela estiver ativada. Por esse motivo temos esse relacionamento 1:1.

Mas elas possuem chaves primárias diferentes então por esse motivo temos duas tabelas, porém com a obrigatoriedade do ID\_CLIENTE (Chave Primária da TB\_CLIENTE, na TB\_COMANDA).

Relacionamentos 1:n são mapeados de forma que a chave primária do lado “1” seja representada do lado “n” como chave estrangeira;

Relacionamentos n:n devem ser transformados em dois relacionamentos 1:n, resultando numa nova tabela;

Estes dois últimos passos ficarão mais legais nos desenhos que vou mostrar para vocês logo abaixo.

Ferramenta para Normalização de Dados

Após passar essas regras começamos a desenhar o nosso banco de dados, bom como nesse caso ficaria muito trabalhoso usar o Paint para desenhar tudo na mão, então eu vou utilizar o MySQL Workbench que é uma ferramenta utilizada para modelar o MySQL, que é um banco de dados muito utilizado para aplicações web e que foi comprado pela Oracle e melhor de tudo, esta ferramenta é totalmente grátis.

### **Formas Normais**

Um pouco mais de teoria com as 3 formas normais de Codd, que vou apresentar a vocês.

#### **1ª Forma Normal (1FN)**

Toda relação deve ter uma chave primária e deve-se garantir que todo atributo seja atômico. Atributos compostos devem ser separados.

Por exemplo, um atributo Endereço deve ser subdividido em seus componentes: Logradouro, Número, Complemento, Bairro, Cidade, Estado e CEP. Além disso, atributos multivalorados devem ser discriminados separadamente ou separados em uma outra relação.

Por exemplo, um atributo multivalorado Telefones poderia ser separado em Telefone Residencial, Telefone

Comercial e Telefone Celular ou, ainda, ser convertido em outra relação que pudesse representar um número indeterminado de telefones.

Já passei para vocês como transformar um Modelo de Relacionamentos de Entidades em um modelo de relacionamento de tabelas. E se alguém ficar com alguma dúvida por favor comentem para que eu possa ajudá-los a resolvê-las e claro fazer com que o meu curso evolua sanando as principais dúvidas de todos.

Para exemplificar vou pegar um dos casos de relacionamento, quando criamos a tabela TB\_PRODUTO\_COMANDA, que contém o Identificador da Comanda e o Identificador do Produto estamos transformando um relacionamento de n:n entre o produto e a comanda em um relacionamento 1:n, pois podem existir vários produtos, mas na tabela TB\_PRODUTO\_COMANDA vai existir apenas um ID\_PRODUTO e campo quantidade para fazer os cálculos na hora da compra e o mesmo serve para a Comanda, onde podem existir várias comandas, mas apenas um identificador de comanda poderá estar atrelado a tabela TB\_PRODUTO\_COMANDA.

### **2ª Forma Normal (2FN)**

Toda relação deve estar na 1FN e devem-se eliminar dependências funcionais parciais, ou seja, todo atributo não chave deve ser totalmente dependente da chave primária. Como exemplo, uma relação que contenha os atributos Código da Obra, Código do Fornecedor, Nome do Fornecedor e Preço de Venda, Considerando que a chave primária é composta pelos atributos Código da Obra e Código do Fornecedor, não está na Segunda Forma Normal, uma vez que o Nome do Fornecedor depende apenas do Código do Fornecedor, e não do Código da Obra. Uma nova relação (Fornecedor) deve ser criada contendo os campos Código do Fornecedor (como chave) e Nome do Fornecedor. Na relação original, ficariam os atributos Código da Obra e o Código do Fornecedor, ambos formando a chave primária composta, e o atributo Preço de Venda. Além disso, o atributo Código do Fornecedor também seria uma chave estrangeira para a nova relação criada. Esta forma normal ajuda a diminuir redundâncias de informações criadas indevidamente.

Essa regra é bem interessante, porque eliminamos informações duplicados e conseguimos conservar a integridade das informações, por exemplo, na tabela TB\_PRODUTO colocamos o nome do tipo do produto, bom vamos dizer que o produto seja uma bebida, e que um funcionário cadastrou o banco de dados Bebida, mas veio outro funcionário e ao cadastrar o produto cadastrou Bebidas, isso faria com que tivéssemos dois tipos de produto cadastrados no banco, que na verdade seriam os mesmos, por esse motivo é necessário criar essas tabelas.

### **3ª Forma Normal (3FN)**

Toda relação deve estar na 2FN e devem-se eliminar dependências funcionais transitivas, ou seja, todo atributo não chave deve ser mutuamente independente. Como exemplo, uma relação que contenha os atributos Matrícula do Funcionário (atributo chave), Nome do Funcionário, Código do Departamento e Nome do Departamento não está na Terceira Forma Normal.

O Nome do Departamento é dependente do Código do Departamento, e não da Matrícula do Funcionário. Uma mudança no nome do departamento, por exemplo, levaria a modificações em todos os funcionários daquele departamento.

Para eliminar este problema, cria-se uma nova relação (Departamento) contendo Código do Departamento e Nome do Departamento. Na relação original, retira-se o Nome de Departamento, mantendo-se o Código do Departamento, agora como chave estrangeira. Esta forma normal também ajuda a diminuir redundâncias e aumentar a independência das relações.



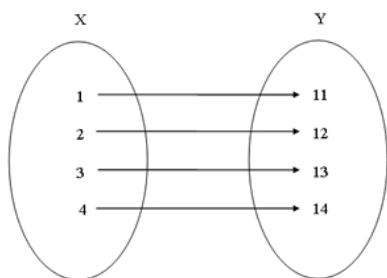
## Normalização e Dependência Funcional

### Dependência Funcional

O Modelo Relacional pegou emprestado da teoria de funções da matemática o conceito de dependência funcional.

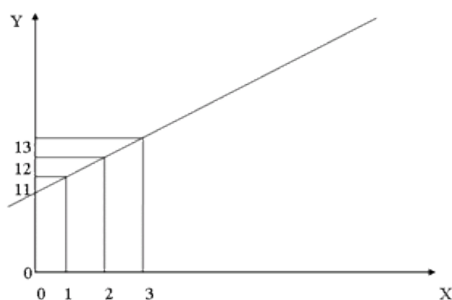
Iremos utilizar então a teoria de funções para explicar a dependência funcional do Modelo Relacional.

Considerando os seguintes conjuntos:

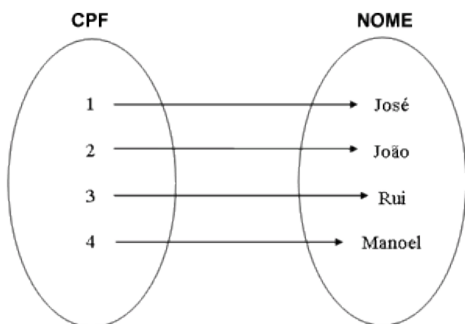


Observe que existe uma dependência entre os valores dos conjuntos, que pode ser expressa pela função  $f(x) = x + 10$ , ou seja,  $y$  é função de  $x$ , ou seja,  $y = f(x) = x + 10$ .

Esta dependência, esta função pode também ser expressa através do gráfico abaixo:



Agora, observe os conjuntos abaixo:



Observe que existe uma dependência entre os valores dos conjuntos, que pode ser expressa pela função  $f(\text{CPF})=\text{nome}$ .

Ou seja, nome é função do CPF, ou seja, se eu tiver um número de CPF, poderei encontrar o nome da pessoa correspondente.

É claro que não existe uma figura gráfica que possa descrever esta função, mas ela existe.

Esta dependência é expressa no Modelo Relacional da seguinte maneira:



**CPF -> NOME**

Leia-se a notação acima das seguintes maneiras:

com um número de CPF eu posso encontrar o nome da pessoa, ou ainda:

nome depende funcionalmente do CPF.

**Regras Para Encontrar Dependências Funcionais****1. Separação**

$A \rightarrow BC$  então  $A \rightarrow B$  e  $A \rightarrow C$

Exemplo:

CPF -> nome, endereço então CPF -> nome e CPF -> endereço

Leia o exemplo acima da seguinte maneira:

Se com um número de CPF eu encontro o nome e o endereço de uma pessoa, então com este mesmo número eu posso encontrar apenas o nome e com este mesmo número eu posso encontrar apenas o endereço.

**2. Acumulação**

$A \rightarrow B$  então  $AC \rightarrow B$

Exemplo:

CPF -> endereço então CPF, idade -> endereço

Leia o exemplo acima da seguinte maneira:

Se com um número de CPF eu encontro o endereço de uma pessoa, então com este mesmo número mais a idade da pessoa eu posso encontrar o endereço também.

**3. Transitividade**

$A \rightarrow B$  e  $B \rightarrow C$  então  $A \rightarrow C$

Exemplo:

CPF -> código-cidade e código-cidade -> nome-cidade então CPF -> nome-cidade

Leia o exemplo acima da seguinte maneira:

Se com um número de CPF eu encontro o código da cidade de uma pessoa, e com o código da cidade eu encontro o nome da cidade, então com o número do CPF eu posso encontrar o nome da cidade.

**4. Pseudo-Transitividade**

$A \rightarrow B$  e  $BC \rightarrow D$  então  $AC \rightarrow D$

Exemplo:

CPF -> código-funcionário e código-funcionário, mês -> salário-funcionário então CPF, mês -> salário-funcionário

Leia o exemplo acima da seguinte maneira:

Se com um número de CPF eu encontro o código do funcionário, e com o código do funcionário mais um certo mês eu encontro o salário que ele recebeu naquele mês, então com o número do CPF mais um certo mês eu posso encontrar o salário que ele recebeu naquele mês.

### Formas Normais

O conceito de normalização foi introduzido por E. F. Codd em 1972.

Inicialmente Codd criou as três primeiras formas de normalização chamando-as de: primeira forma normal (1NF), segunda forma normal (2NF) e terceira forma normal (3NF). Uma definição mais forte da 3NF foi proposta depois por Boyce-Codd, e é conhecida como forma normal de Boyce-Codd (FNBC).

Através do processo de normalização pode-se, gradativamente, substituir um conjunto de entidades e relacionamentos por um outro, o qual se apresenta "purificado" em relação às anomalias de atualização (inclusão, alteração e exclusão) as quais podem causar certos problemas, tais como:

- Grupos repetitivos (atributos multivalorados) de dados;
- Variação temporal de certos atributos, dependências funcionais totais ou parciais em relação a uma chave concatenada;
- Redundâncias de dados desnecessárias;
- Perdas acidentais de informação;
- Dificuldade na representação de fatos da realidade observada;
- Dependências transitivas entre atributos.

Normalização de relações é portanto uma técnica que permite depurar um projeto de banco de dados, através da identificação de inconsistências (informações em duplicidade, dependências funcionais mal resolvidas, etc).

À medida que um conjunto de relações passa para uma forma normal, vamos construindo um banco de dados mais confiável.

O objetivo da normalização não é eliminar todas as inconsistências, e sim controlá-las.

### Primeira Forma Normal

Uma relação está na primeira forma normal se todos os seus atributos são monovalorados e atômicos.

Quando encontrarmos um atributo multivalorado, deve-se criar um novo atributo que individualize a informação que esta multivalorada:

BOLETIM = {matricula-aluno, matéria, notas}

No caso acima, cada nota seria individualizada identificando a prova a qual aquela nota se refere:

BOLETIM = {matricula-aluno, matéria, número-prova, nota}

Quando encontrarmos um atributo não atômico, deve-se dividi-lo em outros atributos que sejam atômicos:

PESSOA = {CPF, nome-completo}

Vamos supor que, para a aplicação que utilizará esta relação, o atributo nome-completo não é atômico, a solução então será:

PESSOA = {CPF, nome, sobrenome}

### Segunda Forma Normal

Uma relação está na segunda forma normal quando duas condições são satisfeitas:

1. A relação estiver na primeira forma normal;
2. Todos os atributos primos dependerem funcionalmente de toda a chave primária.

Observe a relação abaixo:

BOLETIM = {matricula-aluno, codigo-materia, numero-prova, nota, data-da-prova, nome-aluno, endereço-aluno, nome-materia}

Fazendo a análise da dependência funcional de cada atributo primo, chegamos às seguintes dependências funcionais:

- matricula-aluno, codigo-materia, numero-prova -> nota
- codigo-materia, numero-prova -> data-da-prova
- matricula-aluno -> nome-aluno, endereço-aluno
- codigo-materia -> nome-materia

Concluimos então que apenas o atributo primo nota depende totalmente de toda chave primária. Para que toda a relação seja passada para a segunda forma normal, deve-se criar novas relações, agrupando os atributos de acordo com suas dependências funcionais:

BOLETIM = {matricula-aluno, codigo-materia, numero-prova, nota}

PROVA = {codigo-materia, numero-prova, data-da-prova}

ALUNO = {matricula-aluno, nome-aluno, endereço-aluno}

MATERIA = {codigo-materia, nome-materia}

O nome das novas relações deve ser escolhido de acordo com a chave.

### Terceira Forma Normal

Uma relação está na terceira forma normal quando duas condições forem satisfeitas:

1. A relação estiver na segunda forma normal;
2. Todos os atributos primos dependerem não transitivamente de toda a chave primária.

Observe a relação abaixo:

PEDIDO = {numero-pedido, codigo-cliente, data-pedido, nome-cliente, codigo-cidade-cliente, nome-cidade-cliente}

Fazendo a análise da dependência funcional de cada atributo primo, chegamos às seguintes dependências funcionais:

- numero-pedido -> codigo-cliente
- numero-pedido -> data-pedido
- codigo-cliente -> nome-cliente
- codigo-cliente -> codigo-cidade-cliente
- codigo-cidade-cliente -> nome-cidade-cliente

Concluimos então que apenas os atributos primos `codigo-cliente` e `data-pedido` dependem não transitivamente totalmente de toda chave primária.

Observe que:

- `numero-pedido -> codigo-cliente -> nome-cliente`
- `numero-pedido -> codigo-cliente -> codigo-cidade-cliente`
- `numero-pedido -> codigo-cliente -> codigo-cidade-cliente -> nome-cidade-cliente`

Isto é dependência transitiva, devemos resolver inicialmente as dependências mais simples, criando uma nova relação onde `codigo-cliente` é a chave, o `codigo-cliente` continuará na relação `PEDIDO` como atributo primo, porém, os atributos que dependem dele devem ser transferidos para a nova relação:

`PEDIDO = {numero-pedido, codigo-cliente, data-pedido}`

`CLIENTE = {codigo-cliente, nome-cliente, codigo-cidade-cliente, nome-cidade-cliente}`

As dependências transitivas da relação `PEDIDO` foram eliminadas, porém ainda devemos analisar a nova relação `CLIENTE`:

`codigo-cliente -> codigo-cidade-cliente -> nome-cidade-cliente`

Observe que o `nome-cidade-cliente` continua com uma dependência transitiva, vamos resolvê-la da mesma maneira :

`PEDIDO = {numero-pedido, codigo-cliente, data-pedido}`

`CLIENTE = {codigo-cliente, nome-cliente, codigo-cidade-cliente}`

`CIDADE = {codigo-cidade-cliente, nome-cidade-cliente}`

O nome das novas relações deve ser escolhido de acordo com a chave.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## Linguagem SQL

A linguagem SQL surgiu em meados da década de 70, sendo resultado de um estudo de E. F. Codd, membro do laboratório de pesquisa da IBM em San Jose, Califórnia. Este estudo tinha foco em desenvolver uma linguagem que adapta-se ao modelo relacional. O primeiro sistema de BD baseado em SQL tornou-se comercial no final dos anos 70 juntamente com outros sistema de BD's relacionais.

O sucesso da linguagem SQL foi tão grande que obrigou o ANSI (American National Standarts Institute), a padronizar as implementações da linguagem, assim, nos dias de hoje, a maior parte de BD's seguem criteriosamente está padronização, podendo ter algumas variações, mais mesmo assim não afetando na padronização global da linguagem tornando assim a portabilidade mais fácil, se seguida de forma adequada pelo DBA.

Em 1982, foi lançada a primeira versão padronizada da linguagem SQL, que vieram ganhando melhorias de acordo com sua evolução e tornando-se assim, a mais poderosa ferramenta para definição e manipulação de BD's e hoje utilizada em grande parte dos BD existente, tais como MySQL, SQLServer, Firebird dentre outros.

### SQL: Linguagem De Múltiplo Uso

A linguagem SQL tem papel muito importante nos SGDBs atuais, podendo ter diversos enfoques:

Linguagem de Manipulação de Dados ou DML (Data Manipulation Language): A DML é um subconjunto da linguagem SQL, utilizada para Selecionar(SELECT), Inserir(INSERT), Atualizar(UPDATE) e Apagar(DELETE).

Linguagem de Definição de Dados ou DDL (Data Definition Language): A DDL permite ao usuário permite a manipulação de tabelas e elementos associados, tipo chave primária e chaves estrangeira, índices, etc. Os principais comandos são CREATE, DROP, ALTER(em algumas situações).

Linguagem de Controle de Dados ou DCL (Data Control Language): A DCL controla os aspectos destinados a autorização de dados e licenças de usuários para manipulação de dados dentro do BD. Alguns comandos comuns são GRANT(dá privilégios para usuários), REVOKE (revoga privilégios de usuários), COMMIT(em resumo grava dados no BD) e ROLLBACK(descarta dados existentes desde o último COMMIT).

Linguagem de Consultas de Dados ou DQL (Data Query Language): embora na DQL exista somente um comando(SELECT) é o mais utilizado, principalmente para consultas parametrizadas. Lembre que o SELECT também é considerado um comanda DML. São exemplos os apresentados na Listagem 1.

Listagem 1. Exemplos de DQL

```
1SELECT nome FROM clientes; SELECT id,nome FROM pessoas WHERE id = :id;
```

Complementando a DQL, temos também uma série de Cláusulas (FROM, WHERE, GROUP BY, HAVING, ORDER BY, DISTINCT), Operadores Lógicos (AND, OR, NOT), Operadores de Comparação (<, >, <=>, <=, =, >=, BETWEEN, LIKE) e Funções de Soma (AVG, COUNT, SUM, MIN, MAX).

## SQL

SQL (Structured Query Language) é a linguagem padrão universal para manipular bancos de dados relacionais através dos SGBDs. Isso significa que todos os SGBDRs (Sistema de Gerenciamento de Banco de Dados Relacionais) oferecem uma interface para acessar o banco de dados utilizando a linguagem SQL, embora com algumas variações. Logo, saber o que é SQL e como utilizá-la é fundamental para qualquer desenvolvedor de softwares.

A "Linguagem Estruturada de Consultas" (SQL, traduzida para o português) é utilizada para interagir com o SGBD e executar várias tarefas como inserir e alterar registros, criar objetos no banco de dados, gerenciar usuário, consultar informações, controlar transações, etc. Todas as operações realizadas no banco de dados podem ser solicitadas ao SGBD utilizando esta linguagem.

A linguagem SQL é dividida em 4 agrupamentos de acordo com o tipo de operação a ser executada no banco de dados. A saber, DML (Data Manipulation Language, ou Linguagem de Manipulação de Dados em português), DDL (Data Definition Language, ou Linguagem de Definição de Dados em português), DCL (Data Control Language, ou Linguagem de Controle de Dados em português) e DTL (Data Transaction Language, ou Linguagem de Transação de Dados em português).

Alguns autores classificam também uma divisão da linguagem para consultas, a DQL (Data Query Language, Linguagem de Consulta de Dados), que tem apenas um comando (SELECT), porém é mais comum encontrar este comando como integrante da DML, juntamente com os comandos INSERT, UPDATE e DELETE. Vejamos os comandos SQL de cada agrupamento.

### **Dml - Data Manipulation Language**

DML (Linguagem de Manipulação de Dados) é o subconjunto mais utilizado da linguagem SQL, pois é através da DML que operamos sobre os dados dos bancos de dados com instruções de inserção, atualização, exclusão e consulta de informações. Os comandos SQL desse subconjunto são:

INSERT: utilizado para inserir registros (tuplas), em uma tabela.

Exemplo: INSERT into CLIENTE(ID, NOME) values(1,'José');

UPDATE: utilizado para alterar valores de uma ou mais linhas (tuplas) de uma tabela.

Exemplo: UPDATE CLIENTE set NOME = 'João' WHERE ID = 1;

DELETE: utilizado para excluir um ou mais registros (tupla) de uma tabela.

Exemplo: DELETE FROM CLIENTE WHERE ID = 1;

SELECT: O principal comando da SQL, o comando select é utilizado para efetuar consultas no banco de dados.

Exemplo: SELECT ID, NOME FROM CLIENTE;

Nota: Registro, Linha e Tupla são palavras sinônimas para referenciar a uma linha da tabela.

### **DDL - DATA DEFINITION LANGUAGE**

DDL (Linguagem de Definição de Dados) é o subconjunto da SQL utilizado para gerenciar a estrutura do banco de dados. Com a DDL podemos criar, alterar e remover objetos (tabelas, visões, funções, etc.) no banco de dados. Os comandos deste subconjunto são:

CREATE: utilizado para criar objetos no banco de dados.

Exemplo (criar uma tabela): CREATE TABLE CLIENTE ( ID INT PRIMARY KEY, NOME VARCHAR(50));

ALTER: utilizado para alterar a estrutura de um objeto.

Exemplo (adicionar uma coluna em uma tabela existente): ALTER TABLE CLIENTE ADD SEXO CHAR(1);

DROP: utilizado para remover um objeto do banco de dados.

Exemplo (remover uma tabela): DROP TABLE CLIENTE;

### **DCL - Data Control Language**

DCL (Linguagem de Controle de Dados) é o subconjunto da SQL utilizado para controlar o acesso aos dados, basicamente com dois comandos que permite ou bloqueia o acesso de usuários a dados. Vejamos estes comandos:

GRANT: Autoriza um usuário a executar alguma operação.

Exemplo (dar permissão de consulta na tabela cliente para o usuário carlos): GRANT select ON cliente TO carlos;

REVOKE: Restringe ou remove a permissão de um usuário executar alguma operação.

Exemplo (não permitir que o usuário carlos crie tabelas no banco de dados): REVOKE CREATE TABLE FROM carlos;

### **DTL - Data Transaction Language**

DTL (Linguagem de controle de transações) é o subconjunto da SQL que fornece mecanismos para controlar transações no banco de dados. São 3 comandos: iniciar uma transação (BEGIN TRANSACTION), efetivar as alterações no banco de dados (COMMIT) e cancelar as alterações (ROLLBACK).

Quem quer trabalhar com desenvolvimento de softwares precisa aprender a SQL, pois a maioria dos sistemas de informação interage com banco de dados, e essa é a linguagem universal para fazer qualquer coisa nos bancos de dados relacionais (o tipo de banco de dados mais utilizado na indústria). Pode haver pequenas variações na linguagem dependendo do SGBD, mas a sintaxe dos comandos são muito parecidas.

Cada comando citado neste artigo possui uma série de recursos, o comando que tem mais recursos, obviamente, é o comando SELECT. O objetivo deste artigo é apenas apresentar a linguagem SQL e seus comandos, continue ligado aqui no { Dicas de Programação } que vamos ver os detalhes de cada comando desta linguagem.

### **ANSI**

A ANSI (American National Standards Institute) promove o desenvolvimento de Normas Nacionais Americanas (ANS) ao dar acreditação os procedimentos de organizações desenvolvedoras de normas (SDOs). Esses grupos cooperam para desenvolver normas voluntárias de consenso nacional. A acreditação pela ANSI significa que os procedimentos usados pelo órgão de normas em ligação ao desenvolvimento de Normas Nacionais Americanas cumprem os requisitos essenciais do instituto de abertura, equilíbrio, consenso e processo justo.

O processo de ANS é feito para resistirem análises enquanto protege os direitos e interesses de cada participante. Essencialmente, normas nacionais da ANSI aceleram a aceitação de produtos no mercado enquanto deixam claro como melhorar a segurança desses produtos para a proteção dos consumidores.

Os passos do processo de Normas Nacionais Americanas incluem:

Consenso sobre uma norma proposta por um grupo ou “órgão de consenso”, que inclui representantes de partes interessadas e substancialmente afetadas

Revisão pública em base ampla e comentário sobre minutas de normas

Consideração de e resposta a comentários enviados por associados com poder de voto do órgão de consenso relevante e por comentaristas de revisão pública

Incorporação de mudanças aprovadas em uma minuta de norma

Direito de apelação para qualquer participante que ache que os princípios de processo justo não foram suficientemente respeitados durante o desenvolvimento de normas de desenvolvimento conforme os procedimentos acreditados pela ANSI do desenvolvedor de normas

Como mencionado acima, para manter a acreditação da ANSI, os desenvolvedores de normas precisam seguir consistentemente uma série de requisitos ou procedimentos que governam o processo de desenvolvimento de consenso. Esses requisitos são definidos em um documento conhecido como “Requisitos Essenciais da ANSI”; uma série de documentos de orientação ajuda a explicar melhor esses procedimentos.

Um processo justo é a chave para assegurar que ANSs sejam desenvolvidas em um ambiente equitativo, acessível e responsivo às exigências de vários colaboradores. O processo aberto e justo da ANSI





## **Segurança em Banco de dados: Integridade de dados**

A principal característica de um sistema é controlar os processos de uma empresa. Dessa forma, cada solução que encontramos hoje no mercado de tecnologia possui características com objetivo de proporcionar aos clientes uma qualidade considerável em requisitos de segurança, performance, escalabilidade e, acima de tudo, coerência no uso da informação.

Esta coerência se trata de garantir que uma informação será verdadeira, será confiável e íntegra.

Quando um sistema controla os dados de uma organização, estes dados devem ser cuidadosamente analisados, afinal eles estarão de alguma forma interligados entre si no que diz respeito ao processo do negócio como um todo.

Por exemplo, um cadastro de fornecedores estará de alguma forma se comunicando com o cadastro de produtos, afinal os produtos pertencem a um fornecedor. Assim, a integridade de uma entidade pode ter impacto diretamente em outra entidade.

Normalmente cada funcionário tem um cargo e uma função; operar um sistema faz parte do dia a dia deste funcionário. Quando existe mais de um funcionário com a mesma função, eles utilizam o mesmo sistema e realizam os mesmos processos.

Isso pode parecer simples, porém se um sistema não tiver regras de integridade de dados, a forma de inserir os dados neste sistema ocorrerá de forma desordenada, causando um grande problema de registros sem uma regra definida ou mesmo inseridos de forma incorreta.

Podemos imaginar agora um ambiente de uma empresa que usa um sistema para controlar seus processos e, em um determinado momento, optam por criar um BI. Quando criamos um BI, partimos do princípio de que existe uma integridade na informação que será inserida neste BI.

Toda informação do sistema que controla a empresa será enviado por algum processo de ETL (ver **BOX 1**) para esse BI. Um sistema sem integridade de dados pode resultar em um BI não confiável e com uma informação falsa.

### **BOX 1. ETL**

O Extract Transform Load dá nome às ferramentas que têm como função a extração de dados em diversos sistemas, fazem tratamentos e transformações nestes dados, para então inseri-los em DataWarehouses e Data Marts.

Geralmente este BI é consultado por diretores das empresas para apoiarem na tomada de decisões em várias esferas do negócio, e pode funcionar como um termômetro para medir se, por exemplo, um produto é viável ou não no mercado.

Podemos perceber que a falta de integridade na informação pode trazer danos irreversíveis para o negócio, podendo causar problemas de impactos financeiros.

Analistas, desenvolvedores, ADs e DBAs, consomem uma boa parte de seu tempo planejando como usar as regras de integridade de dados nos SGBDs de forma eficiente. Visando esta necessidade, demonstraremos neste artigo os mecanismos existentes para garantir a integridade da informação, e demonstraremos em exemplos práticos como implementar as regras no SQL Server e no Oracle.

## **Segurança em Banco de Dados: conheça as 5 principais causas de ataques**

Atualmente, o maior bem que uma empresa possui é a informação. E, quando se trata de manter a confiabilidade dos próprios dados, são muitos os desafios enfrentados — sendo a segurança em banco de dados um dos maiores deles.

Por várias perspectivas, existem ameaças à integridade da informação que transita dentro de uma organização. Cabe, então, aos especialistas contratados moderar a ocorrência desse cenário e agir de forma concisa contra investidas não autorizadas.

Nesse sentido, é muito importante ter conhecimento sobre as principais brechas possíveis que podem existir nos bancos de dados empresariais. Por isso, continue lendo e veja as principais causas de ataques para que você se previna e não permita que isso aconteça na sua empresa!

## **5 principais causas de ataques a banco de dados**

### **1. SQL Injection**

O tipo mais conhecido de ataque a banco de dados é o SQL Injection. Nessa possibilidade, são incluídas instruções não autorizadas e mal-intencionadas no sistema, que podem dar os mais diversos privilégios ao agente invasor.

Um hacker que consegue acesso usando SQL Injection pode dar a si mesmo permissão total de manipulação das informações armazenadas, e causar um enorme estrago nos dados mal protegidos da empresa.

Existe ainda uma segunda categoria de ataques de injeção: o NoSQL Injection, que age em cima de soluções Big Data.

Assim, para que essas aberturas a acessos indevidos não aconteçam, os bancos devem estar muito bem codificados, com a segurança sempre em dia. A natureza dessas soluções já não utiliza nenhum comando SQL, impossibilitando o primeiro tipo de ataque.

### **2. Privilégios demais a pessoas demais**

Controlar o que se pode acessar e quais ações podem ser realizadas em cima das informações é uma parte básica da segurança em banco de dados. Contudo, existem empresas que ainda não se preocupam com esse risco.

Se muitas pessoas têm acessos a dados sensíveis ou ao ambiente de produção, por exemplo, é muito possível — mesmo que de forma não intencional — que um colaborador delete informações que vão parar o funcionamento de vários sistemas.

E isso pode acarretar em um prejuízo financeiro de forma imediata a empresa. Além do problema do dinheiro, o tempo que será gasto pelo DBA para “apagar esse incêndio” será outro prejuízo em grande escala.

Afinal, uma situação dessas atrasa todos os outros projetos evolutivos em andamento, impedindo a empresa de crescer por uma total falta de cuidado com seu ativo mais precioso.

### **3. Deficiência na auditoria**

Quando novas tabelas, ou mesmo campos são criados em um banco de dados, eles devem passar por um processo extremamente rigoroso de auditoria. Não se pode deixar passar nada, pois um simples erro, como um caractere a menos em um campo, já pode parar o sistema de forma geral.

E como ninguém deveria poder alterar um banco em horários não planejados, isso pode fazer com que durante o restante do dia os funcionários não possam trabalhar mais nos sistemas que acessam o banco de dados.

Caso a empresa não possua recursos ou conhecimento suficiente para garantir a integridade do banco de dados, é interessante terceirizar esse serviço, deixando que profissionais especializados cuidem da saúde do sistema e impedindo que cenários de bloqueio do trabalho de outras áreas **ocorra**.

### **4. Sistemas de segurança fracos e/ou desatualizados**

Um erro muito comum que os usuários cometem nas empresas é ter senhas fáceis, ou até manter as senhas padrão em seu acesso. E, a partir de uma falha como essas, todos podem sair perdendo.

A importância de fortificar a senha deve ser muito bem esclarecida a todos, mesmo aos que não possuem conhecimentos de segurança.

Nesse sentido, os firewalls e as políticas de bloqueio e exceção devem ser sempre atualizadas, e técnicos com alta capacidade devem ser mantidos para cuidar dessa atividade. Até porque nenhum sistema é completamente livre de invasões.

Tecnologias que já estão ultrapassadas, com certeza, já foram destrinchadas por hackers e são mais vulneráveis a quem possui esse tipo de conhecimento.

Justamente por isso, são lançadas, periodicamente, novas versões de programas de proteção, que buscam estar sempre à frente de ataques — e esse ciclo continuará. Logo, para manter seguros seus dados, é preciso usar programas confiáveis, pessoas competentes e manter as atualizações em dia.

## **5. Exposição de mídia storage**

Muitos casos de ataques também ocorrem a backups mal protegidos, e não à base principal de uma corporação. Por isso, quando são feitos backups do banco de dados, é extremamente importante lembrar que a segurança da mídia storage deve ser, pelo menos, igual à do servidor original.

Inclusive, o cuidado com essa tarefa deve ser o mesmo, pois faz parte de atividades que sempre serão realizadas. Assim, uma boa solução é realizar os backups na nuvem, pois a segurança nesse ambiente está sempre à frente em termos de novas tecnologias.

Caso sejam escolhidos backups locais, o processo a seguir deve ser definido e auditado, pois uma cópia da base mal protegida já é uma porta de entrada para quem quer prejudicar sua empresa.

### **Como cuidar da segurança em banco de dados da sua empresa**

É trabalho dos responsáveis pela empresa zelar pela segurança das informações que pertencem a ela. E, para realizar isso com sucesso, é essencial ter um time de alta competência — seja ele próprio ou terceirizado.

Ter conhecimento sobre as tecnologias que garantem maior segurança e saber que isso tem um custo a arcar são atitudes que dão aos gerentes e líderes mais confiança para garantir a integridade dos dados.

Além disso, é sempre importante lembrar, os gastos com a proteção do banco de dados, com certeza, serão menores que os prejuízos de uma invasão, um roubo ou um vazamento de informações cruciais para o negócio.

O armazenamento acaba se tornando parte fundamental do corpo de uma empresa, pois registrar todo o histórico do negócio é fundamental para seus planos futuros. Por isso, investir na segurança em banco de dados é parte essencial do crescimento empresarial saudável e da busca por se destacar no mercado!

E aí, gostou do post? Essas dicas sobre como proteger seu banco de dados foram úteis? Então, aproveite agora para assinar a nossa newsletter e continue por dentro de muitos outros assuntos do mundo da tecnologia!

### **Conceitos sobre Segurança em Banco de Dados**

Os bancos de dados são utilizados para armazenar diversos tipos de informações, desde dados sobre uma conta de e-mail até dados importantes da Receita Federal. A segurança do banco de dados herda as mesmas dificuldades que a segurança da informação enfrenta, que é garantir a integridade, a disponibilidade e a confidencialidade. Um Sistema gerenciador de banco de dados deve fornecer mecanismos que auxiliem nesta tarefa.

Os bancos de dados SQL implementam mecanismos que restringem ou permitem acessos aos dados de acordo com papéis ou roles fornecidos pelo administrador. O comando GRANT concede privilégios específicos para um objeto (tabela, visão, seqüência, banco de dados, função, linguagem procedural, esquema ou espaço de tabelas) para um ou mais usuários ou grupos de usuários.

A preocupação com a criação e manutenção de ambientes seguros se tornou a ocupação principal de administradores de redes, de sistemas operacionais e de bancos de dados. Pesquisas mostram que a maioria dos ataques, roubos de informações e acessos não- autorizados são feitos por pessoas que pertencentes à organização alvo.

Por esse motivo, esses profissionais se esforçam tanto para criar e usar artifícios com a finalidade de eliminar os acessos não-autorizados ou diminuir as chances de sucesso das tentativas de invasão (internas ou externas). Os controles de acesso em sistemas de informação devem certificar que todos os acessos diretos ao sistema ocorram exclusivamente de acordo com as modalidades e as regras pré-estabelecidas, e observadas por políticas de proteção.

De modo geral, os mecanismos de segurança referem-se às regras impostas pelo subsistema de segurança do SGBD, que verifica todas as solicitações de acesso, comparando-as com as restrições de segurança armazenadas no catálogo do sistema. Entretanto existem brechas no sistema e ameaças externas que podem resultar em um servidor de banco de dados comprometido ou na possibilidade de destruição ou no roubo de dados confidenciais.

As ameaças aos bancos de dados podem resultar na perda ou degradação de alguns ou de todos os objetivos de segurança aceitos, são eles: integridade, disponibilidade, confidencialidade. A integridade do banco de dados se refere ao requisito de que a informação seja protegida contra modificação imprópria.

A disponibilidade do banco de dados refere-se a tornar os objetos disponíveis a um usuário ou a um programa ao qual eles têm um direito legítimo. A confidencialidade do banco de dados se refere à proteção dos dados contra a exposição não autorizada. O impacto da exposição não autorizada de informações confidenciais pode resultar em perda de confiança pública, constrangimento ou ação legal contra a organização.

### **Controle de Acesso**

É todo controle feito quanto ao acesso ao BD, impondo regras de restrição, através das contas dos usuários. O Administrador do BD (DBA) é o responsável superior por declarar as regras dentro do SGBD. Ele é o responsável por conceder ou remover privilégios, criar ou excluir usuários, e atribuição de um nível de segurança aos usuários do sistema, de acordo com a política da empresa.

### **Controle de Inferência**

É um mecanismo de segurança para banco de dados estatísticos que atua protegendo informações estatísticas de um indivíduo ou de um grupo. Bancos de dados estatísticos são usados principalmente para produzir estatísticas sobre várias populações.

O banco de dados pode conter informações confidenciais sobre indivíduos. Os usuários têm permissão apenas para recuperar informações estatísticas sobre populações e não para recuperar dados individuais, como, por exemplo, a renda de uma pessoa específica.

### **Controle de Fluxo**

É um mecanismo que previne que as informações fluam por canais secretos e violem a política de segurança ao alcançarem usuários não autorizados. Ele regula a distribuição ou fluxo de informação entre objetos acessíveis. Um fluxo entre o objeto A e o objeto B ocorre quando um programa lê valores de A e escreve valores em B. Os controles de fluxo têm a finalidade de verificar se informações contidas em alguns objetos não fluem explicita ou implicitamente para objetos de menor proteção. Dessa maneira, um usuário não pode obter indiretamente em B aquilo que ele ou ela não puder obter diretamente de A.

### **Criptografia de Dados**

Você pode ler aqui um pouco mais sobre criptografia. É uma medida de controle final, utilizada para proteger dados sigilosos que são transmitidos por meio de algum tipo de rede de comunicação. Ela também pode ser usada para oferecer proteção adicional para que partes confidenciais de um banco de dados não sejam acessadas por usuários não autorizados. Para isso, os dados são codificados através da utilização de algum algoritmo de codificação. Assim, um usuário não autorizado terá dificuldade para decifrá-los, mas os usuários autorizados receberão chaves para decifrar esses dados. A criptografia permite o disfarce da mensagem para que, mesmo com o desvio da transmissão, a mensagem não seja revelada.

### **Usuários**

Abrange usuários e esquema do banco de dados onde cada banco de dados Oracle tem uma lista de nomes de usuários. Para acessar um banco de dados, um usuário deve usar um aplicativo desse tipo e

tentar uma conexão com um nome de usuário válido. Cada nome tem uma senha associada para evitar o uso sem autorização.

Devem ser implementados ainda diferentes perfis de usuário para diferentes tarefas no Oracle, tendo em vista que cada aplicação/usuário tem a sua necessidade de acesso. Existe ainda a possibilidade de proteger os perfis com senha, o que é uma excelente medida. Além dessas medidas, o uso de cotas aumenta a restrição de espaço em disco a ser utilizado por usuários/aplicativos.

### **Domínio de Segurança**

Onde cada usuário tem um domínio de segurança, um conjunto de propriedades que determinam coisas como ações (privilegios e papeis) disponíveis para o usuário; cota de tablespaces (espaço disponível em disco) do usuário; limites de recursos de sistema do usuário.

As tabelas (tablespaces) do sistema, como a system, devem ser protegidas de acessos de usuários diferentes dos usuários de sistema. A liberação de escrita e alteração de dados em tais tabelas é muito comum em ambientes de teste, onde os programadores e DBAs tomam tal atitude para evitar erros de aplicação por falta de privilégios. Porém, em ambientes de produção, tal medida é totalmente desaconselhável.

### **Autoridade**

As autoridades fornecem um método de agrupar privilégios e controlar o nível de acesso dos administradores e operadores da base de dados com relação à manutenção e operações permitidas. As especificações da base de dados estão armazenadas em catálogos da própria base de dados. As autoridades do sistema estão associadas a membros de grupos e armazenadas no arquivo de configuração administrativa do banco de dados. Este arquivo define as concessões de acesso e o que poderá ser executado de acordo com cada grupo.

### **Privilégios**

Os privilégios são permissões únicas dadas a cada usuário ou grupo. Eles definem permissões para tipos de autorização. Pelos privilégios é possível autorizar o usuário a modificar ou alcançar determinado recurso do Banco de Dados.

Os privilégios também são armazenados em catálogos do próprio Banco de Dados, visto que os grupos de autoridade por já possuírem grupos predefinidos de privilégio concedem implicitamente privilégios a seus membros.

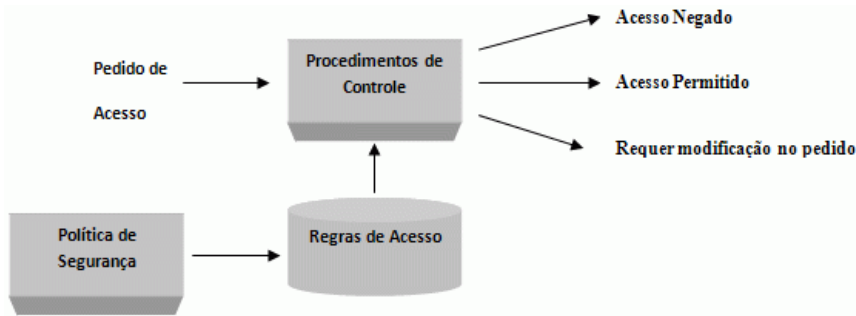
### **Tipos de privilégios discricionários**

O SGBD deve oferecer acesso seletivo a cada relação do banco de dados baseando-se em contas específicas. As operações também podem ser controladas; assim, possuir uma conta não necessariamente habilita o possuidor a todas as funcionalidades oferecidas pelo SGBD. Informalmente existem dois níveis para a atribuição de privilégios para o uso do sistema de banco de dados:

- O nível de conta: Nesse nível, o DBA estabelece os privilégios específicos que cada conta tem, independente das relações no banco de dados.
- O nível de relação (ou tabela): Nesse nível, o DBA pode controlar o privilégio para acessar cada relação ou visão individual no banco de dados.

### **Revogação de Privilégios**

Em alguns casos, interessa conceder um privilégio temporário a um usuário. Por exemplo, o proprietário de uma relação pode querer conceder o privilégio SELECT a um usuário para uma tarefa específica e depois revogar aquele privilégio quando a tarefa estiver completada. Por isso, é necessário um mecanismo para a revogação de privilégios. Em SQL, um comando REVOKE é introduzido com o intento de cancelar privilégios.



### Sistema de Controle de Acesso

Controle de acesso obrigatório e para segurança multi-nível

Neste método, o usuário não tem um meio termo, ou ele tem ou não tem privilégios, sendo utilizado normalmente em BD que classificam dados de usuários, onde é necessário um nível a mais de segurança. A maioria dos SGBDs não oferecem esse tipo de controle de acesso obrigatório, ficando com os controles discricionários ditos anteriormente. Normalmente são utilizados em sistemas governamentais, militares ou de inteligência, assim como industriais e corporativas.

As **classes de segurança** típicas são altamente sigilosas (top secret, TS), secreta (secret, S), confidenciais (confidential) (C) e não Classificada (unclassified, U), em que TS é o nível mais alto e U é o mais baixo.

De uma forma geral, os mecanismos de controle de acesso obrigatório impõem segurança multinível, pois exigem a classificação de usuários e de valores de dados em classes de segurança e impõem as regras que proíbem o fluxo de informação a partir dos níveis de segurança mais altos para os mais baixos.

### Controle de acesso baseado em papéis

É uma abordagem para restringir o acesso a usuários autorizados e uma alternativa aos sistemas de controles de acesso do tipo MAC e DAC. O conceito de controle de acesso baseado em papéis surgiu com os primeiros sistemas computacionais multiusuários interativos. A ideia central do RBAC é que permissões de acesso são associadas a papéis, e estes papéis são associados a usuários. Papéis são criados de acordo com os diferentes cargos em uma organização, e os usuários são associados a papéis de acordo com as suas responsabilidades e qualificações. Vários indivíduos podem ser designados para cada papel. Os privilégios de segurança comuns a um papel são concedidos ao nome dele, e qualquer indivíduo designado para esse papel automaticamente teria esses privilégios concedidos.

Os usuários podem ser facilmente remanejados de um papel para outro. Mudanças no ambiente computacional, como instalação de novos sistemas e remoção de aplicações antigas, modificam apenas o conjunto de permissões atribuídas aos diferentes papéis, sem envolver diretamente o conjunto de usuários.

A separação de tarefas é um requisito importante em diversos SGBDs. É necessária para impedir que um usuário realize sozinho o trabalho que requer o envolvimento de outras pessoas. A exclusão mútua de papéis é um método que pode ser implementado com sucesso.

Outro aspecto relevante nos sistemas RBAC são as restrições temporais possíveis que podem existir nos papéis, como o tempo e a duração das ativações de papéis e o disparo temporizado de um papel por uma ativação de outro papel. O uso de um modelo RBAC é um objetivo altamente desejado para solucionar os principais requisitos de segurança das aplicações baseadas na web.

### Controle de acesso utilizando Triggers

Com a utilização das Triggers é possível criar mecanismos de segurança mais complexos que podem ser disparados cada vez que um evento é chamado. O comando *Insert* na tabela é exemplo de um evento que pode ser usado para disparar uma *Triggers*, além disso, as mesmas podem ser disparadas antes ou depois de comando especificado com o objetivo de prover maior rigor no controle de segurança.

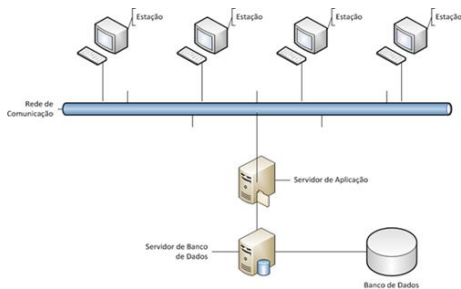
Se o comando executado pelo usuário não for validado pela *Triggers*, um erro é sinalizado do corpo da própria *Triggers* para impedir que a tabela seja modificada indevidamente.



**Banco de Dados Distribuído**

A necessidade de manipular e gerenciar grandes quantidades de dados de maneira confiável, junto com o interesse em descentralizar a informação, fez com que fossem impulsionados os estudos e desenvolvimentos na área de banco de dados distribuídos (BDD).

Diferente do banco de dados convencional, também conhecido como centralizado, no qual toda a massa de dados fica em um único local, no BDD, os dados ficam distribuídos em diversos servidores, e esses servidores podem estar fisicamente próximos ou, em alguns casos, geograficamente distantes.



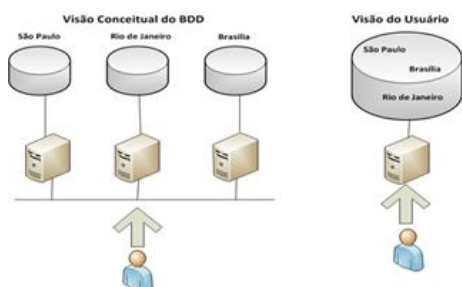
**Arquitetura de Banco de Dados Centralizado**

Como os dados nesse ambiente muitas vezes estão replicados entre servidores, além de melhorar a eficiência no desempenho, consegue prover uma melhora na segurança dos dados em caso de desastres, pois não existe mais um único ponto de falha.



**Arquitetura de Banco de Dados Distribuída**

A arquitetura de BDD é totalmente transparente para os usuários que utilizam aplicações nessa arquitetura distribuída, ou seja, para o usuário, os dados estão centralizados em um único servidor, mas na verdade eles podem estar “espalhados”, distribuídos em vários locais fisicamente separados.



**Arquitetura de Banco de Dados Distribuído**

Na implementação desse ambiente distribuído, existem vários desafios, e manter a consistência dos dados certamente está entre os mais importantes e complexos. Garantir que os dados estão consistentes é altamente complexo, pois uma única transação pode manipular dados em diversos servidores. Para exemplificar, imagine uma instituição financeira na qual o banco de dados está distribuído em três grandes



cidades brasileiras – São Paulo, Brasília e Rio de Janeiro -, e transações distribuídas entre esses servidores são executadas o dia todo para armazenar informações e movimentações de seus clientes.

Agora imagine a situação em que o cliente José sacou R\$ 1.000,00 de sua conta corrente, e essa movimentação deve ser replicada para todos os servidores envolvidos na transação; caso contrário, o saldo de José ficará inconsistente, gerando informações diferentes em determinado servidor.

Isso pode causar grandes transtornos, pois teoricamente o usuário acessa os dados em servidores de acordo com a localidade, diminuindo assim a latência e retornando o resultado com mais rapidez, ou seja, se o usuário está acessando sua conta em São Paulo, o sistema busca os dados nos servidores de São Paulo, e assim por diante. Como no exemplo acima José sacou R\$ 1.000,00 de sua conta, imagine se ele estivesse em São Paulo e essa informação não fosse replicada para o servidor de Brasília? No servidor de São Paulo, ele estaria com determinado saldo e em Brasília com R\$ 1.000,00 a mais em sua conta.

Esse é um dos exemplos, mas poderia ser o estoque de um produto que foi vendido, porém não foi executada baixa em todos os servidores envolvidos na transação, gerando informações falsas sobre a quantidade em estoque desse produto. Claro que esse ambiente é só um exemplo simples para demonstrar a consistência dos dados.

A fim de garantir toda essa segurança e confiabilidade dos dados, o BDD preserva as propriedades ACID, onde:

- A – Atomicidade: Indica que a transação deve ter todas as suas ações concluídas ou não. Caso todas as ações da transação sejam terminadas com sucesso, então é executado o Commit na transação inteira; caso contrário, a transação inteira deve ser revertida, Rollback.
- C – Consistência: A execução de uma transação isolada preserva a consistência do banco de dados.
- I – Isolamento: Cada transação não toma conhecimento de outras transações concorrentes.
- D – Durabilidade: Depois que transação for completada com sucesso (Commit), as mudanças que ela faz no banco de dados persistem.

### **Protocolos de Efetivação**

Em ambiente distribuído, podem existir transações locais e globais. Transações locais seriam aquelas que iniciam e terminam no mesmo nó, não afetando outros servidores para realizar determinada transação. Já as globais seriam aquelas que iniciam em determinado nó e terminam em outro nó, ou iniciam e terminam no mesmo nó, porém no meio da transação foi necessário acessar outro nó para realizar alguma operação, afetando assim outro servidor em algum momento do processamento.

Os protocolos de efetivação garantem integridade de transação distribuída que atinge mais de um servidor, fazendo com que a transação só seja efetuada (Commit) ou abortada (Rollback) quando todos os servidores entram em um acordo.

- Two-Phase Commit

O protocolo de efetivação Two-Phase Commit é composto por duas fases, em que todos os servidores envolvidos na transação T entram em acordo se efetivaram (Commit) ou abortaram (Rollback) a transação.

- Three-Phase Commit

O protocolo Three-Phase Commit é uma continuação do protocolo de duas fases em que é adicionada uma terceira fase que busca a diminuição de falhas do coordenador. Com a implementação da terceira fase, a troca de dados entre coordenador e gerenciadores de transação aumenta o grau de complexidade e proporciona um maior tráfego na rede (overhead).

### **Vantagens na utilização de BDD**

- A confiabilidade assim como a disponibilidade visam ao funcionamento do sistema mesmo com falhas, mas sem violar a consistência do banco de dados.

- Autonomia local, um departamento pode controlar seus dados (já que é o mais familiarizado com eles).
- O aumento de desempenho é uma vantagem em um sistema de banco de dados distribuídos, a consulta pode ser subdividida e realizada paralelamente.
- Maior disponibilidade – uma falha em um banco de dados afetará somente um fragmento, em vez do banco de dados inteiro.



Falha em um servidor

### Desvantagens na utilização de BDD

- A complexidade dos problemas é uma desvantagem, porque os problemas são mais complexos do que em bancos de dados centralizados, pois além de haver os problemas que são comuns em banco de dados centralizados, que normalmente ocorrem nos servidores locais, haverá os problemas que surgem com a comunicação entre esses servidores locais.
- Implantação mais cara – o aumento da complexidade e uma infraestrutura mais extensa significam custo extra de trabalho.
- Falta de padrões – ainda não há metodologias e ferramentas para ajudar usuários a converterem um SGBD centralizado para um SGBD distribuído.

Com o crescente avanço tecnológico em comunicação de dados, BDD passou a ser uma possibilidade para trabalhar com banco de dados em grande escala, possibilitando a distribuição geográfica (WAN). Como foi demonstrado, existem algumas desvantagens devido à complexidade e aos custos, mas com certeza os pontos positivos as superam.

Essa área é muito ampla, demanda muito estudo e, conseqüentemente, vários pontos de dúvidas interessantes, pois a construção de uma arquitetura como essa envolve vários departamentos, como infraestrutura, segurança da informação e banco de dados.

Um sistema de banco de dados distribuído (BDD) consiste em uma relação de nós, cada qual podendo participar na execução de transações que acessam dados em um ou mais nós. Em um sistema de banco de dados distribuído, o banco de dados é armazenado em diversos computadores (nós).

Os computadores, em um sistema distribuído, comunicam-se uns com os outros por intermédio de vários meios de comunicação, tais como: redes de alta velocidade, redes sem fio ou linhas telefônicas, eles não compartilham a memória principal e o relógio.

A diferença principal entre sistemas de banco de dados centralizados e distribuídos é que no primeiro os dados estão localizados em um único lugar, enquanto que no outro os dados residem em diversos locais. Esta distribuição de dados é motivo de muitas preocupações e dificuldades.

Os processadores em um sistema distribuído podem variar em tamanho e função, podendo incluir microcomputadores, estações de trabalho, minicomputadores e sistemas de computadores de uso em geral.

Estes processadores são geralmente chamados de nós, dependendo do contexto no qual eles estejam mencionados. Usa-se principalmente o termo nó (lugar, posição), a fim de enfatizar a distribuição física destes sistemas. Veja o exemplo na Figura 1:

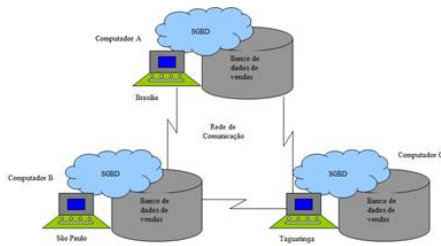


Figura 1: Figura demonstrando o sistema de banco de dados distribuído.

### Armazenamento Distribuído Dos Dados

Uma relação  $r$  (ou tabela) possui diversos enfoques para o armazenamento em um banco de dados distribuído (BDD):

- Replicação: o sistema mantém réplicas idênticas da relação, onde cada réplica é armazenada em sites diferentes, resultando na replicação dos dados
- Fragmentação: a relação é particionada em vários fragmentos, onde cada fragmento é armazenado em um site diferente
- Replicação e fragmentação: a relação é particionada em vários segmentos, e o sistema mantém diversas réplicas de cada fragmento

#### Replicação de dados

A replicação de dados significa que um determinado objeto de dados lógico pode possuir diversos representantes armazenados, em nós. O grau de suporte para a replicação é um pre-requisito para atingir o verdadeiro potencial de um sistema distribuído.

#### Fragmentação de dados

Uma relação é dividida em fragmentos, onde cada fragmento contém informação suficiente para permitir a reconstrução da relação original.

Existem duas formas de fazer a fragmentação:

- Fragmentação Horizontal: divide a relação separando as tuplas de  $r$  em dois ou mais fragmentos.
- Fragmentação Vertical: divide a relação pela decomposição do esquema  $R$  da relação  $r$ .

#### Fragmentação e Replicação de Dados

As técnicas de fragmentação e replicação podem ser aplicadas sucessivamente a uma mesma relação. Um fragmento pode ser replicado, e as réplicas podem ser fragmentadas novamente e assim por diante.

### Processamento De Consultas Distribuídas

A transparência para leitura é mais fácil de se conseguir e manter do que a transparência para atualização. O maior problema para a atualização é garantir que todas as réplicas e fragmentos sejam atualizados, após uma atualização em uma das réplicas ou fragmentos. A atualização deve ser prolongada para todas as cópias (réplicas e fragmentos) existentes no sistema.

Um dos fatores mais importantes no desempenho de uma consulta, em uma base centralizada, é a quantidade de acesso a disco necessária para atingir o resultado. Em um banco distribuído os problemas aumentam, pois existe também a preocupação com a transmissão de dados na rede. Um fator interessante para a consulta realizada em uma base distribuída é que para os diversos sites podem processar partes da consulta em paralelo.

Na realização de uma consulta simples (trivial), como consultar todas as tuplas da relação CONTA, pode caracterizar um processamento não tão trivial, pois CONTA pode estar fragmentada, replicada ou ambas.

### Transações

O acesso a diversos itens de dados em um sistema distribuído é normalmente acompanhado de transações que têm de preservar as propriedades ACID:

- A: Atomicidade
- C: Consistência
- I: Isolamento
- D: Durabilidade

### Características da ACID

- Atomicidade: Todas as operações da transação são refletidas corretamente no BD ou nenhuma será.
- Consistência: A execução de uma transação isolada preserva a consistência do banco de dados.
- Isolamento: Cada transação não toma conhecimento de outras transações concorrentes.
- Durabilidade: Depois da transação completar-se com sucesso, as mudanças que ela faz no banco de dados persistem.

### Tipos de transação

- Locais: mantem acesso e atualizam somente a base de dados local.
- Globais: mantem acesso e atualizam diversas bases de dados locais.

### Funções adicionais

- Rastreamento de dados.
- Processamento de consultas distribuídas.
- Gerenciamento de transações distribuídas.
- Gerenciamento de dados replicados.
- Recuperação de banco de dados distribuído.
- Segurança.
- Gerenciamento do diretório distribuído

### Vantagens

- Gerenciamento de dados distribuído com níveis diferentes de transparência.
  - Transparência de distribuição ou de rede.
  - Transparência de replicação.
  - Transparência de fragmentação.
- Melhoria da confiabilidade e na disponibilidade.
- Melhoria no desempenho.
- Expansão mais fácil.



## Engenharia de Software

Desenvolvimento de software é uma atividade de crescente importância na sociedade contemporânea. A utilização de computadores nas mais diversas áreas do conhecimento humano tem gerado uma crescente demanda por soluções computadorizadas.

Para os iniciantes na Ciência de Computação, desenvolver software é, muitas vezes, confundido com programação. Essa confusão inicial pode ser atribuída, parcialmente, pela forma como as pessoas são introduzidas nesta área de conhecimento, começando por desenvolver habilidades de raciocínio lógico, através de programação e estruturas de dados. Aliás, nada há de errado nessa estratégia. Começamos resolvendo pequenos problemas que gradativamente vão aumentando de complexidade, requerendo maiores conhecimentos e habilidades.

Entretanto, chega-se a um ponto em que, dado o tamanho ou a complexidade do problema que se pretende resolver, essa abordagem individual, centrada na programação não é mais indicada. De fato, ela só é aplicável para resolver pequenos problemas, tais como calcular médias, ordenar conjuntos de dados etc, envolvendo basicamente o projeto de um único algoritmo. Contudo, é insuficiente para problemas grandes e complexos, tais como aqueles tratados na automação bancária, na informatização de portos ou na gestão empresarial. Em tais situações, uma abordagem de engenharia é necessária.

Observando outras áreas, tal como a Engenharia Civil, podemos verificar que situações análogas ocorrem. Por exemplo, para se construir uma casinha de cachorro, não é necessário elaborar um projeto de engenharia civil, com plantas baixa, hidráulica e elétrica, ou mesmo cálculos estruturais. Um bom pedreiro é capaz de resolver o problema a contento. Talvez não seja dada a melhor solução, mas o produto resultante pode atender aos requisitos pré-estabelecidos. Essa abordagem, contudo, não é viável para a construção de um edifício.

Nesse caso, é necessário realizar um estudo aprofundado, incluindo análises de solo, cálculos estruturais etc, seguido de um planejamento da execução da obra e desenvolvimento de modelos (maquetes e plantas de diversas naturezas), até a realização da obra, que deve ocorrer por etapas, tais como fundação, alvenaria, acabamento etc. Ao longo da realização do trabalho, deve-se realizar um acompanhamento para verificar prazos, custos e a qualidade do que se está construindo.

Visando melhorar a qualidade dos produtos de software e aumentar a produtividade no processo de desenvolvimento, surgiu a Engenharia de Software. A Engenharia de Software trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de software.

Assim como em outras áreas, em uma abordagem de engenharia de software, inicialmente o problema a ser tratado deve ser analisado e decomposto em partes menores, em uma abordagem “dividir para conquistar”. Para cada uma dessas partes, uma solução deve ser elaborada. Solucionados os subproblemas isoladamente, é necessário integrar as soluções. Para tal, uma arquitetura deve ser estabelecida. Para apoiar a resolução de problemas, procedimentos (métodos, técnicas, roteiros etc) devem ser utilizados, bem como ferramentas para parcialmente automatizar o trabalho.

Neste cenário, muitas vezes não é possível conduzir o desenvolvimento de software de maneira individual. Pessoas têm de trabalhar em equipes, o esforço tem de ser planejado, coordenado e acompanhado, bem como a qualidade do que se está produzindo tem de ser sistematicamente avaliada.

## Qualidade de Software

Uma vez que um dos objetivos da Engenharia de Software é melhorar a qualidade dos produtos de software desenvolvidos, uma questão deve ser analisada: O que é qualidade de software?

Se perguntarmos a um usuário, provavelmente, ele dirá que um produto de software é de boa qualidade se ele satisfizer suas necessidades, sendo fácil de usar, eficiente e confiável. Essa é uma perspectiva externa de observação pelo uso do produto. Por outro lado, para um desenvolvedor, um produto de boa qualidade tem de ser fácil de manter, sendo o produto de software observado por uma perspectiva interna. Já para um cliente, o produto de software deve agregar valor a seu negócio (qualidade em uso).

Em última instância, podemos perceber que a qualidade é um conceito com múltiplas facetas (perspectivas de usuário, desenvolvedor e cliente) e que envolve diferentes características (por exemplo, usabilidade, confiabilidade, eficiência, manutenibilidade, portabilidade, segurança, produtividade) que devem ser alcançadas em níveis diferentes, dependendo do propósito do software. Por exemplo, um sistema de tráfego aéreo tem de ser muito mais eficiente e confiável do que um editor de textos. Por outro lado, um software educacional a ser usado por crianças deve primar muito mais pela usabilidade do que um sistema de venda de passagens aéreas a ser operado por agentes de turismo especializados.

Que há de comum nas várias perspectivas discutidas acima é que todas elas estão focadas no produto de software. Ou seja, estamos falando de qualidade do produto. Entretanto, como garantir que o produto final de software apresenta essas características? Apenas avaliar se o produto final as apresenta é uma abordagem indesejável para o pessoal de desenvolvimento de software, tendo em vista que a constatação a posteriori de que o software não apresenta a qualidade desejada pode implicar na necessidade de refazer grande parte do trabalho. É necessário, pois, que a qualidade seja incorporada ao produto ao longo de seu processo de desenvolvimento. De fato, a qualidade dos produtos de software depende fortemente da qualidade dos processos usados para desenvolvê-los e mantê-los.

Seguindo uma tendência de outros setores, a qualidade do processo de software tem sido apontada como fundamental para a obtenção da qualidade do produto. Abordagens de qualidade de processo, tal como a série de padrões ISO 9000, sugerem que melhorando a qualidade do processo de software, é possível melhorar a qualidade dos produtos resultantes. A premissa por detrás dessa afirmativa é a de que processos bem estabelecidos, que incorporam mecanismos sistemáticos para acompanhar o desenvolvimento e avaliar a qualidade, no geral, conduzem a produtos de qualidade. Por exemplo, quando se diz que um fabricante de eletrodomésticos é uma empresa certificada ISO 9001 (uma das normas da série ISO 9000), não se está garantindo que todos os eletrodomésticos por ele produzidos são produtos de qualidade. Mas sim que ele tem um bom processo produtivo, o que deve levar a produtos de qualidade.

Um processo de software, em uma abordagem de Engenharia de Software, envolve diversas atividades que podem ser classificadas quanto ao seu propósito em:

**Atividades de Desenvolvimento (ou técnicas ou de Construção):** são as atividades diretamente relacionadas ao processo de desenvolvimento do software, ou seja, que contribuem diretamente para o desenvolvimento do produto de software a ser entregue ao cliente. São exemplos de atividades de desenvolvimento: especificação e análise de requisitos, projeto e implementação.

**Atividades de Gerência:** são aquelas relacionadas ao planejamento e acompanhamento gerencial do projeto, tais como realização de estimativas, elaboração de cronogramas, análise dos riscos do projeto etc.

**Atividades de Garantia da Qualidade:** são aquelas relacionadas com a garantia da qualidade do produto em desenvolvimento e do processo de software utilizado, tais como revisões e inspeções de produtos (intermediários ou finais) do desenvolvimento.

As atividades de desenvolvimento formam a espinha dorsal do desenvolvimento e, de maneira geral, são realizadas segundo uma ordem estabelecida no planejamento. As atividades de gerência e de controle da qualidade são, muitas vezes, ditas atividades de apoio, pois não estão ligadas diretamente à construção do produto final: o software a ser entregue para o cliente, incluindo toda a documentação necessária. Essas atividades, normalmente, são realizadas ao longo de todo o ciclo de vida, sempre que necessário ou em pontos pré- estabelecidos durante o planejamento, ditos marcos ou pontos de controle. A figura 1.1 mostra a relação entre esses tipos de atividades.

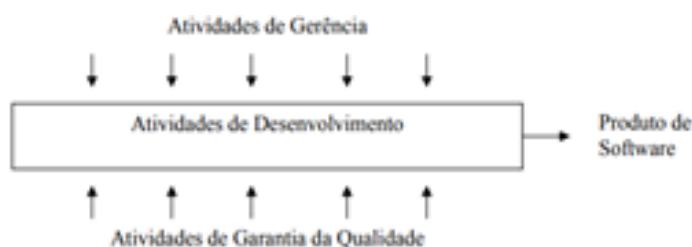


Figura 1.1 – Atividades do Processo de Software

## A Organização deste Texto

Nesta disciplina, procuramos oferecer uma visão geral da Engenharia de Software, discutindo as principais atividades do processo e como realizá-las. Nos capítulos que se seguem, os seguintes temas são abordados:

Capítulo 2 – Processo de Software – enfoca os processos de software, os elementos que compõem um processo, a definição de processos para projetos, modelos de processo, normas e modelos de qualidade de processo de software e a automatização do processo de software.

Capítulo 3 – Planejamento e Gerência de Projetos – são abordadas as principais atividades da gerência de projetos, a saber: definição do escopo do projeto, estimativas, análise de riscos, elaboração de cronograma, elaboração do plano de projeto e acompanhamento de projetos.

Capítulo 4 – Gerência da Qualidade – trata das principais atividades de garantia da qualidade, incluindo a medição e métricas associadas, revisões e inspeções e a gerência de configuração de software.

Capítulo 5 – Especificação e Análise de Requisitos – são discutidos o que é um requisito de software e tipos de requisitos. A seguir, são abordadas a especificação e a análise de requisitos, usando o método da Análise Essencial de Sistemas como base.

Capítulo 6 – Projeto de Sistema – aborda os conceitos básicos de projeto de sistemas, tratando da arquitetura do sistema a ser desenvolvido e do projeto de seus módulos, segundo a abordagem do Projeto Estruturado de Sistemas.

Capítulo 7 – Implementação e Testes – são enfocadas as atividades de implementação e testes, sendo esta última tratada em diferentes níveis, a saber: teste de unidade, teste de integração, teste de validação e teste de sistema.

Capítulo 8 – Entrega e Manutenção – discute as questões relacionadas à entrega do sistema para o cliente, tais como o treinamento e a documentação de entrega, e a atividade de manutenção do sistema.

## Processo de Software

O Que É Um Processo De Software.

Um processo de software pode ser visto como o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software. Um processo eficaz deve, claramente, considerar as relações entre as atividades, os artefatos produzidos no desenvolvimento, as ferramentas e os procedimentos necessários e a habilidade, o treinamento e a motivação do pessoal envolvido.

## Elementos Que Compõem Um Processo De Software

Processo de Software

Processos

Atividades

Pré-atividades Sub-atividades Artefatos

Insumos Produtos

Recursos

Recursos Humanos Ferramentas de Software Hardware

Procedimentos

Métodos Técnicas Roteiros

## Definição de Processos



Há vários aspectos a serem considerados na definição de um processo de software. No centro da arquitetura de um processo de desenvolvimento estão as atividades-chave desse processo: análise e especificação de requisitos, projeto, implementação e testes, que são a base sobre a qual o processo de desenvolvimento deve ser construído. Entretanto, a definição de um processo envolve a escolha de um modelo de ciclo de vida, o detalhamento (decomposição) de suas macro-atividades, a escolha de métodos, técnicas e roteiros (procedimentos) para a sua realização e a definição de recursos e artefatos necessários e produzidos.

Um processo de software não pode ser definido de forma universal. Para ser eficaz e conduzir à construção de produtos de boa qualidade, um processo deve ser adequado ao domínio da aplicação e ao projeto específico. Deste modo, processos devem ser definidos caso a caso, considerando-se as especificidades da aplicação, a tecnologia a ser adotada na sua construção, a organização onde o produto será desenvolvido e o grupo de desenvolvimento.

Em suma, o objetivo de se definir um processo de software é favorecer a produção de sistemas de alta qualidade, atingindo as necessidades dos usuários finais, dentro de um cronograma e um orçamento previsíveis.

A escolha de um modelo de ciclo de vida (ou modelo de processo) é o ponto de partida para a definição de um processo de desenvolvimento de software. Um modelo de ciclo de vida organiza as macro-atividades básicas, estabelecendo precedência e dependência entre as mesmas. Para maiores detalhes sobre os modelos de processo existentes, vide as seguintes referências sugeridas: [1,2,3].

Um modelo de ciclo de vida pode ser entendido como passos ou atividades que devem ser executados durante um projeto. Para a definição completa do processo, a cada atividade, devem ser associados técnicas, ferramentas e critérios de qualidade, entre outros, formando uma base sólida para o desenvolvimento. Adicionalmente, outras atividades tipicamente de cunho gerencial, devem ser definidas, entre elas atividade de gerência e de controle e garantia da qualidade.

De maneira geral, o ciclo de vida de um software envolve as seguintes fases:

**Planejamento:** O objetivo do planejamento de projeto é fornecer uma estrutura que possibilite ao gerente fazer estimativas razoáveis de recursos, custos e prazos. Uma vez estabelecido o escopo de software, uma proposta de desenvolvimento deve ser elaborada, isto é, um plano de projeto deve ser elaborado configurando o processo a ser utilizado no desenvolvimento de software. À medida que o projeto progride, o planejamento deve ser detalhado e atualizado regularmente. Pelo menos ao final de cada uma das fases do desenvolvimento (análise e especificação de requisitos, projeto, implementação e testes), o planejamento como um todo deve ser revisto e o planejamento da etapa seguinte deve ser detalhado. O planejamento e o acompanhamento do progresso fazem parte do processo de gerência de projeto.

**Análise e Especificação de Requisitos:** Nesta fase, o processo de levantamento de requisitos é intensificado. O escopo deve ser refinado e os requisitos identificados. Para entender a natureza do software a ser construído, o engenheiro de software tem de compreender o domínio do problema, bem como a funcionalidade e o comportamento esperados. Uma vez identificados os requisitos do sistema a ser desenvolvido, estes devem ser modelados, avaliados e documentados. Uma parte vital desta fase é a construção de um modelo descrevendo o que o software tem de fazer (e não como fazê-lo).

**Projeto:** Esta fase é responsável por incorporar requisitos tecnológicos aos requisitos essenciais do sistema, modelados na fase anterior e, portanto, requer que a plataforma de implementação seja conhecida. Basicamente, envolve duas grandes etapas: projeto da arquitetura do sistema e projeto detalhado. O objetivo da primeira etapa é definir a arquitetura geral do software, tendo por base o modelo construído na fase de análise de requisitos. Esta arquitetura deve descrever a estrutura de nível mais alto da aplicação e identificar seus principais componentes. O propósito do projeto detalhado é detalhar o projeto do software para cada componente identificado na etapa anterior. Os componentes de software devem ser sucessivamente refinados em níveis de maior detalhamento, até que possam ser codificados e testados.

**Implementação:** O projeto deve ser traduzido para uma forma passível de execução pela máquina. A fase de implementação realiza esta tarefa, isto é, cada unidade de software do projeto detalhado é implementada.

Testes: inclui diversos níveis de testes, a saber, teste de unidade, teste de integração e teste de sistema. Inicialmente, cada unidade de software implementada deve ser testada e os resultados documentados. A seguir, os diversos componentes devem ser integrados sucessivamente até se obter o sistema. Finalmente, o sistema como um todo deve ser testado.

Entrega e Implantação: uma vez testado, o software deve ser colocado em produção. Para tal, contudo, é necessário treinar os usuários, configurar o ambiente de produção e, muitas vezes, converter bases de dados. O propósito desta fase é estabelecer que o software satisfaz os requisitos dos usuários. Isto é feito instalando o software e conduzindo testes de aceitação (validação). Quando o software tiver demonstrado prover as capacidades requeridas, ele pode ser aceito e a operação iniciada.

Operação: nesta fase, o software é utilizado pelos usuários no ambiente de produção.

Manutenção: Indubitavelmente, o software sofrerá mudanças após ter sido entregue para o usuário. Alterações ocorrerão porque erros foram encontrados, porque o software precisa ser adaptado para acomodar mudanças em seu ambiente externo, ou porque o cliente necessita de funcionalidade adicional ou aumento de desempenho. Muitas vezes, dependendo do tipo e porte da manutenção necessária, essa fase pode requerer a definição de um novo processo, onde cada uma das fases precedentes é re-aplicada no contexto de um software existente ao invés de um novo.

São fatores que influenciam a definição de um processo: Tipo de Software (p.ex., sistema de informação, sistema de tempo real etc), Paradigma (estruturado, orientado a objetos etc), Domínio da Aplicação, Tamanho e Complexidade, Características da Equipe etc.

Embora diferentes projetos requeiram processos com características específicas para atender às suas particularidades, é possível estabelecer um conjunto de ativos de processo (sub-processos, atividades, sub-atividades, artefatos, recursos e procedimentos) a ser utilizado na definição de processos de software de uma organização.

Essas coleções de ativos de processo de software constituem os chamados processos padrão de desenvolvimento de software. Processos para projetos específicos podem, então, ser definidos a partir da instanciação do processo de software padrão da organização, levando em consideração suas características particulares. Esses processos instanciados são ditos processos de projeto.

De fato, o modelo de definição de processos baseado em processos padrão pode ser estendido para comportar vários níveis. Primeiro, pode-se definir um processo padrão da organização, contendo os ativos de processo que devem fazer parte de todos os processos de projeto da organização.

Esse processo padrão pode ser especializado para agregar novos ativos de processo, considerando aspectos, tais como tecnologias de desenvolvimento, paradigmas ou domínios de aplicação.

Assim, obtêm-se processos mais completos, que consideram características da especialização desejada.

Por fim, a partir de um processo padrão ou de um processo especializado, é possível instanciar um processo de projeto, que será o processo a ser utilizado em um projeto de software específico. Para definir esse processo devem ser consideradas as particularidades de cada projeto.

Para apoiar a definição de processos, diversas normas e modelos de qualidade de processo de software foram propostas, dentre elas: ISO 9001, ISO/IEC 12207, ISO/IEC 15504, CMM e CMMI. O objetivo dessas normas e modelos de qualidade é apontar características que um bom processo de software tem de apresentar, deixando a organização livre para estruturar essas características segundo sua própria cultura.

Assim, usando essas normas e modelos de qualidade, em uma abordagem de definição de processos em níveis, é possível definir processos para projetos específicos, que levem em consideração as particularidades de cada projeto, sem, no entanto, desconsiderar aspectos importantes para se atingir a qualidade do processo. A figura 2.1 ilustra essa abordagem de definição de processos de software em níveis.

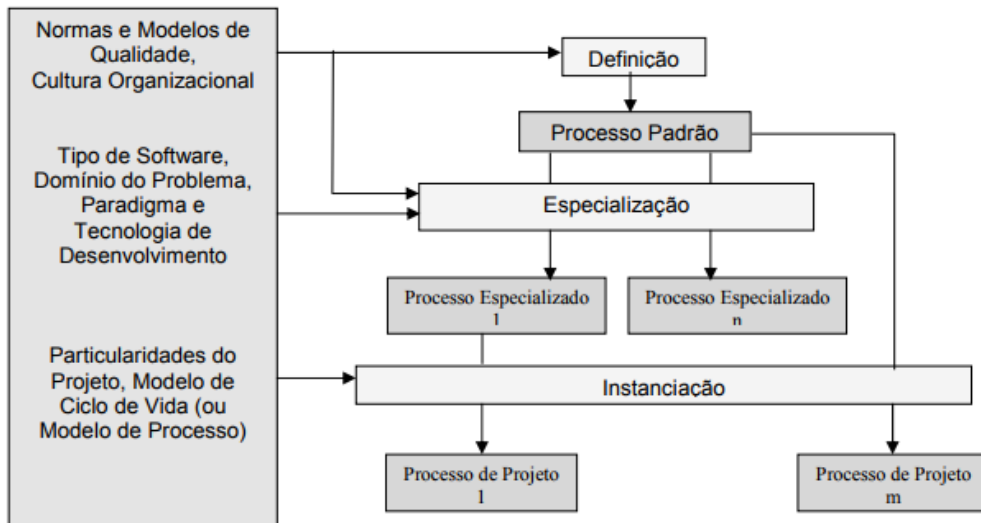


Figura 2.1 – Modelo para Definição de Processos em Níveis

Com o aumento da complexidade dos processos de software, passou a ser imprescindível o uso de ferramentas e ambientes de apoio à realização de suas atividades, visando, sobretudo, a atingir níveis mais altos de qualidade e produtividade. Ferramentas CASE (Computer Aided Software Engineering) passaram, então, a ser utilizadas para apoiar a realização de atividades específicas, tais como planejamento e análise e especificação de requisitos.

Apesar dos benefícios do uso de ferramentas CASE individuais, atualmente, o número e a variedade de ferramentas têm crescido a tal ponto que levou os engenheiros de software a pensarem não apenas em automatizar os seus processos, mas sim em trabalhar com diversas ferramentas que interajam entre si e forneçam suporte a todo ciclo de vida do desenvolvimento, dando origem ao Ambientes de Desenvolvimento de Software (ADSs).

ADSs buscam combinar técnicas, métodos e ferramentas para apoiar o engenheiro de software na construção de produtos de software, abrangendo todas as atividades inerentes ao processo: gerência, desenvolvimento e controle da qualidade.

### Gerência De Projetos De Software

A Gerência de Projetos de Software envolve o Produto, o Processo e as Pessoas envolvidas no projeto.

#### Produto

No que se refere ao produto, a primeira coisa a se fazer é definir o escopo do projeto. Para tal, é necessário fazer um levantamento de requisitos inicial. A ideia é decompor o problema, em uma abordagem “dividir para conquistar”. Inicialmente, o sistema deve ser decomposto em subsistemas que são, por sua vez, decompostos em módulos. Os módulos podem, ainda, ser recursivamente decompostos em sub-módulos ou funções, até que se tenha uma visão geral das funcionalidades a serem tratadas no projeto.

#### Pessoas:

Em um projeto de software, há várias pessoas envolvidas, exercendo diferentes papéis, tais como: Gerente de Projeto, Desenvolvedor (Analistas, Projetistas, Engenheiro de Software, Programadores, Engenheiros de Testes), Gerente da Qualidade, Clientes, Usuários. O número de papéis e suas denominações podem ser bastante diferentes dependendo da organização e até mesmo do projeto.

As pessoas trabalhando em um projeto são organizadas em equipes. Assim, o conceito de equipe pode ser visto como um conjunto de pessoas trabalhando em diferentes tarefas, mas objetivando uma meta comum. Essa não é uma característica do desenvolvimento de software, mas da organização de pessoas em qualquer atividade humana. Assim, a definição de equipe dada é válida para uma ampla variedade de situações, tal como uma equipe de futebol.

Para a boa formação de equipes, devem ser definidos os papéis necessários e devem ser considerados aspectos fundamentais, a saber liderança, organização (estrutura da equipe) e coordenação. Além disso, há diversos fatores que afetam a formação de equipes: relacionamentos inter-pessoais, tipo do projeto, criatividade etc.

No que se refere à organização / estrutura das equipes, há diversos tipos de equipes, tais como os citados por Pressman: Democrática Descentralizada, Controlada Descentralizada, Controlada Centralizada.

Por fim, na formação de equipes deve-se levar em conta o tamanho da equipe. Quanto maior o número de membros da equipe, maior a quantidade de caminhos possíveis de comunicação, o que pode ser um problema, uma vez que o número de pessoas que podem se comunicar com outras pode afetar a qualidade do produto resultante.

**Estrutura de Divisão do Trabalho:**

Uma boa gerência de projetos começa com a fusão das visões de Produto, Processo e Pessoas. Cada função ou módulo a ser desenvolvido pela equipe do projeto deve passar pelas várias atividades definidas no processo de software. Essa pode ser uma base bastante efetiva para a elaboração de estimativas, incluindo a alocação de recursos, já que é sempre mais fácil estimar porções menores de trabalho. Assim, é útil elaborar uma estrutura de divisão do trabalho, considerando duas dimensões principais: produto e processo, como mostra a Tabela 3.1.

Tabela 3.1 – Estrutura de Divisão do Trabalho considerando a fusão das visões de produto e processo.

	<b>Atividades do processo</b>			
	<b>Análise e Especificação de Requisitos</b>	<b>Projeto</b>	<b>Implementação</b>	<b>Testes</b>
<b>Módulos / Funções</b>				
Módulo 1				
Módulo 2				
....				

Atividades Típicas da Gerência de Projetos:

A gerência de projetos envolve a realização de diversas atividades, abaixo relacionadas:

Determinação do Escopo do Software Definição do Processo de Software do Projeto Realização de Estimativas

Estimativa de Tamanho Estimativa de Esforço

Estimativa (Alocação) de Recursos

Estimativa de Tempo (Elaboração do Cronograma do Projeto) Estimativa de Custos

Gerência de Riscos

Elaboração do Plano de Projeto

**O Planejamento e o Acompanhamento do Projeto**

As atividades acima relacionadas são realizadas diversas vezes ao longo do projeto. Tipicamente, no início do projeto, elas têm de ser realizadas para produzir uma primeira visão gerencial sobre o projeto, quando são conjuntamente denominadas de planejamento do projeto. À medida que o projeto avança, contudo, o plano do projeto deve ser revisto, uma vez que problemas podem surgir ou porque se ganha

um maior entendimento sobre o problema. Essas revisões do plano de projeto são ditas atividades de acompanhamento do projeto e tipicamente são realizadas nos marcos do projeto.

Os marcos de um projeto são estabelecidos durante a definição do processo e tipicamente correspondem ao término de atividades importantes do processo de desenvolvimento, tais como Análise e Especificação de Requisitos, Projeto e Implementação. O propósito de um marco é garantir que os interessados tenham uma visão do andamento do projeto e concordem com os rumos a serem tomados.

Em uma atividade de acompanhamento do projeto, o escopo pode ser revisto, alterações no processo podem ser necessárias, bem como devem ser monitorados os riscos e revisadas as estimativas (de tamanho, esforço, tempo e custo).

### **Estimativas**

Antes mesmo de serem iniciadas as atividades técnicas de um projeto, o gerente e a equipe de desenvolvimento devem estimar o trabalho a ser realizado, os recursos necessários, o tempo de duração e, por fim, o custo do projeto. Apesar das estimativas serem um pouco de arte e um pouco de ciência, esta importante atividade não deve ser conduzida desordenadamente. As estimativas podem ser consideradas a fundação para todas as outras atividades de planejamento de projeto. Para alcançar boas estimativas de prazo, esforço e custo, existem algumas opções:

Postergar as estimativas até o mais tarde possível no projeto.

Usar técnicas de decomposição.

Usar um ou mais modelos empíricos para estimativas de custo e esforço.

Basear as estimativas em projetos similares que já tenham sido concluídos.

A primeira opção, apesar de ser atraente, não é prática, pois estimativas devem ser providas logo no início do projeto (fase de planejamento do projeto). No entanto, deve-se reconhecer que quanto mais tarde for feita a estimativa, maior o conhecimento do projeto e menores as chances de se cometer erros. Assim, é fundamental revisar as estimativas na medida em que o projeto avança (atividades de acompanhamento do projeto).

Técnicas de decomposição, a segunda opção, usam, conforme discutido anteriormente, a abordagem “dividir para conquistar” na realização de estimativas, através da decomposição do projeto em módulos / funções (decomposição do produto) e atividades mais importantes (decomposição do processo). Assim, uma tabela como a Tabela 3.1 pode ser utilizada para estimar, por exemplo, tamanho ou esforço.

Modelos empíricos, tipicamente, usam fórmulas matemáticas, derivadas em experimentos, para prever esforço como uma função de tamanho (linhas de código ou pontos de função). Entretanto, deve-se observar que os dados empíricos que suportam a maioria desses modelos são derivados de um conjunto limitado de projetos. Além disso, fatores culturais da organização não são considerados no uso de modelos empíricos, pois os projetos que constituem a base de dados do modelo são externos à organização. Apesar de suas limitações, modelos empíricos podem ser úteis como um ponto de partida para organizações que ainda não têm dados históricos, até que a organização possa estabelecer suas próprias correlações.

Finalmente, na última opção, dados de projetos anteriores armazenados em um repositório de experiências da organização podem prover uma perspectiva histórica importante e ser uma boa fonte para estimativas. Através de mecanismos de busca, é possível recuperar projetos similares, suas estimativas e lições aprendidas, que podem ajudar a elaborar estimativas mais precisas. Nesta abordagem, os fatores culturais são considerados, pois os projetos foram desenvolvidos na própria organização.

Vale frisar que essas abordagens não são excludentes; muito pelo contrário. O objetivo é ter várias formas para realizar estimativas e usar seus resultados para se chegar a estimativas mais precisas.

Quando se fala em estimativas, está-se tratando na realidade de diversos tipos de estimativas: tamanho, esforço, recursos, tempo e custos. Geralmente, a realização de estimativas começa pelas estima-

tivas de tamanho. A partir delas, estima-se o esforço necessário e, na sequência, alocam-se os recursos necessários, elaboram-se o cronograma do projeto (estimativas de tempo) e, por fim, estima-se o custo do projeto.

Estimativa De Tamanho:

Entre as várias formas de se medir tamanho de um software, a mais simples, direta e altamente utilizada é a contagem do número de linhas de código (Lines Of Code - LOC) dos programas fonte. Existem alguns estudos que demonstram a alta correlação entre essa métrica e o tempo necessário para se desenvolver um sistema. Entretanto, o uso dessa métrica apresenta algumas desvantagens. Primeiro, verifica-se que ela é fortemente ligada à tecnologia empregada, sobretudo a linguagem de programação. Segundo, pode ser difícil estimar essa grandeza no início do desenvolvimento, sobretudo se não houver dados históricos relacionados com a linguagem de programação utilizada no projeto.

Visando possibilitar a realização de estimativas de tamanho mais cedo no processo de software, foram propostas outras métricas em um nível de abstração mais alto. O exemplo mais conhecido é a contagem de Pontos de Função (PFs), que busca medir as funcionalidades do sistema requisitadas e recebidas pelo usuário, de forma independente da tecnologia usada na implementação. Seu maior problema é que os dados necessários para a Análise de PFs são bastante imprecisos no início de um projeto e, portanto, gerentes de projeto são, muitas vezes, obrigados a produzir estimativas antes de um estudo mais aprofundado. Assim, os pontos de função devem ser recontados ao longo do processo (nas atividades de acompanhamento de projetos), para que ajustes de previsões possam ser realizados e controlados, fornecendo feedback para situações futuras.

Muitas organizações coletam dados para permitir a conversão de PFs em LOCs. Quando não há dados organizacionais para se fazer essa conversão, podem ser utilizados dados gerais reportados na literatura. Nesse trabalho, uma lista de valores de LOCs por PFs é fornecida para diversas linguagens, que, na média, apresenta valores como os mostrados na Tabela 3.2.

Tabela 3.2 – Relação Número Médio de LOC / PF para algumas Linguagens de

**Programação**

Linguagem de Programação	Número Médio de LOC/PF
C	162
C++	66
Java	63
SQL	40

Estimativas de Esforço:

Para a realização de estimativas de tempo e custo, é fundamental estimar, antes, o

esforço necessário para completar o projeto ou cada uma de suas atividades. Estimativas de esforço podem ser obtidas diretamente pelo julgamento de especialistas, tipicamente usando técnicas de decomposição, ou podem ser computadas a partir de dados de tamanho ou de dados históricos.

Quando as estimativas de esforço são feitas com base apenas no julgamento dos especialistas, uma tabela como a Tabela 3.1 pode ser utilizada, em que cada célula corresponde ao esforço necessário para efetuar uma atividade no escopo de um módulo específico. Uma tabela como essa pode ser produzida também com base em dados históricos de projetos similares já realizados na organização.

Quando estimativas de tamanho são usadas como base, deve-se considerar um fator de produtividade, indicando quanto em unidades de esforço é necessário para completar um projeto (ou módulo), descrito em unidades de tamanho. Assim, uma organização pode coletar dados de vários projetos e estabelecer, por exemplo, quantos em homens-hora (uma unidade de esforço) são necessários para desenvolver 1000 LOCs (KLOC) ou 1 PF (unidades de tamanho). Esses fatores de produtividade devem levar em

conta características dos projetos e da organização. Assim, pode ser útil ter vários fatores de produtividade, considerando classes de projetos específicas.

Assim como em outras situações, quando uma organização não tem ainda dados suficientes para definir seus próprios fatores de produtividade, modelos empíricos podem ser usados. Existem diversos modelos que derivam estimativas de esforço a partir de dados de LOC ou PFs. De maneira geral, todos eles têm a seguinte estrutura:

$$E = A + B * (T)^c$$

onde A, B e C são constantes derivadas empiricamente, E é o esforço em pessoas-mês e T é a estimativa de tamanho em LOCs ou PFs.

Por exemplo, o modelo proposto por Bailey-Basili estabelece a seguinte fórmula para se obter o esforço necessário em pessoas-mês para desenvolver um projeto, tomando por base o tamanho do mesmo em KLOC:

$$E = 5,5 + 0,73 * (KLOC)^{1,16}$$

Segundo os critérios citados por Pfleeger, esse é um dos modelos empíricos mais precisos. Contudo, deve-se observar que modelos empíricos diferentes conduzem a resultados muito diferentes, o que indica que esses modelos devem ser adaptados para as condições da organização. Uma forma de se fazer essa adaptação consiste em experimentar o modelo usando resultados de projetos já finalizados, comparar os valores obtidos com os dados reais e analisar a eficácia do modelo. Se a concordância dos resultados não for boa, as constantes do modelo devem ser recalculadas usando dados organizacionais.

#### **Alocação De Recursos:**

Estimar os recursos necessários para realizar o esforço de desenvolvimento é outra importante tarefa. Quando falamos em recursos, estamos englobando pessoas, hardware e software. No caso de software, devemos pensar em ferramentas de software, tais como ferramentas CASE ou software de infraestrutura (p.ex., um sistema operacional), bem como componentes de software a serem reutilizados no desenvolvimento, tais como bibliotecas de interface ou uma camada de persistência de dados.

Em todos os casos (recursos humanos, de hardware e de software), é necessário observar a disponibilidade do recurso. Assim, é importante definir a partir de que data o recurso será necessário, por quanto tempo ele será necessário e qual a quantidade de horas necessárias por dia nesse período, o que, para recursos humanos, convencionamos denominar dedicação. Observe que já entramos na estimativa de tempo. Assim, alocação de recursos e estimativa de tempo são atividades realizadas normalmente em paralelo.

No que se refere a recursos humanos, outros fatores têm de ser levados em conta. A competência para realizar a atividade para a qual está sendo alocado é fundamental. Assim, é preciso analisar com cuidado as competências dos membros da equipe para poder realizar a alocação de recursos. Outros fatores, como liderança, relacionamento inter-pessoal etc, importantes para a formação de equipes, são igualmente relevantes para a alocação de recursos humanos a atividades.

#### **Estimativa De Tempo:**

De posse das estimativas de esforço e realizando em paralelo a alocação de recursos, é possível estimar o tempo de cada atividade e, por conseguinte, do projeto. Se a estimativa de esforço tiver sido realizada para o projeto como um todo, então ela deverá ser distribuída pelas atividades do projeto. Novamente, dados históricos de projetos já concluídos na organização são uma boa base para se fazer essa distribuição.

No entanto, muitas vezes, uma organização não tem ainda esses dados disponíveis. Embora as características do projeto sejam determinantes para a distribuição do esforço, uma diretriz inicial útil consiste em considerar a distribuição mostrada na Tabela 3.3.

Tabela 3.3 – Distribuição de Esforço pelas Principais Atividades do Processo de Software.

Planejamento	Especificação e Análise de Requisitos	Projeto	Implementação	Teste e Entrega
Até 3%	De 10 a 25%	De 20 a 25%	De 15 a 20%	De 30 a 40%

De posse da distribuição de esforço por atividade e realizando paralelamente a alocação de recursos, pode-se criar uma rede de tarefas com o esforço associado a cada uma das atividades. A partir dessa rede, pode-se estabelecer qual é o caminho crítico do projeto, isto é, qual o conjunto de atividades que determina a duração do projeto. Um atraso em uma dessas atividades provocará atraso no projeto como um todo.

Finalmente, a partir da rede de tarefas, deve-se elaborar um Gráfico de Tempo (ou Gráfico de Gantt), estabelecendo o cronograma do projeto. Gráficos de Tempo podem ser elaborados para o projeto como um todo (cronograma do projeto), para um conjunto de atividades, para um módulo específico ou mesmo para um membro da equipe do projeto.

Estimativa de Custo:

De posse das demais estimativas, é possível estimar os custos do projeto. De maneira geral, os seguintes itens devem ser considerados nas estimativas de custos:

Custos relativos ao esforço empregado pelos membros da equipe no projeto;

Custos de hardware e software (incluindo manutenção);

Outros custos relacionados ao projeto, tais como custos de viagens e treinamentos realizados no âmbito do projeto;

Despesas gerais, incluindo gastos com água, luz, telefone, pessoal de apoio administrativo, pessoal de suporte etc.

Para a maioria dos projetos, o custo dominante é o que se refere ao esforço empregado, juntamente com as despesas gerais. Sommerville sugere que, de modo geral, os custos relacionados com as despesas gerais correspondem a um valor equivalente aos custos relativos ao esforço empregado pelos membros da equipe no projeto. Assim, para efeitos de estimativas de custos, pode-se considerar esses dois itens como sendo um único, computado em dobro.

Custos de hardware e software, ainda que menos influentes, não devem ser desconsiderados, sob pena de provocarem prejuízos para o projeto. Uma forma de tratar esses custos é considerar a depreciação com base na vida útil do equipamento ou da versão do software utilizada.

Quando o custo do projeto estiver sendo calculado como parte de uma proposta para o cliente, então será preciso definir o preço cotado. Uma abordagem para definição do preço pode ser considerá-lo como o custo total do projeto mais o lucro. Entretanto, a relação entre o custo do projeto e o preço cotado para o cliente, normalmente, não é tão simples assim.

### Gerência de Riscos

Uma importante tarefa da gerência de projetos é prever os riscos que podem prejudicar o bom andamento do projeto e definir ações a serem tomadas para evitar sua ocorrência ou, quando não for possível evitar a ocorrência, para diminuir seus impactos.

Um risco é qualquer condição, evento ou problema cuja ocorrência não é certa, mas que pode afetar negativamente o projeto, caso ocorra. Assim, os riscos envolvem duas características principais:

Incerteza – um risco pode ou não acontecer, isto é, não existe nenhum risco 100% provável;

Perda – se o risco se tornar realidade, consequências não desejadas ou perdas acontecerão.

Desta forma, é de suma importância que riscos sejam identificados durante um projeto de software, para que ações possam ser planejadas e utilizadas para evitar que um risco se torne real, ou para



minimizar seus impactos, caso ele ocorra. Esse é o objetivo da Gerência de Riscos, cujo processo envolve as seguintes atividades:

**Identificação de riscos:** visa identificar possíveis ameaças (riscos) para o projeto, antecipando o que pode dar errado;

**Análise de riscos:** trata de analisar os riscos identificados, estimando sua probabilidade e impacto (grau de exposição ao risco);

**Avaliação de riscos:** busca priorizar os riscos e estabelecer um ponto de corte, indicando quais riscos serão gerenciados e quais não serão;

**Planejamento de ações:** trata do planejamento das ações a serem tomadas para evitar (ações de mitigação) que um risco ocorra ou para definir o que fazer quando um risco se tornar realidade (ações de contingência);

**Elaboração do Plano de Riscos:** todos os aspectos envolvidos na gerência de riscos devem ser documentados em um plano de riscos, indicando os riscos que compõem o perfil de riscos do projeto, as avaliações dos riscos, a definição dos riscos a serem gerenciados e, para esses, as ações para evitá-los ou para minimizar seus impactos, caso venham a ocorrer.

**Monitoramento de riscos:** à medida que o projeto progride, os riscos têm de ser monitorados para se verificar se os mesmos estão se tornando realidade ou não. Novos riscos podem ser identificados, o grau de exposição de um risco pode mudar e ações podem ter de ser tomadas. Essa atividade é realizada durante o acompanhamento do progresso do projeto.

Na identificação de riscos, trabalhar com uma série de riscos aleatórios pode ser um fator complicador, principalmente em grandes projetos, em que o número de riscos é relativamente grande. Assim, a classificação de riscos em categorias de risco, definindo tipos básicos de riscos, é importante para guiar a realização dessa atividade. Cada organização deve ter seu próprio conjunto de categorias de riscos. Para efeito de exemplo, podem ser consideradas categorias tais como: tecnologia, pessoal, legal, organizacional, de negócio etc.

Uma vez identificados os riscos, deve ser feita uma análise dos mesmos à luz de suas duas principais variáveis: a probabilidade do risco se tornar real e o impacto do mesmo, caso ocorra. Na análise de riscos, o gerente de projeto deve executar quatro atividades básicas:

estabelecer uma escala que reflita a probabilidade de um risco, (ii) avaliar as consequências dos riscos, (iii) estimar o impacto do risco no projeto e no produto e (iv) calcular o grau de exposição do risco, que é uma medida casando probabilidade e impacto.

De maneira geral, escalas para probabilidades e impactos são definidas de forma qualitativa, tais como: probabilidade - alta, média ou baixa, e impacto - baixo, médio, alto ou muito alto. Isso facilita a análise dos riscos, mas, por outro lado, pode dificultar a avaliação. Assim, a definição de medidas quantitativas para o risco pode ser importante, pois tende a diminuir a subjetividade na avaliação. Jalote propõe os valores quantitativos mostrados nas tabelas 3.4 e 3.5 para as escalas acima.

Tabela 3.4 - Categorias de Probabilidade

<b>Probabilidade</b>	<b>Faixa de Valores</b>
Baixa	até 30%
Média	30 a 70%
Alta	acima de 70%

Tabela 3.5 - Categorias de Impacto

<b>Impacto</b>	<b>Faixa de Valores</b>
Baixo	de 0 a 3
Médio	de 3 a 7
Alto	de 7 a 9
Muito Alto	de 9 a 10

Usando valores quantitativos para expressar probabilidade e impacto, é possível obter um valor numérico para o grau de exposição ao risco, dado por:  $E(r) = P(r) * I(r)$ , onde  $E(r)$  é o grau de exposição associado ao risco  $r$  e  $P(r)$  e  $I(r)$  correspondem, respectivamente, aos valores numéricos de probabilidade e impacto do risco  $r$ .

De posse do grau de exposição de cada um dos riscos que compõem o perfil de riscos do projeto, pode-se passar à avaliação dos mesmos. Uma tabela ordenada pelo grau de exposição pode ser montada e uma linha de corte nessa tabela estabelecida, indicando quais riscos serão tratados e quais serão desprezados.

Para os riscos a serem gerenciados, devem ser definidos planos de ação, estabelecendo ações a serem adotadas para, em primeiro lugar, evitar que os riscos ocorram (ações de mitigação) ou, caso isso não seja possível, ações para tratar e minimizar seus impactos (ações de contingência). Esse é o propósito da atividade de planejamento das ações.

Ao longo de todo o processo da gerência de riscos, as decisões envolvidas devem ser documentadas no plano de riscos. Esse plano servirá de base para a atividade de monitoramento dos riscos, quando os riscos têm de ser monitorados para se verificar se os mesmos estão se tornando realidade ou não. Caso estejam se tornando (ou já sejam) uma realidade, deve-se informar que ações foram tomadas. No caso do risco se concretizar, deve-se informar, também, quais as consequências da sua ocorrência.

### **Elaboração do Plano de Projeto**

Todas as atividades realizadas no contexto da gerência de projeto devem ser documentadas em um Plano de Projeto. Cada organização deve estabelecer um modelo ou padrão para a elaboração desse documento, de modo que todos os projetos da organização contenham as informações consideradas relevantes.

### **Gerência da Qualidade**

#### **Documentação**

A documentação produzida em um projeto de software é de suma importância para se gerenciar a qualidade, tanto do produto sendo produzido, quanto do processo usado para seu desenvolvimento.

No desenvolvimento de software, são produzidos diversos documentos, dentre eles, documentos descrevendo processos (plano de projeto, plano de qualidade etc.), registrando requisitos e modelos do sistema (documentos de especificação de requisitos, análise e projeto) e apoiando o uso do sistema gerado (manual do usuário, ajuda on-line, tutoriais etc.).

Uma documentação de qualidade propicia uma maior organização durante o desenvolvimento de um sistema, facilitando modificações e futuras manutenções no mesmo. Além disso, reduz o impacto da perda de membros da equipe, reduz o tempo de desenvolvimento de fases posteriores, reduz o tempo de manutenção e contribui para redução de erros, aumentando assim, a qualidade do processo e do produto gerado. Dessa forma, a criação da documentação é tão importante quanto a criação do software em si.

Há, portanto, a necessidade de se definir um processo para controlar a documentação de uma organização, dito processo de documentação, incluindo atividades de planejamento, análise, aprovação ou reprovação, identificação de alterações, situação da revisão atual, disponibilidade das versões pertinentes de documentos aplicáveis, dentre outras. Algumas dessas atividades estão relacionadas com o

controle e a garantia da qualidade de software, outras com a gerência da configuração do software, conforme discutido a seguir.

É importante notar que o planejamento da documentação tem uma estreita relação com o processo de software definido para o projeto. Ou seja, os documentos a serem gerenciados são aqueles previstos como saídas das atividades do processo. Assim, tendo sido definido o processo do projeto, o planejamento da sua documentação consiste apenas em selecionar quais artefatos, dentre os muitos produzidos ao longo do processo, serão efetivamente submetidos à gerência de configuração de software e ao controle e garantia da qualidade.

### **Controle e Garantia da Qualidade**

Durante o processo de desenvolvimento de software, ocorrem enganos, interpretações errôneas e outras faltas (defeitos ou erros), principalmente provocados por problemas na comunicação e transformação da informação, que podem resultar em mau funcionamento do sistema produzido. Assim, é muito importante detectar esses defeitos o quanto antes, preferencialmente na atividade em que foram cometidos, como forma de diminuir retrabalho e, por conseguinte, custos de alterações. As atividades que se preocupam com essa questão são coletivamente denominadas atividades de garantia da qualidade de software e devem ser realizadas ao longo de todo o processo de desenvolvimento de software [5].

Dentre as atividades de controle e garantia da qualidade estão as atividades de Verificação, Validação e Testes (VV&T). O objetivo da verificação é assegurar que o software esteja sendo construído de forma correta. Deve-se verificar se os artefatos produzidos atendem aos requisitos estabelecidos e se os padrões organizacionais (de produto e processo) foram consistentemente aplicados. Por outro lado, o objetivo da validação é assegurar que o software que está sendo desenvolvido é o software correto, ou seja, se os requisitos e o software dele derivado atendem ao uso específico proposto. Por fim, os testes de software são atividades de validação e verificação que consistem da análise dinâmica do mesmo, isto é, envolvem a execução do produto de software.

Uma vez que as atividades de VV&T são tão importantes, elas devem ser cuidadosamente planejadas, dando origem a um Plano de Garantia da Qualidade.

Os artefatos que compõem a documentação do projeto são as entradas (insumos) para as atividades de garantia da qualidade, quando os mesmos são verificados quanto à aderência em relação aos padrões de documentação da organização e validados em relação aos seus propósitos e aos requisitos que se propõem a atender. Assim, uma questão imprescindível para a gerência da qualidade é a definição de padrões organizacionais.

### **Padrões Organizacionais**

Uma vez que a gerência da qualidade envolve tanto a qualidade do processo quanto a qualidade do produto, devem ser estabelecidos padrões organizacionais de produto e de processo. Os padrões de processo são os ditos processos padrão ou processos padrão especializados, discutidos no capítulo 2.

Os padrões de produto, por sua vez, são padrões que se aplicam a artefatos produzidos ao longo do processo de software. Podem ser, dentre outros, modelos de documentos, roteiros e normas, dependendo do artefato a que se aplicam. Tipicamente, para documentos, modelos de documentos e roteiros são providos.

Um modelo de documento define a estrutura (seções, sub-seções, informações de cabeçalho e rodapé de página etc), o estilo (tamanho e tipos de fonte, cores etc) e o conteúdo esperado para documentos de um tipo específico.

Documentos tais como Plano de Projeto, Especificação de Requisitos e Especificação de Projeto devem ter modelos de documentos específicos associados. Documentos padronizados são importantes, pois facilitam a leitura e a compreensão, uma vez que os profissionais envolvidos estão familiarizados com seu formato.

Quando não é possível ou desejável estabelecer uma estrutura rígida como um modelo de documento, roteiros dando diretrizes gerais para a elaboração de um artefato devem ser providos. Em situações onde não é possível definir uma estrutura, tal como no código-fonte, normas devem ser providas. Assim,

tomando o exemplo do código-fonte, é extremamente pertinente a definição de um padrão de codificação, indicando nomes de variáveis válidos, estilos de indentação, regras para comentários etc.

Padrões organizacionais, sejam de processo sejam de produto, são muito importantes, pois eles fornecem um meio de capturar as melhores práticas de uma organização e facilitam a realização de atividades de avaliação da qualidade, que podem ser dirigidas pela avaliação da conformidade em relação ao padrão. Além disso, sendo organizacionais, todos os membros da organização tendem a estar familiarizados com os mesmos, facilitando a manutenção dos artefatos ou a substituição de pessoas no decorrer do projeto. Para aqueles artefatos tidos como os mais importantes no planejamento da documentação, é saudável que haja um padrão organizacional associado.

Dada a importância de padrões organizacionais, eles devem ser elaborados com bastante cuidado. Uma boa prática, conforme já sugerido para a definição de processos padrão, consiste em usar como base padrões gerais propostos por instituições nacionais ou internacionais voltadas para a área de qualidade de software, tal como a ISO.

### Revisões

Para se controlar a qualidade em um projeto de software, uma abordagem bastante usada consiste em se realizar revisões. Nas revisões, processos, documentos e outros artefatos são revisados por um grupo de pessoas, com o objetivo de verificar se os mesmos estão em conformidade com os padrões organizacionais estabelecidos e se o propósito de cada um deles está sendo atingido, incluindo o atendimento a requisitos do cliente e usuários. Assim, o objetivo de uma revisão é detectar erros e inconsistências em artefatos e processos, sejam eles relacionados à forma, sejam em relação ao conteúdo, e apontá-los aos responsáveis pela sua elaboração.

processo de revisão começa com o planejamento da revisão, quando uma equipe de revisão é formada, tendo à frente um líder. A equipe de revisão deve incluir os membros da equipe que possam efetivamente úteis para atingir o objetivo da revisão. Muitas vezes, a pessoa responsável pela elaboração do artefato a ser revisado integra a equipe de revisão.

Dando início ao processo de revisão propriamente dito, normalmente, o autor do artefato apresenta o mesmo e descreve a perspectiva utilizada para a sua construção. Além disso, o propósito da revisão deve ser previamente informado e o material a ser revisado deve ser entregue com antecedência para que cada membro da equipe de revisão possa avaliá-lo. Uma vez que todos estejam preparados, uma reunião é convocada pelo líder. Essa reunião deverá ser relativamente breve (duas horas, no máximo), uma vez que todos já estão preparados para a mesma. Durante a reunião, o líder orientará o processo de revisão, passando por todos os aspectos relevantes a serem revistos. Todas as considerações dos demais membros da equipe de revisão devem ser discutidas e as decisões registradas, dando origem a uma ata de reunião de revisão, contendo uma lista de defeitos encontrados.

### Gerência de Configuração

Durante o processo de desenvolvimento de software, vários artefatos são produzidos e alterados constantemente, evoluindo até que seus propósitos fundamentais sejam atendidos. Ferramentas de software, tais como compiladores e editores de texto, e processos também podem ser substituídos por versões mais recentes ou mesmo por outras, no caso de ferramentas. Porém, caso essas mudanças não sejam devidamente documentadas e comunicadas, poderão acarretar diversos problemas, tais como: dois ou mais desenvolvedores podem estar alterando um mesmo artefato ao mesmo tempo; não se saber qual a versão mais atual de um artefato; não se refletir alterações nos artefatos impactados por um artefato em alteração. Esses problemas podem gerar vários transtornos como incompatibilidade entre os grupos de desenvolvimento, inconsistências, retrabalho, atraso na entrega e insatisfação do cliente.

Assim, para que esses transtornos sejam evitados, é de suma importância o acompanhamento e o controle de artefatos, processos e ferramentas, através de um processo de gerência de configuração de software, durante todo o ciclo de vida do software.

A Gerência de Configuração de Software (GCS) visa estabelecer e manter a integridade dos itens de software ao longo de todo o ciclo de vida do software, garantindo a completeza, a consistência e a correção de tais itens, e controlando o armazenamento, a manipulação e a distribuição dos mesmos.

Para tal, tem de identificar e documentar os produtos de trabalho que podem ser modificados, estabelecer as relações entre eles e os mecanismos para administrar suas diferentes versões, controlar modificações e permitir auditoria e a elaboração de relatórios sobre o estado de configuração.

Pelos objetivos da GCS, pode-se notar que ela está diretamente relacionada com as atividades de garantia da qualidade de software.

As atividades da GCS podem ser resumidas em:

**Planejamento da GCS:** Um plano deve ser elaborado descrevendo as atividades da gerência de configuração, procedimentos e responsáveis pela execução dessas atividades.

**Identificação da Configuração:** refere-se à identificação dos itens de software e suas versões a serem controladas, estabelecendo linhas básicas.

**Controle de Versão:** combina procedimentos e ferramentas para administrar diferentes versões dos itens de configuração criados durante o processo de software.

**Controle de Modificação:** combina procedimentos humanos e ferramentas automatizadas para controlar as alterações feitas em itens de software. Para tal, o seguinte processo é normalmente realizado: solicitação de mudança, aprovação ou rejeição da solicitação, registro de retirada para alteração (check-out), análise, avaliação e realização das alterações, revisão e registro da realização das alterações (check-in).

**Auditoria de Configuração:** visa avaliar um item de configuração quanto a características não consideradas nas revisões, tal como se os itens relacionados aos solicitados foram devidamente atualizados.

**Relato da situação da configuração:** refere-se à preparação de relatórios que mostrem a situação e o histórico dos itens de software controlados. Tais relatórios podem incluir, dentre outros, o número de alterações nos itens, as últimas versões dos mesmos e identificadores de liberação.

## O Processo de GCS

Primeiro passo do processo de gerência de configuração de software é a confecção de um plano de gerência de configuração, que inicia com a identificação dos itens que serão colocados sob gerência de configuração, chamados itens de configuração. Os itens mais relevantes para serem submetidos à gerência de configuração são aqueles mais usados durante o ciclo de vida, os mais genéricos, os mais importantes para segurança, os projetados para reutilização e os que podem ser modificados por vários desenvolvedores ao mesmo tempo. Os itens não colocados sob gerência de configuração podem ser alterados livremente.

Após a seleção dos itens, deve-se descrever como eles se relacionam. Isso será muito importante para as futuras manutenções, pois permite identificar de maneira eficaz os itens afetados em decorrência de uma alteração. Além disso, deve-se criar um esquema de identificação dos itens de configuração, com atribuição de nomes exclusivos, para que seja possível estabelecer a evolução de cada versão dos itens.

Após a identificação dos itens de configuração, devem ser planejadas as linhas-base dentro do ciclo de vida do projeto. Uma linha-base (ou baseline) é uma versão estável de um sistema contendo todos os componentes que constituem este sistema em um determinado momento. Nos pontos estabelecidos pelas linhas-base, os itens de configuração devem ser identificados, analisados, corrigidos, aprovados e armazenados em um local sob controle de acesso, denominado repositório central, base de dados de projeto ou biblioteca de projeto. Assim, quaisquer alterações nos itens daí em diante só poderão ser realizadas através de procedimentos formais de controle de modificação.

Passo seguinte do processo de GCS é o controle de versão, que combina procedimentos e ferramentas para identificar, armazenar e administrar diferentes versões dos itens de configuração que são criados durante o processo de software. A idéia é que a cada modificação que ocorra em um item de configuração, uma nova versão ou variante seja criada. Versões de um item são geradas pelas diversas alterações, enquanto variantes são as diferentes formas de um item, que existem simultaneamente, e atendem a requisitos similares.

Talvez a mais importante atividade do processo de GCS é o controle de alterações, que combina procedimentos humanos e ferramentas automatizadas para o controle das alterações realizadas nos itens de configuração. Assim que uma alteração é solicitada, o impacto em outros itens de configuração e o custo para a modificação devem ser avaliados. Um responsável deve decidir se a alteração poderá ou não ser realizada.

Caso a alteração seja liberada, pessoas são indicadas para sua execução. Assim que não houver ninguém utilizando os itens de configuração envolvidos, cópias deles são retiradas do repositório central e colocadas em uma área de trabalho do desenvolvedor, através de um procedimento denominado check-out. A partir deste momento, nenhum outro desenvolvedor poderá alterar esses itens. Os desenvolvedores designados fazem as alterações necessárias e, assim que essas forem concluídas, os itens são submetidos a uma revisão. Se as alterações forem aprovadas, os itens são devolvidos ao repositório central, estabelecendo uma nova linha-base, em um procedimento chamado check-in.

Porém, mesmo com os mecanismos de controle mais bem-sucedidos, não é possível garantir que as modificações foram corretamente implementadas. Assim, revisões e auditorias de configuração de software são necessárias no processo de GCS. Essas atividades de garantia da qualidade tentam descobrir omissões ou erros na configuração e se os procedimentos, padrões, regulamentações ou guias foram devidamente aplicados no processo e no produto.

Enfim, o último passo do processo de GCS é a preparação de relatórios, que é uma tarefa que tem como objetivo relatar a todas as pessoas envolvidas no desenvolvimento e manutenção do software as seguintes questões: (i) O que aconteceu? (ii) Quem fez? (iii) Quando aconteceu? (iv) O que mais será afetado? O acesso rápido às informações agiliza o processo de desenvolvimento e melhora a comunicação entre as pessoas, evitando, assim, muitos problemas de alterações do mesmo item de configuração, com intenções diferentes e, às vezes, conflitantes.

### **Métricas e Medição**

Para poder controlar a qualidade, medir é muito importante. Se não é possível medir, expressando em números, apenas uma análise qualitativa (e, portanto, subjetiva) pode ser feita, o que, na maioria das vezes, é insuficiente. Com medições, as tendências (boas ou más) podem ser detectadas, melhores estimativas podem ser feitas e melhorias reais podem ser conseguidas.

Quando se trata de avaliação quantitativa, quatro conceitos, muitas vezes usados equivocadamente com o mesmo sentido, são importantes: medida, medição, métrica e indicador. Uma medida fornece uma indicação quantitativa da extensão, quantidade, dimensão, capacidade ou tamanho de um atributo de um produto ou de um processo. Quando os dados de um único ponto são coletados, uma medida é estabelecida (p.ex., quantidade de erros descobertos em uma revisão).

Medição é o ato de medir, isto é, de determinar uma medida. Já uma métrica procura relacionar medidas individuais com o objetivo de se ter uma idéia da eficácia do processo, do projeto ou do produto sendo medido. Por fim, desenvolve-se métricas para se obter indicadores, isto é, para se ter uma compreensão do processo, produto ou projeto sendo medido.

Seja o seguinte exemplo: deseja-se saber se uma pessoa está com seu peso ideal ou não. Para tal, duas medidas são importantes: altura (H) e peso (P). Ao medir essas dimensões, está-se efetuando uma medição. A métrica "índice de massa corporal (IMC)" é calculada segundo a seguinte fórmula:  $IMC = P / H^2$ . A partir dessa métrica, a Organização Mundial de Saúde estabeleceu indicadores que apontam se um adulto está acima do peso, se está obeso ou abaixo do peso ideal considerado saudável, conforme abaixo:

Se  $IMC < 18,5$ , então o indivíduo está abaixo do peso ideal considerado saudável; Se  $18,5 \leq IMC < 25$ , então o indivíduo está com o peso normal;

Se  $25 \leq IMC \leq 30$ , então o indivíduo está acima do peso; Se  $IMC > 30$ , então o indivíduo está obeso.

Realizando medições, uma pessoa pode obter suas medidas para altura e peso. A partir delas, pode calcular a métrica IMC e ter um indicador se está abaixo do peso, no peso ideal, acima do peso ou obeso. Conhecendo esse indicador, a pessoa pode ajustar seu processo de alimentação, obtendo melhorias reais para sua saúde. Uma vez que a medição de software se preocupa em obter valores numéricos (medidas) para alguns atributos de um produto ou de um processo, uma questão importante

passa a ser: Que atributos medir? Modelo de qualidade definido na norma ISO/IEC 9126-1 trata dessa questão, estando sub-dividido em dois modelos: o modelo de qualidade para características externas e internas e o modelo de qualidade para qualidade em uso.

O modelo de qualidade para características externas e internas classifica os atributos de qualidade de software em seis características (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade) que são, por sua vez, desdobradas em sub-características. As sub-características podem ser desdobradas em mais níveis, até se ter sub-características diretamente mensuráveis, para as quais métricas são aplicadas. As normas ISO/IEC 9126-2 e 9126-3 apresentam, respectivamente, métricas externas e internas.

Esse modelo de qualidade que preconiza a análise de características de qualidade a partir de suas sub-características de forma recursiva até que se tenham métricas para coletar dados é aplicável não somente a produto, mas também para avaliar a qualidade de processos de software. Assim, de maneira geral, na avaliação quantitativa da qualidade, é necessário:

Definir características de qualidade relevantes para avaliar a qualidade do objeto em questão (produto ou processo).

Para cada característica de qualidade selecionada, definir sub-características de qualidade relevantes que tenham influência sobre a mesma, estabelecendo um modelo ou fórmula de computar a característica a partir das sub-características. Fórmulas baseadas em peso são bastante utilizadas:  $cq = p_1 * scq_1 + ... + p_n * scq_n$ .

Usar procedimento análogo ao anterior para as sub-características não passíveis de mensuração direta.

Para as sub-características diretamente mensuráveis, selecionar métricas, coletar medidas e computar as métricas segundo a fórmula ou modelo estabelecido.

Fazer o caminho inverso, agora computando sub-características não diretamente mensuráveis, até se chegar a um valor para a característica de qualidade.

Concluído o processo de medição, deve-se comparar os valores obtidos com padrões estabelecidos para a organização, de modo a se obter os indicadores da qualidade. A partir dos indicadores, ações devem ser tomadas visando à melhoria.

Vale destacar que, especialmente no caso da avaliação da qualidade de software, métricas relacionadas a defeitos são bastante úteis, tal como (número de erros / ponto de função).

único modo racional de melhorar um processo é medir atributos específicos, obter um conjunto de métricas significativas baseadas nesses atributos e usar os valores das métricas para fornecer indicadores que conduzirão um processo de melhoria.

#### Levantamento e Análise De Requisitos

Uma das principais medidas do sucesso de um software é o grau no qual ele atende aos objetivos e requisitos para os quais foi construído. De forma geral, a Engenharia de Requisitos de Software é o processo de identificar todos os envolvidos, descobrir seus objetivos e necessidades e documentá-los de forma apropriada para análise, comunicação e posterior implementação.

#### Requisito e Tipos de Requisitos

As descrições das funções que um sistema deve incorporar e das restrições que devem ser satisfeitas são os requisitos para o sistema. Isto é, os requisitos de um sistema definem o que o sistema deve fazer e as circunstâncias sob as quais deve operar. Em outras palavras, os requisitos definem os serviços que o sistema deve fornecer e dispõem sobre as restrições à operação do mesmo.

Requisitos são, normalmente, classificados em requisitos funcionais e não funcionais. Requisitos funcionais, como o próprio nome indica, apontam as funções que o sistema deve fornecer e como o sistema deve se comportar em determinadas situações. Já os requisitos não funcionais descrevem restri-

ções sobre as funções oferecidas, tais como restrições de tempo, de uso de recursos etc. Alguns requisitos não funcionais dizem respeito ao sistema como um todo e não a funcionalidade específica. Dependendo da natureza, os requisitos não funcionais podem ser classificados de diferentes maneiras, tais como requisitos de desempenho, requisitos de portabilidade, requisitos legais, requisitos de conformidade etc.

### **O Processo da Engenharia de Requisitos**

Processo de descobrir, analisar, documentar e verificar / validar requisitos é chamado de processo de engenharia de requisitos. Os processos de engenharia de requisitos variam muito de uma organização para outra, mas, de maneira geral, a maioria dos processos de Engenharia de Requisitos é composta das seguintes atividades:

**Elicitação / Descoberta / Levantamento de Requisitos:** Nesta fase, os usuários, clientes e especialistas de domínio são identificados e trabalham junto com os engenheiros de requisitos para descobrir, articular e entender a organização como um todo, o domínio da aplicação, os processos de negócio específicos, as necessidades que o software deve atender e os problemas e deficiências dos sistemas atuais. Os diferentes pontos de vista dos participantes do processo, bem como as oportunidades e restrições existentes, os problemas a serem resolvidos, o desempenho requerido e as restrições também devem ser levantados.

**Análise de Requisitos:** visa a estabelecer um conjunto acordado de requisitos consistentes e sem ambiguidades, que possa ser usado como base para o desenvolvimento do software. Para tal, diversos tipos de modelos são construídos. Geralmente, a análise de requisitos inclui também a negociação para resolver conflitos detectados.

**Documentação de Requisitos:** é a atividade de representar os resultados da Engenharia de Requisitos em um documento, contendo os requisitos do software.

**Verificação e Validação de Requisitos:** A verificação de requisitos avalia se o documento de Especificação de Requisitos está sendo construído de forma correta, de acordo com padrões previamente definidos, sem conter requisitos ambíguos, incompletos ou, ainda, requisitos incoerentes ou impossíveis de serem testados. Já a validação diz respeito a avaliar se esse documento está correto, ou seja, se os requisitos e modelos documentados atendem às reais necessidades e requisitos dos usuários / cliente.

**Gerência de Requisitos:** se preocupa em gerenciar as mudanças nos requisitos já acordados, manter uma trilha dessas mudanças, gerenciar os relacionamentos entre os requisitos e as dependências entre o documento de requisitos e os demais artefatos produzidos no processo de software, de forma a garantir que mudanças nos requisitos sejam feitas de maneira controlada e documentada.

É possível notar que, das cinco atividades do processo de Engenharia de Requisitos anteriormente listadas, as três últimas são, na realidade, instanciações para a Engenharia de Requisitos de atividades genéricas já discutidas no capítulo 4, a saber: documentação, garantia da qualidade e gerência de configuração. Assim sendo, neste capítulo, somente as duas primeiras atividades (Levantamento e Análise de Requisitos) são discutidas.

### **Levantamento de Requisitos**

Levantamento de requisitos é uma atividade complexa que não se resume somente a perguntar às pessoas o que elas desejam e também não é uma simples transferência de conhecimento. Várias técnicas de levantamento de requisitos são normalmente empregadas pelos engenheiros de requisitos (ou analistas de sistemas), dentre elas entrevistas, questionários, prototipação, investigação de documentos, observação, dinâmicas de grupo etc.

### **Análise de Requisitos**

A análise de requisitos (muitas vezes chamada análise de sistemas) é, em última instância, uma atividade de construção de modelos. Um modelo é uma representação de alguma coisa do mundo real, uma abstração da realidade, e, portanto, representa uma seleção de características do mundo real relevantes para o propósito do sistema em questão.



Modelos são fundamentais no desenvolvimento de sistemas. Tipicamente eles são construídos para:

Possibilitar o estudo do comportamento do sistema;

Facilitar a comunicação entre os componentes da equipe de desenvolvimento e clientes e usuários;

Possibilitar a discussão de correções e modificações com o usuário;

Formar a documentação do sistema.

Um modelo enfatiza um conjunto de características da realidade, que corresponde à dimensão do modelo. Além da dimensão que um modelo enfatiza, modelos possuem níveis de abstração. O nível de abstração de um modelo diz respeito ao grau de detalhamento com que as características do sistema são representadas. Em cada nível há uma ênfase seletiva nos detalhes representados. No caso do desenvolvimento de sistemas, geralmente, são considerados três níveis:

**Conceitual:** considera características do sistema independentes do ambiente computacional (hardware e software) no qual o sistema será implementado. Essas características são dependentes unicamente das necessidades do usuário. Modelos conceituais são construídos na atividade de análise de requisitos.

**Lógico:** características dependentes de um determinado tipo de sistema computacional. Essas características são, contudo, independentes de produtos específicos. Tais modelos são típicos da fase de projeto.

**Físico:** características dependentes de um sistema computacional específico, isto é, uma linguagem e um compilador específico, um sistema gerenciador de bancos de dados específico, o hardware de um determinado fabricante etc. Tais modelos podem ser construídos tanto na fase de projeto quanto na fase de implementação.

Conforme apontado acima, nas primeiras etapas do processo de desenvolvimento (levantamento de requisitos e análise), o engenheiro de software representa o sistema através de modelos conceituais. Nas etapas posteriores, as características lógicas e físicas são representadas em novos modelos.

Para a realização da atividade de análise, uma escolha deve ser feita: o paradigma de desenvolvimento. Paradigmas de desenvolvimento estabelecem a forma de se ver o mundo e, portanto, definem as características básicas dos modelos a serem construídos. Por exemplo, o paradigma orientado a objetos parte do pressuposto que o mundo é povoado por objetos, ou seja, a abstração básica para se representar as coisas do mundo são os objetos. Já o paradigma estruturado adota uma visão de desenvolvimento baseada em um modelo entrada- processamento-saída. No paradigma estruturado, os dados são considerados separadamente das funções que os transformam e a decomposição funcional é usada intensamente.

Neste texto, as atividades de Levantamento e Análise de Requisitos são discutidas à luz do paradigma estruturado, mais especificamente utilizando os conceitos da Análise Essencial de Sistemas.

### **Análise Essencial de Sistemas**

Método da Análise Essencial de Sistemas preconiza que, de uma forma geral, um sistema deve ser modelado através de três dimensões:

**Dados:** diz respeito aos aspectos estáticos e estruturais do sistema;

**Controle:** leva em conta aspectos temporais e comportamentais do sistema;

**Funções:** considera a transformação de valores.

Em relação ao grau de abstração, a Análise Essencial considera dois níveis: o nível essencial e o nível de implementação, representados, respectivamente, pelos seguintes modelos:

**Modelo Essencial:** representa o sistema num grau de abstração completamente independente de restrições tecnológicas.

Modelo de Implementação: passa a considerar as restrições tecnológicas impostas pela plataforma de hardware e software a ser utilizada para implementar o sistema.

Podemos perceber que o modelo de implementação não corresponde a um modelo de análise propriamente dito, uma vez que considera aspectos de implementação, característica marcante da fase de projeto. De fato, na abordagem da Análise Essencial, este modelo corresponde a uma espécie de zona nebulosa entre as fases de análise e de projeto. Por considerarmos que um modelo considerando aspectos da plataforma de implementação é melhor caracterizado na fase de projeto, neste texto, não trataremos do modelo de implementação.

A Análise Essencial é, de fato, uma evolução da Análise Estruturada de Sistemas, mais especificamente da Análise Estruturada Moderna [6]. Os conceitos introduzidos pela Análise Essencial endereçavam inicialmente as duas principais dificuldades que os analistas enfrentavam com a aplicação da Análise Estruturada: a distinção entre requisitos lógicos e físicos, e a ausência de uma abordagem para dividir o sistema em partes tão independentes quanto possível, de modo a facilitar o processo de análise.

Durante muito tempo, no paradigma estruturado, houve grandes debates entre os profissionais de desenvolvimento de sistemas sobre por qual perspectiva se deveria começar a especificação de um sistema: pelos dados ou pelas funções? Os argumentos, igualmente válidos, exploravam considerações como:

Dados são mais estáveis que funções.

Sem um entendimento das funções a serem desempenhadas pelo sistema, como definir o escopo e os dados necessários?

A Análise Essencial procurou estabelecer um novo ponto de partida para a especificação de um sistema: a identificação dos eventos que o afetam.

Conforme apontado anteriormente, um problema bastante relevante na especificação consiste em decidir como efetuar uma boa divisão do problema. A Análise Estruturada propõe que essa divisão seja obtida por meio de uma abordagem top-down. Embora essa seja uma boa maneira de se atacar um problema complexo – partir da visão geral e ir descendo, em uma visão hierárquica, a níveis de detalhes cada vez maiores – na prática, essa abordagem não se mostrou eficiente como estratégia para a decomposição de sistemas. A Análise Essencial propõe uma outra forma de particionamento, a qual é baseada nos eventos, e que mostrou ser mais efetiva do que a abordagem top-down, pois torna mais fácil a identificação das funções e entidades que compõem o sistema.

A Análise Essencial de Sistemas, através da técnica de particionamento por eventos, oferece uma boa estratégia para modelar o comportamento do sistema, visando satisfazer os requisitos do usuário, presumindo-se que dispomos de tecnologia perfeita e que ela pode ser obtida a custo zero.

A Análise Essencial preserva todos os modelos da Análise Estruturada. De fato, embora diferentes, a melhor maneira de encarar a Análise Essencial é considerá-la uma evolução da Análise Estruturada. Isso porque a Análise Essencial introduz novos conceitos e abordagens, dentre eles: tecnologia perfeita, requisitos verdadeiros e falsos, eventos e respostas, atividades essenciais e memória essencial, conforme discutido a seguir.

### **Tecnologia Perfeita**

Durante a fase de análise, o analista deve abstrair a tecnologia que será utilizada na implementação do sistema. Para orientar essa abstração, a Análise Estruturada recomenda que o analista, durante a fase de análise, concentre-se apenas nos aspectos lógicos do sistema, evitando pensar nos aspectos físicos. O problema dessa abordagem é que a diferença entre “aspectos lógicos e físicos” não é clara.

Partindo do princípio que os aspectos físicos de um sistema estão ligados à tecnologia de implementação, a Análise Essencial, com o objetivo de facilitar a identificação dos detalhes lógicos do sistema, adota uma abstração da tecnologia de implementação, denominada tecnologia perfeita. A tecnologia perfeita não possui limitações, isto é, existe um processador perfeito, capaz de executar qualquer processamento de forma instantânea, sem qualquer custo, sem consumir energia, sem gerar calor, sem jamais cometer erros ou parar de funcionar, e um repositório perfeito, capaz de armazenar quantidades

infinitas de dados e de ser acessado instantaneamente por qualquer processador, da forma que for mais conveniente.

Naturalmente, não existe uma tecnologia de implementação com tais características.

Então, Qual É A Utilidade Dessa Abstração?

Quando o analista pensa em aspectos físicos, ele está, na verdade, tentando identificar (e resolver) as limitações de uma determinada tecnologia. Pensamentos típicos do gênero são: quanto de espaço em disco vou precisar? Qual o melhor método de acesso aos dados, considerando as funções do sistema? De que capacidade de processamento devo necessitar? Contudo, nenhuma dessas preocupações é própria da fase de análise.

Considerando que a tecnologia a ser utilizada na implementação do sistema é perfeita, todas as perguntas anteriores deixam de ter importância e, portanto, não precisam estar no foco do analista. Assim sendo, para distinguir um requisito lógico de um requisito físico, utilizando a abstração de tecnologia perfeita, formule a seguinte pergunta ao identificar um requisito qualquer: “Para atender ao seu propósito, o sistema necessitará possuir essa capacidade ou essa característica, mesmo considerando que ele será implementado em uma tecnologia perfeita?” Se a resposta for sim, esse requisito é verdadeiro e deve ser modelado.

### **Requisito Verdadeiro e Requisito Falso**

Uma característica ou capacidade que um sistema deve possuir para atender ao seu propósito, mesmo considerando que será implementado em uma tecnologia perfeita, é dita um requisito verdadeiro. O conjunto de requisitos verdadeiros compreende a essência do sistema.

Um requisito falso, por outro lado, é uma capacidade ou característica que o sistema não precisa possuir para atender ao seu propósito, caso ele disponha de uma tecnologia de implementação perfeita.

### **Evento, Estímulo, Ação e Resposta**

Evento e resposta são nomes genéricos de interações entre o ambiente externo e o sistema. Um evento pode ser definido informalmente como um acontecimento do mundo exterior que requer do sistema uma resposta.

Corresponde a alguma mudança no ambiente externo que funcionará como um estímulo para o sistema, isto é, o sistema deve responder a esse estímulo para atender ao seu propósito. Uma resposta é o resultado da execução de um conjunto de ações no sistema, como consequência do reconhecimento pelo sistema de que um evento ocorreu. Uma resposta tipicamente pode ser:

Um fluxo de dados saindo do sistema para uma entidade externa;

Uma mudança de estado em um depósito de dados (inclusão, exclusão ou alteração);

Um fluxo de controle saindo de uma função para ativar uma outra.

Quando um evento ocorre, é produzido um estímulo para o sistema. Ao receber o estímulo, o sistema compreende que o evento ocorreu e ativa os processos (realiza as ações) necessários para produzir a resposta.

Os eventos são classificados em três tipos diferentes, dependendo da maneira como ocorrem e da natureza do estímulo que produzem:

Evento orientado por fluxo de dados: é provocado por uma entidade externa, a qual envia dados para o sistema. O estímulo produzido por esse tipo de evento é a chegada de um fluxo de dados que vai ativar uma função. Esses eventos são nomeados da seguinte forma:

<Entidade externa que provocou o evento> +

<ação – verbo na voz ativa> +

<estímulo – fluxo de dados enviado ao sistema> Ex.: Cliente envia pedido.

Cliente cancela pedido.

Evento orientado por controle: o estímulo provocado por este tipo de evento é a chegada ao sistema de um fluxo de controle, o qual apenas notifica o sistema da ocorrência do evento. Pode haver fluxos de dados complementares associados ao fluxo de controle, mas não é por meio da chegada do fluxo de dados que o sistema toma conhecimento da ocorrência do evento. Esse tipo de evento pode ser nomeado de duas maneiras, tendo por base a origem do evento:

Caso 1 – Uma entidade externa envia um comando (fluxo de controle) para o interior do sistema, ativando uma função.

<Entidade externa que provocou o evento> +

<ação – verbo na voz ativa> +

<complemento>

Ex.: Gerente solicita relação de clientes.

Diretoria autoriza o pagamento de uma fatura.

Caso 2 – Uma função é ativada por um fluxo de controle oriundo de outra função interna do sistema.

<Sujeito> + <ação – verbo na voz passiva>

Ex.: Nível de reabastecimento do estoque de um produto é atingido.

Evento temporal: é aquele em que o estímulo é a chegada ao sistema da informação de haver passado um determinado intervalo de tempo. Esses eventos estimulam as ações que o sistema tem de executar em datas previamente conhecidas, isto é, diariamente, mensalmente etc (o tempo passa e chega o momento do sistema fazer alguma coisa). Pode haver fluxos de dados complementares associados ao evento, mas não é através da chegada desses que o sistema toma conhecimento da ocorrência do evento. Os eventos temporais podem ser nomeados da seguinte forma:

<Indicação de tempo> + <ação> + <complemento> Ex.: Mensalmente, emitir relatório de vendas.

### **Atividades Essenciais**

São todas as tarefas que o sistema deve executar para atender completamente ao seu propósito, mesmo considerando que ele será implementado em uma tecnologia perfeita. Uma atividade essencial deve executar todo o conjunto de ações necessárias para responder completamente a um e somente um evento. As atividades essenciais subdividem-se em:

Atividades Fundamentais: produzem uma informação que é parte do propósito declarado do sistema. Assim sendo, o propósito do sistema é atendido pelas atividades fundamentais.

Atividades Custodiais: criam e mantêm a memória necessária à execução das atividades fundamentais, tipicamente, adquirindo dados do ambiente externo ao sistema e os armazenando nos depósitos de dados.

Como as atividades essenciais respondem completamente a um e somente um evento, a comunicação entre elas será feita sempre via memória e nunca diretamente. Essa característica da comunicação entre atividades essenciais torna o particionamento por eventos uma abordagem adequada para dividir o problema em partes independentes.

### **Memória Essencial**

A memória essencial corresponde ao conjunto mínimo de dados que deve ser armazenado pelo sistema para atender ao seu propósito, considerando que ele será implementado em uma tecnologia perfeita.

Modelo normalmente utilizado para modelar a memória essencial é o Modelo de Entidades e Relacionamentos (MER). Nos Diagramas de Fluxo de Dados (DFDs), a memória essencial também é representada, neste caso pelos depósitos de dados. Assim, ao se considerar a memória essencial de um

sistema, essas duas visões da mesma têm de estar consistentes. Para garantir essa consistência a seguinte regra deve ser observada: cada depósito de dados de um DFD deve corresponder a uma entidade ou a um relacionamento com atributos do MER.

Para levar em conta a abstração da tecnologia perfeita, os depósitos de dados / entidades não devem considerar a forma como os relacionamentos entre eles são estabelecidos. Por exemplo, quando se utilizam bancos de dados relacionais, chaves estrangeiras (atributos determinantes transpostos entre entidades) são armazenadas para representar um relacionamento entre entidades. Entretanto, deve-se ressaltar que essa é uma característica específica dos bancos de dados relacionais, uma tecnologia nada perfeita. Lembre-se que, na fase de análise, a tecnologia de implementação ainda não foi selecionada e deve ser considerada perfeita.

### Especificação da Essência do Sistema

A Análise Essencial sugere a construção de dois modelos principais, o modelo essencial e o modelo de implementação. Conforme discutido anteriormente, entendemos que apenas o modelo essencial deve ser objeto da fase de análise e, assim, discutiremos apenas a especificação da essência do sistema.

A especificação da essência do sistema, produto das fases de levantamento e análise de requisitos, é composta de dois modelos, como mostra a figura 5.1:

**Modelo Ambiental:** define a fronteira entre o sistema e o resto do mundo. Oferece uma perspectiva externa do sistema e é o principal produto da atividade de levantamento de requisitos.

**Modelo Comportamental:** define o comportamento do sistema necessário para interagir com o ambiente. Explora características internas do sistema e é o principal produto da atividade de análise de requisitos.

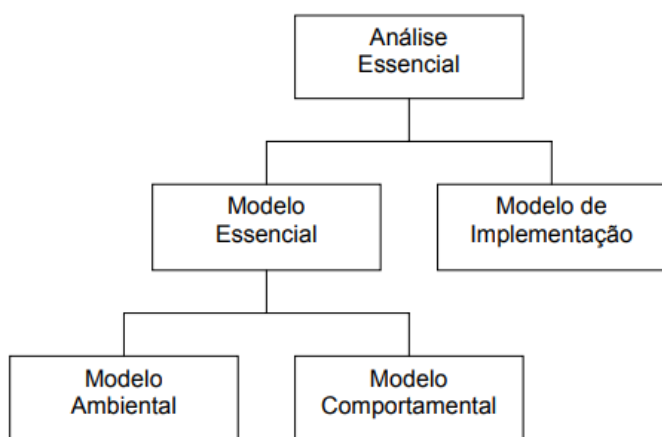


Figura 5.1 – A Análise Essencial e seus Modelos.

### O Modelo Ambiental

Representa o que o sistema deve fazer para atender ao ambiente, segundo uma perspectiva externa, ou seja, do usuário. É composto dos seguintes artefatos:

**Propósito do Sistema:** enuncia a finalidade do sistema. Pode ser acompanhado de uma breve descrição do contexto do sistema (mini-mundo).

**Lista de Eventos:** lista de eventos aos quais o sistema deve responder. Deve conter, pelo menos, o nome do evento, o estímulo e a resposta externa do sistema.

**Diagrama de Contexto:** representa o sistema como um único processo e suas interações com o ambiente. Pode ser acompanhado de um dicionário de dados.

A declaração de propósito (objetivos) do sistema deve ser elaborada em poucas frases, simples e precisas, em linguagem destituída de vocabulário técnico, passível de entendimento pelos clientes e usuários do sistema e pela administração da empresa, em geral. Não deve fornecer detalhes sobre como o sistema vai operar.

A lista de eventos é o produto principal do levantamento de requisitos quando a Análise Essencial é adotada, uma vez que os eventos constituem uma parte fundamental da abordagem desse método. De fato, o primeiro passo na especificação de um sistema é identificar a quais eventos do mundo exterior ele deverá responder.

Uma vez definidos os eventos, é possível construir o Diagrama de Contexto do sistema, mostrando como ele responde a todos os eventos externos relevantes.

Finalmente, pode ser útil elaborar uma descrição de como o sistema responderá a cada um dos eventos identificados na Lista de Eventos.

**Lista de Eventos**

Dado que a Lista de Eventos é o principal produto do levantamento de requisitos segundo a Análise Essencial, vale um exame mais cuidadoso desse artefato. Um formato largamente utilizado é o mostrado na tabela 5.1, que apresenta parte de uma lista de eventos para um sistema de biblioteca.

Evento	Tipo	Estímulo	Ação	Resposta
Bibliotecário cadastra dados de usuário	FD	dados-usuário	Cadastrar Usuário	(usuário cadastrado)
Diariamente, cancelar reservas vencidas	T	-	Cancelar Reservas	(reservas vencidas excluídas)
Usuário consulta livros	C	solicitação de consulta, parâmetro da consulta (título, autor ou assunto)	Consultar Livro	dados-livros

Tabela 5.1 – Exemplo de uma lista de eventos.

Nas duas primeiras colunas, o evento a ser tratado é descrito (segundo as regras de nomeação de eventos anteriormente apresentadas) e classificado – evento orientado por fluxo de dados (FD), orientado por controle (C) e evento temporal (T).

Na terceira coluna, são listados os estímulos que apontam para o sistema a ocorrência do evento. No caso dos eventos orientados por fluxo de dados, tipicamente um fluxo de dados representa o estímulo. Para os eventos temporais, é a passagem do tempo, considerado um estímulo implícito e, portanto, não é necessário representá-lo. Para os eventos orientados por controle, um fluxo de controle informando a solicitação feita ao sistema é o principal estímulo. Vale lembrar que, tanto no caso dos eventos temporais quanto dos eventos orientados a controle, fluxos de dados auxiliares podem ser também parte do estímulo. Neste caso, eles também devem aparecer.

Por fim, as duas últimas colunas registram a ação a ser feita pelo sistema e a resposta produzida. Quando a resposta for interna ao sistema, ela é escrita entre parênteses.

Para sistemas de médio a grande porte, a lista de eventos pode ser muito extensa. Um bom recurso para tratar a complexidade neste caso, pode ser a construção de listas de eventos separadas para cada um dos subsistemas identificados. Além disso, atividades custodiais que apenas cadastram dados no sistema, abrangendo a inclusão de um novo item, alteração de dados de um item existente, consulta básica ao item e exclusão de um item, podem ser tratadas como um único evento “Cadastrar Item”, classificado como orientado a fluxo de dados, ainda que a consulta e a exclusão sejam tipicamente eventos orientados por controle. A premissa para essa simplificação é que, na maioria dos sistemas,

haverá um grande número de eventos desta natureza e que seguem um comportamento bastante padrão e conhecido, não sendo necessário detalhá-lo.

### **O Modelo Comportamental**

modelo comportamental representa o que o interior do sistema deve fazer para atender aos eventos apontados pelo modelo ambiental, sendo o principal produto da análise de requisitos segundo o método da Análise Essencial. Esse modelo contém os seguintes artefatos:

#### **Diagrama de Entidades e Relacionamentos (DER)**

Diagramas de Fluxos de Dados (DFDs) Particionados por Eventos: Para cada evento do sistema, um DFD pode ser elaborado. Assim, a quantidade de diagramas pode chegar ao número de eventos na lista.

Diagramas de Estados: Representa o comportamento das entidades e relacionamentos com atributos ao longo do tempo. Tipicamente, constrói-se um diagrama de estados para cada entidade ou relacionamento com atributo do DER que possuir comportamento significativo, isto é, possuir mais de um estado ao longo de seu ciclo de vida.

Diagramas de Fluxos de Dados Organizados em Níveis Hierárquicos: representa os processos em níveis hierárquicos, a partir do diagrama zero. Os processos do diagrama zero são obtidos através do agrupamento de atividades essenciais dos DFDs particionados por eventos. Um critério de agrupamento bastante razoável é considerar o grau de coesão e acoplamento entre atividades essenciais. As seguintes heurísticas podem ser utilizadas, em conjunto ou em separado:

Procurar agrupar em um único processo todas as atividades essenciais que acessam um determinado depósito de dados, verificando se o processo resultante desse agrupamento é adequado para representar uma das funções do sistema.

Agrupar todas as atividades de custódia referentes a um mesmo depósito de dados.

Procurar identificar uma função do sistema, agrupando atividades essenciais que interagem com uma mesma entidade externa.

Representar no DFD-zero, um processo para cada uma das funções do negócio. Agrupar as atividades essenciais nos processos para os quais as suas ações mais contribuem.

Usando esta abordagem para a construção de diagramas hierárquicos, adotamos uma estratégia middle-out (do meio para fora), onde, a partir dos eventos, estabelecemos atividades essenciais (meio) para depois agrupá-las em níveis superiores (para cima) e, em seguida, especificá-las e, se necessário, explodi-las (para baixo).

Dicionário de Dados: descreve os dados representados no DER, nos DFDs e nos Diagramas de Estados.

Especificação da Lógica dos Processos: descreve a lógica dos processos do DFD que não foram detalhados em diagramas de nível inferior (lógica dos processos primitivos).

Como se pode perceber, a Análise Essencial faz uso praticamente das mesmas técnicas de modelagem da Análise Estruturada, a saber a Modelagem de Dados (utilizando modelos de Entidades e Relacionamentos), a Modelagem Funcional (utilizando Diagramas de Fluxo de Dados – DFDs) e a Modelagem de Controle (utilizando Diagramas de Estados). Isso é bastante natural, já que a Análise Essencial é, de fato, uma extensão da Análise Estruturada.

Na realidade, a principal diferença entre a Análise Essencial e a Análise Estruturada está na estratégia para atacar o problema: a primeira defende uma abordagem baseada em eventos, onde a Análise de Eventos passa a ser um passo fundamental; a segunda é baseada apenas na decomposição top-down da funcionalidade do Sistema.

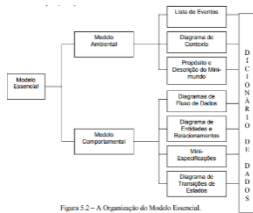


Figura 5.2 – A Organização do Modelo Essencial.

## Modelagem de Dados

A primeira atividade realizada no processo de construção do Modelo Comportamental da Análise Essencial de Sistemas deve ser a modelagem de dados e, para tal, o modelo de Entidades e Relacionamentos (ER) é utilizado. O modelo ER é uma técnica utilizada para representar os dados a serem armazenados em um sistema, tendo sido desenvolvida originalmente para dar suporte ao projeto de bancos de dados.

### Diagrama De Entidades E Relacionamentos

Basicamente, um diagrama ER representa as entidades do mundo real e os relacionamentos entre elas, que um sistema de informação precisa simular internamente. Além disso, entidades e relacionamentos podem ter atributos.

### Entidades

Uma entidade é uma representação abstrata de alguma coisa do mundo real que temos interesse em monitorar o comportamento. Pode ser a representação de um ser, um objeto, um organismo social etc. Assim, o funcionário João é uma entidade.

Entretanto, desejamos representar, de fato, conjuntos de entidades, isto é, grupos de entidades que têm características semelhantes. No modelo ER, conjuntos de entidades são representados graficamente por retângulos, como mostra a figura 5.3.

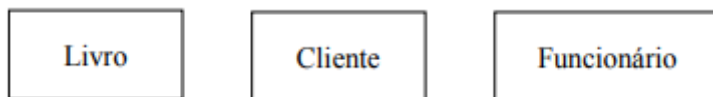


Figura 5.3 – Representação Gráfica de Conjuntos de Entidades.

Um conjunto de entidades representa todos os elementos do mundo real referidos pelo conjunto. Por exemplo, em um sistema de uma biblioteca, o conjunto de entidades Livro representa todos os livros de uma biblioteca.

Para estabelecermos uma padronização, neste texto, usaremos nomes de conjuntos de entidades sempre no singular e escritos com a primeira letra maiúscula. No entanto, isto não representa efetivamente uma regra. Além disso, usaremos o termo entidade para referenciar tanto entidades quanto conjuntos de entidades, de maneira indistinta, ainda que sejam conceitos diferentes. Essa é uma prática bastante comum e o contexto será suficiente para que o leitor perceba se estamos tratando de um conjunto de entidades ou de uma entidade específica.

### Relacionamentos

Um relacionamento é uma abstração de uma associação entre duas ou mais entidades. Por exemplo, podemos querer registrar que o funcionário João (uma entidade do conjunto Funcionário) está lotado (um relacionamento) no departamento de Vendas (uma entidade do conjunto Departamento). Um relacionamento Binário, como o citado no exemplo anterior, é uma representação abstrata da associação entre duas entidades.

Da mesma forma que com as entidades, estamos mais interessados em modelar conjunto de relacionamentos. Um conjunto de relacionamentos é um subconjunto do produto cartesiano dos conjuntos de entidades envolvidos, sendo representado por um losango com um verbo para indicar a ação e uma seta para informar o sentido de leitura, como mostra a figura 5.4. Além disso, assim como fizemos com



entidades, usaremos indistintamente o termo relacionamento para designar tanto relacionamentos entre entidades específicas como para referenciar o conjunto de relacionamentos que existem entre conjuntos de entidades. Opcionalmente, usaremos o termo instância (tanto de entidades quanto de relacionamentos) para referenciar um elemento do conjunto (de entidades e de relacionamentos, respectivamente).

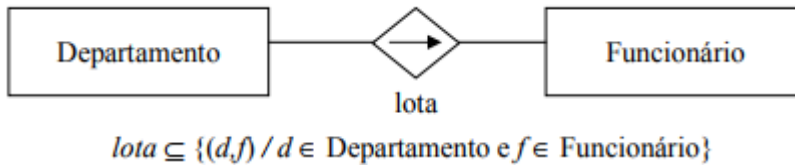


Figura 5.4 – Representação gráfica para relacionamentos.

É importante notar que todos os relacionamentos binários possuem uma leitura inversa. Se um departamento lota funcionários, então funcionários estão lotados em departamentos.

Conforme anteriormente mencionado, um conjunto de relacionamentos é um subconjunto do produto cartesiano das entidades envolvidas. É necessário, portanto, descrever de forma mais apurada qual é esse subconjunto. Isto é feito via definição de cardinalidades. Uma cardinalidade indica os números mínimo (cardinalidade mínima) e máximo (cardinalidade máxima) de associações possíveis em um relacionamento. Seja o seguinte caso: Um professor tem que estar lotado em um e somente um departamento, enquanto um departamento deve ter no mínimo 13 professores e no máximo um número arbitrário (N). Essa restrição imposta pelo mundo real deve ser considerada no modelo ER e ela é registrada usando-se cardinalidades, como mostra a figura 5.5.

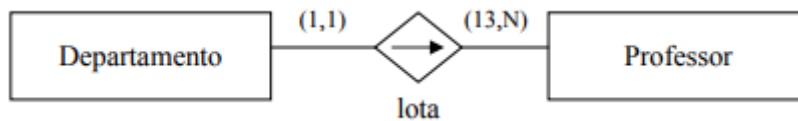


Figura 5.5 – Representação de cardinalidades em relacionamentos.

Vale destacar que a cardinalidade mínima aponta a quantidade de instâncias mínima necessária para que a associação seja estabelecida, considerando o momento em que uma instância de uma entidade é criada. Assim, no exemplo anterior, quando um novo professor for ser registrado no sistema, ele terá obrigatoriamente de estar lotado em um departamento.

Relacionamentos podem ser classificados em relação à cardinalidade mínima em relacionamentos totais e parciais. Dizemos que R é um relacionamento total em A se e somente se:  $\forall a \in A, \exists b \in B / (a, b) \in R$ , isto é, todo elemento de A tem de participar obrigatoriamente de R e, conseqüentemente, a cardinalidade mínima de R em relação a A é 1. Por outro lado, dizemos que R é um relacionamento parcial em A se e somente se:  $\exists a \in A$

$/ \forall b \in B / (a, b) \notin R$ , isto é, nem todo elemento de A participa de R e, conseqüentemente, a cardinalidade mínima de R em relação a A é 0.

É importante frisar que, entre duas entidades, podem existir vários relacionamentos diferentes, como mostra o exemplo da figura 5.6.

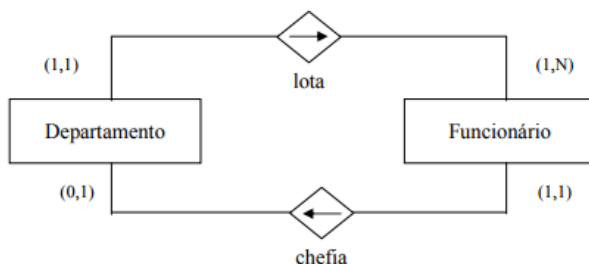


Figura 5.6 – Dois relacionamentos diferentes entre as mesmas entidades.

Além disso, uma entidade pode participar de relacionamentos com quaisquer outras entidades do modelo, inclusive com ela mesma (auto-relacionamento), como mostra a figura 5.7.

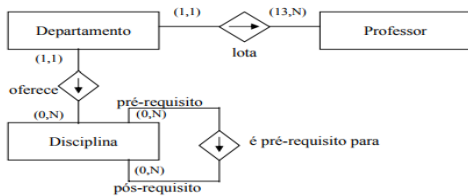


Figura 5.7 – Exemplo de auto-relacionamento.

No caso de auto-relacionamentos, é útil distinguir qual a atuação de cada elemento do conjunto de entidades no relacionamento e, portanto, é importante atribuir papéis. Assim no caso do auto-relacionamento é pré-requisito para da figura 5.7, estamos dizendo que:

Pré-requisitos  $\subseteq \{ (d1, d2) / d1, d2 \in \text{Disciplina} \text{ e}$

Papel (d1) = pré-requisito e papel (d2) = pós-requisito}

Até o momento, tratamos apenas de relacionamentos binários. Entretanto relacionamentos n-ários são também possíveis, ainda que muito menos corriqueiros. Um relacionamento ternário, por exemplo, só se caracteriza pelo terno, como mostra a figura 5.8. Os relacionamentos ternários normalmente são difíceis de se dar um nome e por isso é usual representá-los pelas iniciais das três entidades envolvidas, como mostra o exemplo da figura Neste exemplo, estamos dizendo que lojas compram materiais de fornecedores, sendo que uma loja pode comprar vários materiais diferentes, de fornecedores diferentes. Já um fornecedor pode vender vários materiais para diferentes lojas. Por fim um material pode ser adquirido por várias lojas a partir de vários fornecedores.

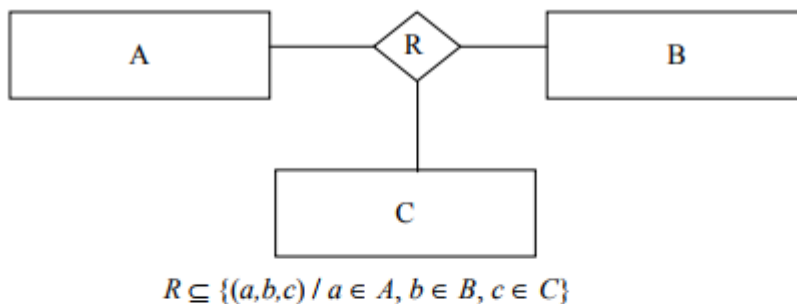


Figura 5.8 – Relacionamento Ternário.

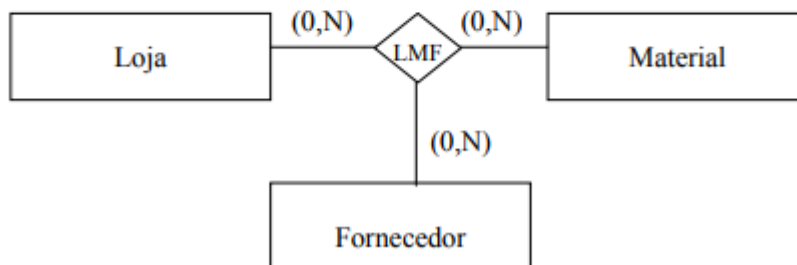


Figura 5.9 – Exemplo de Relacionamento Ternário.

Alguns relacionamentos são tão importantes que assumem o status de entidades. No modelo ER, esses relacionamentos são chamados de agregação ou agregados. Assim, uma agregação é uma abstração através da qual relacionamentos entre duas entidades são tratados como entidades em um nível mais alto, ou seja, um relacionamento binário R e as entidades envolvidas X e Y são considerados uma única entidade A, agregando todas as informações. Essa “nova entidade”, a agregação, pode, então, relacionar-se com outras entidades do modelo, como mostra a figura 5.10.

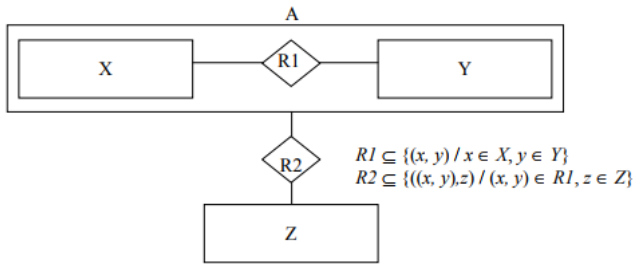


Figura 5.10 – Agregação.

A figura 5.11 mostra um exemplo de agregação no contexto bancário. Nesse exemplo, o par Cliente-Conta assume o status de um Correntista, para o qual um Cartão Magnético pode ser emitido.

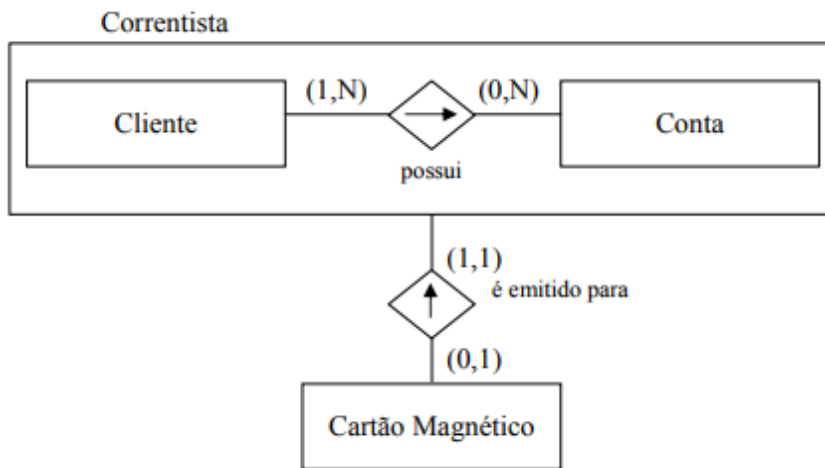


Figura 5.11 – Exemplo de Agregação.

Para prover maior facilidade na elaboração dos diagramas ER, representaremos a agregação com um retângulo envolvendo apenas o relacionamento, como mostra o exemplo da figura 5.12.

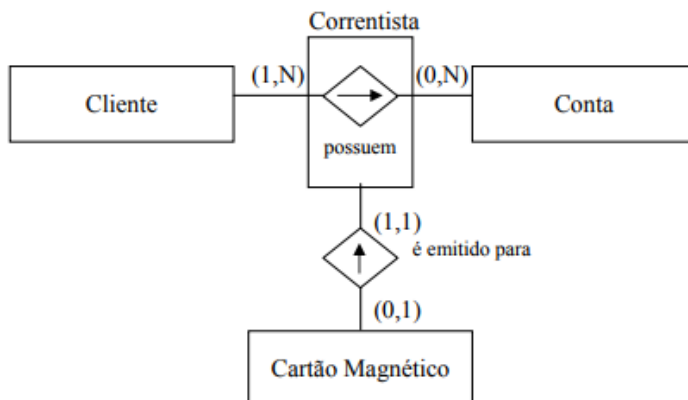


Figura 5.12 – Outra representação para Agregados.

É importante observar que agregações envolvendo relacionamentos N:1 ou 1:N, normalmente, não são necessárias. Em relacionamentos desta natureza, cada entidade cuja cardinalidade máxima é 1 (Y) só aparece no máximo uma única vez no relacionamento e, conseqüentemente, já representa o par que eventualmente possa ocorrer.

Assim, as duas versões apresentadas nas figuras 5.13 e 5.14 são equivalentes quanto às informações apresentadas e preferimos utilizar sempre a versão da figura 5.14, representando agregações apenas em relacionamentos N:N.

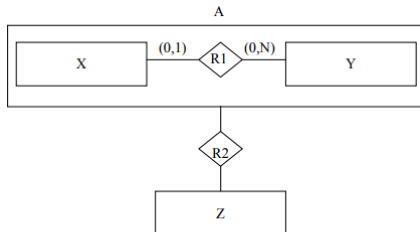


Figura 5.13 – Agregado em relacionamento 1:N.

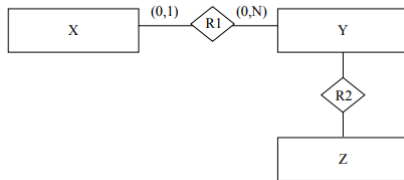


Figura 5.14 – Solução mais apropriada.

Às vezes, quer se representar um caráter opcional no que diz respeito à totalidade dos relacionamentos. Essa restrição pode ser de dois tipos:

Ou obrigatório: Apenas um dos relacionamentos ocorre efetivamente, mas sempre um deles ocorre. No exemplo da figura 5.15, todo contrato é financiado (não existe contrato que não seja financiado), mas pode ser financiado ou por um banco ou por um fornecedor.

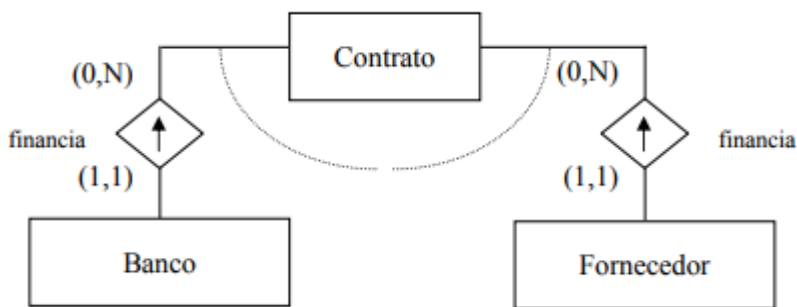


Figura 5.15 – Exemplo de Conector Ou-Obrigatório.

Ou opcional: Apenas um dos relacionamentos ocorre efetivamente, mas pode ser que nenhum dos dois ocorra. No exemplo da figura 5.16, nem todo contrato é financiado. Mas se um contrato for financiado, ele será financiado ou por um banco ou por um fornecedor.

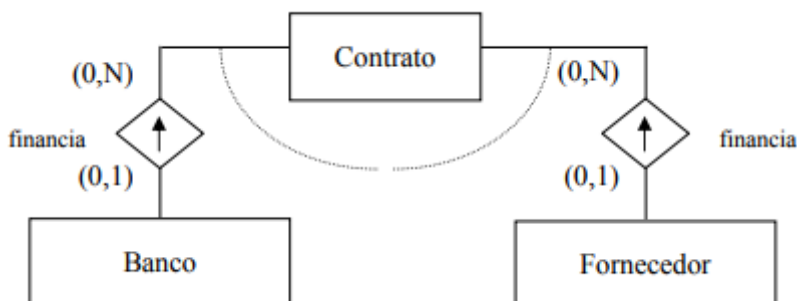


Figura 5.16 – Exemplo de Conector Ou-Opcional.

**Atributos**

Os atributos são utilizados para descrever características ou propriedades relevantes de entidades e relacionamentos. O conjunto de atributos de uma entidade ou relacionamento deve ser:

Completo: deve abranger todas as informações pertinentes.

Fatorado: cada atributo deve capturar um aspecto em separado.

Independente: os domínios de valores de atributos devem ser independentes uns dos outros.

Quando um atributo pode assumir apenas um único valor, ele é dito um atributo monovalorado. Por exemplo, os atributos nome e data de nascimento de uma entidade Funcionário são monovalorados, tendo em vista que uma instância de Funcionário, por exemplo, João, possui apenas um nome e uma data de nascimento.

Por outro lado, quando um atributo pode assumir vários valores para uma mesma instância, ele é dito multivalorado. Atributos multivalorados são representados com um asterisco (\*) associado. Por exemplo, o atributo telefone da entidade Funcionário é multivalorado, já que um mesmo funcionário pode ter mais de um telefone.

Atributos podem ter um valor vazio associado. Isso acontece quando para uma instância não existe um valor para aquele atributo, ou ele ainda não é conhecido. Por exemplo, o atributo telefone da entidade Funcionário pode receber um valor vazio, já que um funcionário específico pode não ter nenhum telefone, ou em um dado momento ele não ser conhecido.

Uma vez que estamos falando de conjuntos de entidades e relacionamentos, muitas vezes, é útil apontar que atributos são capazes de identificar univocamente um elemento de um conjunto. O conjunto de um ou mais atributos que identificam um elemento do conjunto é dito um atributo determinante. Atributos determinantes são sublinhados, como forma de destaque.

A figura 5.17 mostra uma representação gráfica para atributos. Ainda que essa notação possa ser empregada, de maneira geral, atributos são representados apenas no dicionário de dados para evitar modelos complexos e de difícil leitura.

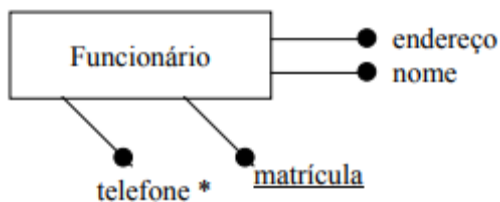


Figura 5.17 – Representação gráfica para atributos.

Vale destacar que atributos também são usados para descrever características de relacionamentos (atributos de relacionamentos) e que todas as considerações feitas até então são válidas. Atributos de relacionamentos são atributos que não são de nenhuma das duas entidades, mas sim do relacionamento entre elas e, em geral, estão relacionados com “protocolos” e datas, ou são resultantes de “avaliações”, tal como no exemplo da figura 5.18.

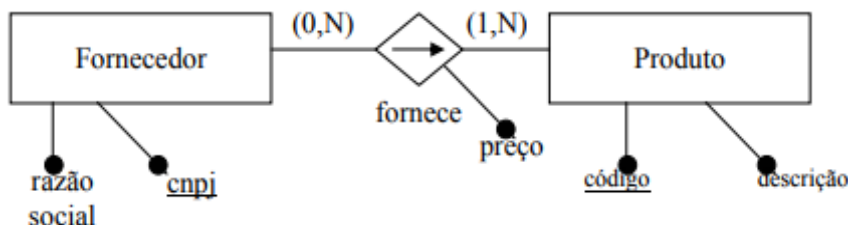


Figura 5.18 – Atributo de relacionamento.

Na prática, apenas os atributos de relacionamentos N:N são caracterizados como atributos de relacionamentos. No caso de relacionamentos 1:N ou N:1, os atributos do relacionamento podem ser perfeitamente caracterizados como atributos da entidade cuja a cardinalidade máxima é 1. No exemplo da figura 5.19, o atributo data-de-lotação pode ser perfeitamente caracterizado como atributo da entidade

Funcionário, já que um funcionário está lotado em apenas um departamento. Logo, a data de lotação do funcionário é a data de lotação dele no departamento ao qual está associado.

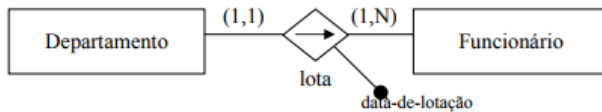


Figura 5.19 – Atributo de relacionamento caracterizado como atributo de uma das entidades.

Quando o relacionamento é N:N, há um teste que pode ser aplicado para se deduzir se um atributo é de um dos dois conjuntos de entidades ou se é do relacionamento. Seja o exemplo da figura 5.20.

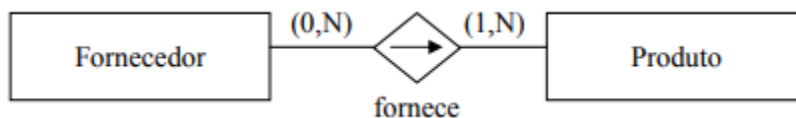


Figura 5.20 – Relacionamento N:N com atributos.

Fixa-se um produto, por exemplo impressora, e varia-se os fornecedores desse produto. Evidentemente o valor do atributo pode mudar. Por exemplo, a Casa do Analista vende uma impressora por R\$ 350, enquanto a loja Compute vende a mesma impressora por R\$

310. Se o valor do atributo mudar ao variarmos o elemento do outro conjunto de entidades, é porque este não é atributo do primeiro conjunto de entidades, no caso Produto.

Procedimento análogo deve ser feito, agora, para a outra entidade. Fixando-se um fornecedor e variando-se os produtos temos: A Casa do Analista vende uma impressora por R\$ 350 e um microcomputador por R\$ 1.700. Como o valor do atributo variou para a mesma entidade, é sinal de que ele não é atributo de Fornecedor.

Se o atributo preço não é nem de Produto, nem de Fornecedor, então é um atributo do relacionamento entre os dois conjuntos de entidades, como mostra a figura 5.21.

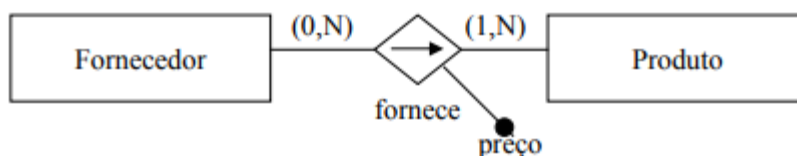


Figura 5.21 – Atributo do Relacionamento.

Uma outra questão a ser considerada relacionada a atributos é: a informação que se deseja modelar deve ser tratada como atributo de uma entidade ou como uma segunda entidade relacionada à primeira? Analisemos o seguinte exemplo: Será que departamento que oferece uma disciplina deve ser modelado como atributo da entidade Disciplina, ou merece ser uma nova entidade relacionada a ela? De forma geral, convém tratar um elemento de informação como uma segunda entidade se:

O elemento em questão tem atributos próprios;

A segunda entidade resultante é relevante para a organização;

O elemento em questão de fato identifica a segunda entidade;

A segunda entidade pode ser relacionada a diversas ocorrências da entidade-1 (1:N);

A segunda entidade relaciona-se a outras entidades que não a entidade-1.

Podemos notar que, no exemplo, todos os critérios anteriormente enumerados foram satisfeitos e, portanto, departamento deve ser tratado como uma nova entidade, como mostra a figura 5.22.

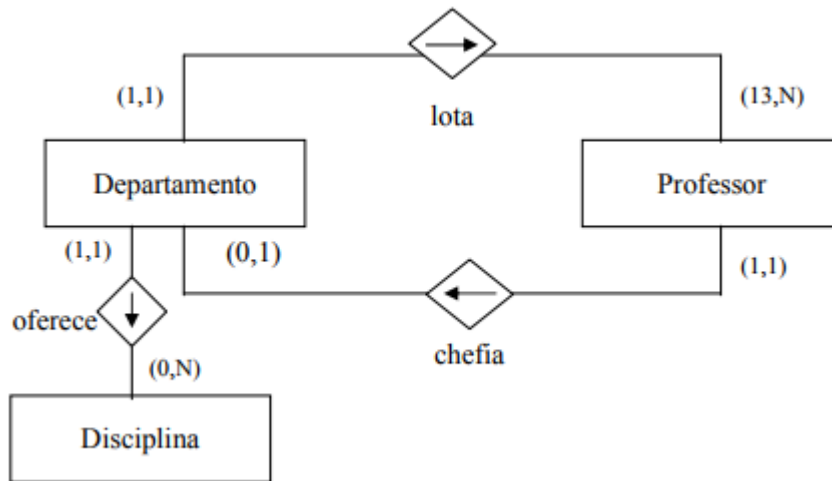


Figura 5.22 – Departamento como Nova Entidade.

**Particionamento de Entidades**

Muitas vezes, instâncias de entidades do mundo real se subdividem em categorias com atributos e relacionamentos parcialmente distintos. Passa a ser interessante, então, representar os atributos e relacionamentos comuns em um supertipo e os atributos e relacionamentos específicos em subtipos. Essa distinção pode ser feita por meio de:

Generalização: uma entidade de um nível mais alto é criada, para capturar as características comuns de entidades de nível mais baixo.

Especialização: uma entidade de nível mais alto de abstração é desmembrada em várias entidades de nível mais baixo.

A figura 5.23 mostra um exemplo de particionamento.

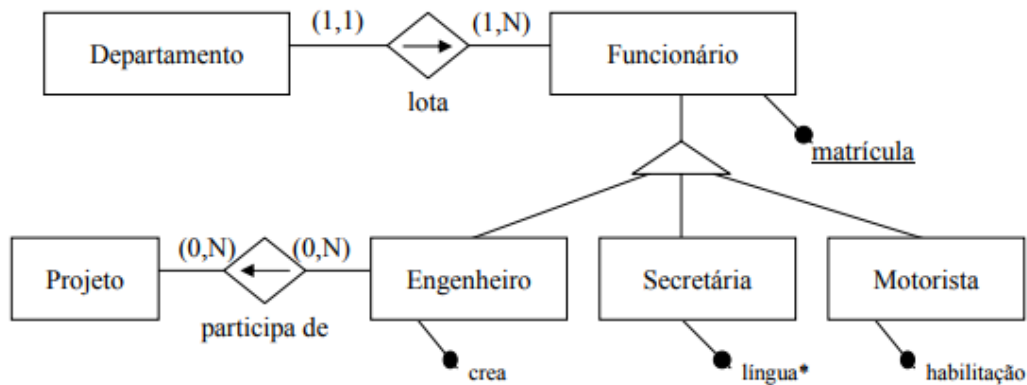


Figura 5.23 – Particionamento de um Conjunto de Entidades

Vale a pena observar que a simbologia utilizada na representação de entidades, relacionamentos e atributos mostrada neste texto não é padronizada, isto é, não foi definido um padrão único a ser seguido por todos que utilizam o Modelo ER. Assim, diferentes textos podem utilizar simbologias diferentes.

**Restrições de Integridade**

Muitas vezes, não somos capazes de modelar toda a estrutura de informação necessária com um diagrama ER, sobretudo no que diz respeito a restrições. Para suprir essa deficiência de representação dos diagramas ER, devemos adicionar ao modelo descrições textuais na forma de restrições de integridade, isto é, restrições do mundo real que devem ser descritas para manter a integridade do modelo.

Há dois tipos básicos de restrições de integridade: restrições sobre os possíveis relacionamentos e restrições sobre os valores dos atributos.

### Restrições de Integridade em Relacionamentos

Alguns relacionamentos só podem ocorrer se determinada restrição for satisfeita. Um exemplo de restrição de integridade são as cardinalidades. Por exemplo, se um funcionário só pode estar lotado em um único departamento, então não é possível relacionar um funcionário já lotado a um novo departamento.

A cardinalidade é uma das poucas restrições de integridade que são expressas no próprio diagrama ER. Outras restrições de integridade passíveis de representação nos diagramas ER são aquelas envolvidas nos conectores ou-opcional e ou-exclusivo, conforme discutido anteriormente. Entretanto, há outras situações que não conseguimos capturar. Seja o exemplo da figura 5.24.

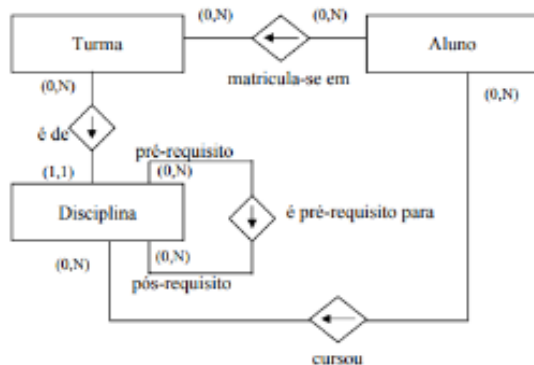


Figura 5.24 – Exemplo de auto-relacionamento.

Neste exemplo, gostaríamos de dizer, dentre outras coisas, que um aluno não pode se matricular duas vezes na mesma turma, ainda que ele possa se matricular em várias turmas. Além disso, um aluno só pode se matricular em uma turma de uma disciplina, se já tiver cursado todos os pré-requisitos daquela disciplina. Essas duas restrições sobre possíveis relacionamentos não são passíveis de serem capturadas pela notação dos diagramas ER e devem ser, então, escritas em linguagem natural como parte do modelo ER, mais precisamente no dicionário de dados do projeto.

Primeiro tipo de restrição apontado no exemplo anterior é dito uma restrição de integridade de repetição e indica quantas vezes os mesmos dois elementos das entidades podem ser relacionados.

Segundo tipo é dito uma restrição de integridade de dependência, apontando que um relacionamento pode ser restringido por um outro relacionamento, ou depender de seus relacionamentos anteriores.

### Restrições de Integridade sobre o Domínio dos Atributos

Ainda visando manter a integridade do modelo de dados, devemos descrever no dicionário de dados restrições de integridade que regem os valores dos atributos, isto é, o conjunto de valores que um atributo pode assumir. Esta tarefa deve ser feita utilizando-se dos seguintes recursos:

Enumeração: lista explícita de valores.

Ex: estado civil : solteiro, casado, desquitado, divorciado e viúvo.

Normas de aceitação: regras para se identificar se o valor é válido ou não.

Ex: nome: qualquer conjunto de caracteres alfanuméricos, começado por uma letra.

Intervalo: descrição de um subconjunto de um intervalo conhecido. Ex: mês: de 1 até 12.

Uma vez estabelecido o domínio, é interessante determinar valores possíveis e prováveis, isto é, alguns valores, apesar de poderem ocorrer, é pouco provável que ocorram, dependendo do contexto. Por exemplo, com relação ao atributo idade de um empregado, o valor 81 é um valor possível, mas será



que ele é um valor provável, considerando que a aposentadoria ocorre de maneira compulsória aos 70 anos?

Outros aspectos que devem ser considerados na descrição dos atributos são:

Obrigatoriedade: estabelecer se um determinado atributo pode ter um valor nulo a ele associado.

Ex: telefone: opcional; nome: obrigatório.

Dependência: os valores que um atributo pode assumir, muitas vezes, são dependentes dos valores de outros atributos. Neste caso é importante relacionar no dicionário de projeto como se dá esta dependência.

Ex: O valor do atributo dia depende fundamentalmente do valor do atributo mês.

A data de demissão de um funcionário tem de ser temporalmente posterior à sua data de admissão.

**Dicionário De Dados**

Dicionário de Dados é uma listagem organizada de todos os elementos de dados pertinentes ao sistema, com definições precisas para que os usuários e desenvolvedores possam conhecer o significado de todos os itens de dados manipulados pelo sistema. Esta listagem contém, em ordem alfabética, as seguintes definições:

Entidades e relacionamentos com atributos de um DER.

Depósitos de dados e fluxos de dados dos dfds, sendo que os primeiros devem corresponder às entidades e relacionamentos do DER.

Estruturas de dados que compõem os depósitos de dados ou fluxos de dados.

Elementos de dados que compõem os depósitos de dados, fluxos de dados ou estruturas de dados.

A figura 5.25 apresenta a notação adotada neste texto para elaboração de Dicionários de Dados.

Símbolo	Significado
=	é composto de
+	e
( )	dado ou estrutura opcional
[   ]	dados ou estruturas alternativas (ou exclusivo)
n{ }m	repetição de dados ou estruturas, onde n representa o número mínimo de repetições e m o número máximo. Se n e m não são especificados, significa zero ou mais repetições.
/* */	delimitadores de comentários
	atributo determinante

Figura 5.25 – Notação para Dicionários de Dados.

Os exemplos mostrados a seguir ilustram diversas situações e o emprego das notações.

O cliente pode possuir um telefone.

Cliente = /\*clientes da livraria\*/

cpf + nome + endereço + (telefone)

O cliente pode possuir mais de um telefone (ou mesmo nenhum). Cliente = cpf + nome + endereço + {telefone}

O cliente pode possuir até três telefones.

Cliente = cpf + nome + endereço + {telefone}3

O cliente pode possuir telefone comercial, residencial ou ambos.

Cliente = cpf + nome + endereço + [telefone-comercial | telefone-residencial | telefone-comercial + telefone-residencial]

### Modelagem De Estados

Diagramas de Estados são utilizados para descrever o comportamento de uma entidade ou de um relacionamento, com o objetivo de mostrar o comportamento do mesmo ao longo do seu tempo de vida. Diagramas de Estado descrevem todos os possíveis estados pelos quais uma entidade / relacionamento pode passar e as transições dos estados como resultado de eventos (estímulos) que atingem o mesmo. A figura 5.26 mostra a notação básica para diagramas de estado.



Figura 5.26 - Notação Básica para Diagramas de Estados.

Estados são representados por retângulos com os cantos arredondados e transições por setas, sendo que ambos podem ser rotulados. O rótulo de um estado pode conter, além de seu nome, uma indicação de que o estado possui uma atividade associada a ele, cuja sintaxe é:

/ atividade

Rótulo de uma transição pode ter até três partes, todas elas opcionais:

Evento [condição] / ação

Basicamente a semântica de um diagrama de estados é a seguinte: quando um evento ocorre, se a condição é verdadeira, a transição ocorre e a ação é realizada. A entidade passa, então, para um novo estado. Se neste estado, uma atividade deve ser realizada, ela é iniciada.

Fato de uma transição não possuir um evento associado, indica que a transição ocorrerá tão logo a atividade associada ao dado estado tiver sido concluída, desde que a condição seja verdadeira.

Quando uma transição não possuir uma condição associada, então ela ocorrerá sempre quando o evento ocorrer.

Embora ambos os termos ação e atividade denotem processos, eles não devem ser confundidos. Ações são associadas a transições e são consideradas processos instantâneos, isto é, ocorrem muito rapidamente, não sendo passíveis de interrupção. Atividades são associadas a estados, podendo durar bastante tempo. Assim, uma atividade pode ser interrompida por algum evento.

### Modelagem Funcional

A partir deste momento, passaremos a nos preocupar com a modelagem das funções que o sistema deverá executar para atender aos anseios dos usuários do sistema.

A técnica mais difundida para esta finalidade é a utilização de Diagramas de Fluxo de Dados - DFDs, proposta por Gane e Sarson em e por De Marco. Muitos outros autores citam esta técnica em suas obras, sendo que destacamos como referência.

Um Diagrama de Fluxo de Dados é um instrumento para a modelagem de processos, que representa um sistema como uma rede de processos, interligados entre si por fluxos de dados e depósitos de dados.

DFDs utilizam-se de quatro símbolos gráficos, visando representar os seguintes componentes: Processos, Fluxos de Dados, Depósitos de Dados e Entidades Externas. A figura 5.27 mostra a notação usada por Yourdon, que será a adotada neste texto. Através da utilização desses quatro componentes, podemos representar satisfatoriamente os processos e interações entre os elementos de um sistema.

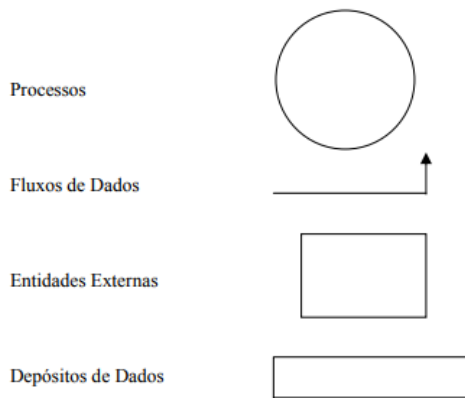


Figura 5.27 – Notação básica para construção de DFDs.

Além dos Diagramas de Fluxo de Dados, são necessários para uma completa modelagem das funções:

Dicionário de Dados;

Descrição da lógica dos processos simples que não mereçam ser decompostos em outros.

Um DFD mostra as fronteiras do sistema: aquilo que não for uma Entidade Externa será interno ao sistema, delimitando assim a fronteira do sistema. Além disso, mostra todas as relações entre dados (armazenados e que fluem no sistema) e os processos que manipulam e transformam esses dados, encarnando totalmente a filosofia do paradigma estruturado.

**Processos**

Representam as transformações e manipulações feitas sobre os dados em um sistema e correspondem a procedimentos ou funções que um sistema tem de prover. A ocorrência de um evento de um dos seguintes tipos deve ser representada como um processo em um DFD:

transformações do conteúdo de um dado de entrada no conteúdo de um dado de saída, sem armazenamento interno no sistema (figura 5.28);

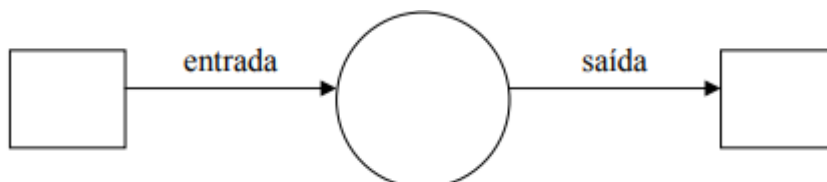


Figura 5.28 – Transformações de dados.

Inserções ou modificações do conteúdo de dados armazenados, a partir do conteúdo (possivelmente transformado) de dados de entrada, como mostra a figura 5.29;

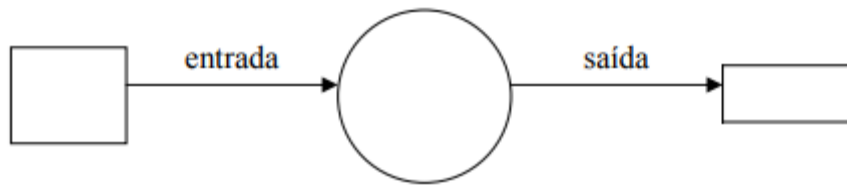


Figura 5.29 – Armazenamento de dados.

Transformações de dados previamente armazenados no conteúdo de um dado de saída, como mostra a figura 5.30.

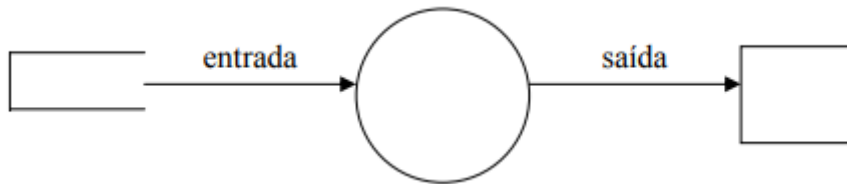


Figura 5.30 – Geração de dados de saída a partir de dados armazenados.

Um processo é representado por um círculo, com uma sentença simples (verbo + objetos) em seu interior e, opcionalmente, um identificador (número). A sentença deve tentar descrever o melhor possível a função a ser desempenhada, sem ambiguidades. Devem ser evitados nomes muito físicos (p. ex., gravar, imprimir etc) ou muito técnicos (p. ex., apagar, fazer backup etc).

Os processos representados em um DFD não precisam ser necessariamente funções a serem informatizadas. Muitas vezes, para se prover um entendimento mais completo do sistema, processos manuais ou mistos (parte manual, parte informatizada) são representados.

Toda transformação de dados deve ser representada e, deste modo, não se admite ligação direta entre entidades externas e depósitos de dados. Por outro lado, devemos observar se um mesmo fluxo de dados entra e sai de um processo sem modificação, já que todo processo transforma dados.

Como já mencionado anteriormente, para uma completa modelagem das funções, são necessários, além dos DFDs, um Dicionário de Dados e as Especificações das Lógicas dos processos. Deste modo, só teremos um entendimento completo de um processo, após descrevermos sua lógica.

As especificações das lógicas dos processos só devem ser feitas para processos simples. Processos complexos devem ser decompostos em outros processos, até se atingir um nível de reduzida complexidade. Esta descrição não deve ser confundida com o detalhamento integral da lógica do processo que deverá ser feito na fase de projeto, mas deve servir de base para essa outra etapa.

Uma heurística para se definir se um processo merece ou não ser representado em um DFD é dada em função da descrição de sua lógica. Quando essa descrição utilizar aproximadamente uma ou duas páginas, então o processo parece estar adequado. Processos descritos em três ou quatro linhas são simples demais para serem tratados como processos em um DFD. Por outro lado, se a descrição da lógica do processo necessitar de quatro ou mais páginas, então esse processo está muito abrangente e não deve ser tratado como um único processo, mas sim deve ser decomposto em processos de menor complexidade. Para situações desta natureza, duas técnicas são utilizadas: fissão ou explosão, como estudaremos mais à frente.

Como regra geral, os fluxos de erro e exceção não devem ser mostrados nos diagramas, mas apenas na descrição da lógica dos processos. Essa regra só deve ser desrespeitada quando tais fluxos forem muito significativos para a comunidade usuária.

### Fluxos de Dados

Fluxos de dados são utilizados para representar a movimentação de dados através do sistema. São simbolizados por setas, que identificam a direção do fluxo, e devem ter associado um nome o mais

significativo possível, de modo a facilitar a validação do diagrama com os usuários. Esse nome deve ser um substantivo que facilite a identificação do dado (ou pacote de dados) transportado.

Um fluxo de dado em um DFD pode ser considerado como um caminho através do qual poderão passar uma ou mais estruturas de dados em tempo não especificado. Note que em um DFD não se representam elementos de natureza não informacional, isto é, dinheiro, pessoas, materiais etc.

Devemos observar se um fluxo de dados entra e sai de um processo sem modificação. Isso representa uma falha, haja visto que um processo transforma dados. Embora possa parecer um tanto óbvio, é bom lembrar que um mesmo conteúdo pode ter diferentes significados em pontos distintos do sistema e, portanto, os fluxos devem ter nomes diferentes.

No DFD da figura 5.31, um mesmo conjunto de informações sobre um cliente tem significados diferentes quando passa pelos fluxos dados-novo-cliente e dados-cliente. No primeiro caso, os dados ainda não foram validados e, portanto, podem ser válidos ou inválidos, enquanto, no segundo fluxo, esses mesmos dados já foram validados.

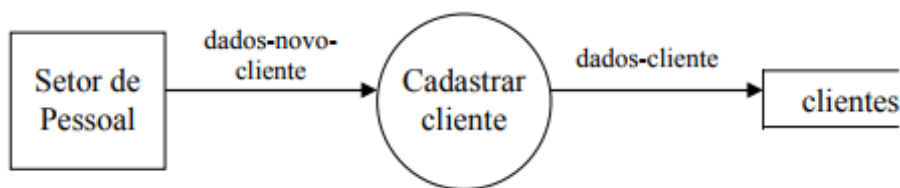


Figura 5.31 – Mesmo conteúdo de dados em fluxos diferentes.

Fluxos de dados que transportam exatamente o mesmo conteúdo de/para um depósito de dados, não precisam ser nomeados. No exemplo da figura 5.31, se o fluxo dados-cliente apresentar exatamente o mesmo conteúdo do depósito clientes, não há necessidade de nomeá-lo, como mostra a figura 5.32.

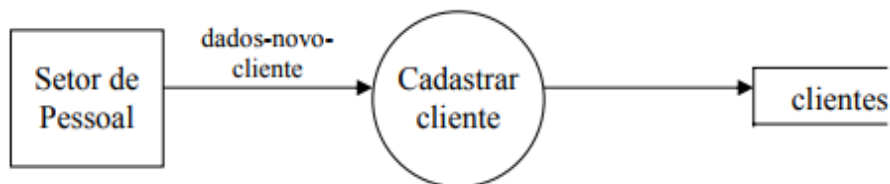


Figura 5.32 – Fluxo de dados não nomeado.

Fluxos de erro ou exceção (no exemplo, dados-cliente-inválidos) só devem ser mostrados em um DFD, se forem muito significativos para o seu entendimento. Caso contrário, devem ser tratados apenas na descrição da lógica do processo.

Setas ramificadas significam que o mesmo fluxo de dados está indo de uma fonte para dois destinos diferentes, isto é, cópias do mesmo pacote de dados estão sendo enviadas para diferentes partes do sistema, como mostra a figura 5.33.

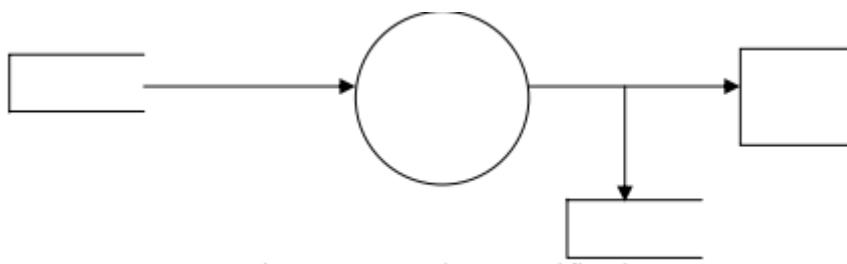


Figura 5.33 – Fluxo ramificado.

Quando for necessário cruzar fluxos de dados em um DFD, devemos utilizar um arco, como mostra a figura 5.34.

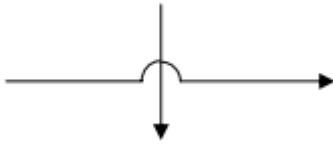


Figura 5.34 – Fluxos de dados que se cruzam em um diagrama.

É importante realçar que DFDs não indicam a sequência na qual fluxos de dados entram ou saem de um processo. Essa sequência é descrita apenas na especificação do processo.

### Depósitos de Dados

Depósitos de dados são pontos de retenção permanente ou temporária de dados, que permitem a integração entre processos assíncronos, isto é, processos realizados em tempos distintos. Sem nos comprometermos quanto ao aspecto físico, representam um local de armazenamento de dados entre processos.

Um depósito de dados é representado por um retângulo sem a linha lateral direita, com um nome e um identificador (opcional) em seu interior. Às vezes, para evitar o cruzamento de linhas de fluxos de dados ou para impedir que longas linhas de fluxos de dados saiam de um lado para outro do diagrama, um mesmo depósito de dados pode ser representado mais de uma vez no diagrama. Nessa situação, adicionamos uma linha vertical na lateral esquerda do retângulo, como mostra a figura 5.35.



Figura 5.35 – Notação para depósitos de dados.

Um depósito de dados não se altera quando um pacote de informação sai dele através de um fluxo de dados. Por outro lado, um fluxo para um depósito representa uma das seguintes ações:

- uma inclusão, isto é, um ou mais novos pacotes de informação estão sendo introduzidos no depósito;
- uma atualização, ou seja, um ou mais pacotes estão sendo modificados, sendo que isso pode envolver a alteração de todo um pacote, ou apenas de parte dele;
- uma exclusão, isto é, pacotes de informação estão sendo removidos do depósito.

A semântica dos acessos aos depósitos de dados é mostrada na figura 5.36.

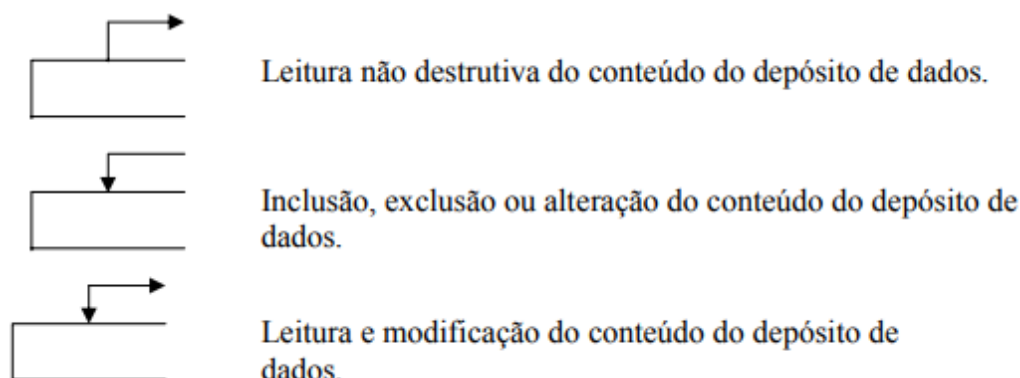


Figura 5.36 – Semântica dos acessos a depósitos de dados em um DFD.

Quando examinamos fluxos de dados que entram ou saem de um depósito, surge uma dúvida: o fluxo representa um único pacote, múltiplos pacotes, partes de um pacote, ou partes de vários pacotes de dados? Em algumas situações, essas dúvidas podem ser respondidas pelo simples exame do rótulo do fluxo e, para tal, adotamos a seguinte convenção:

se um fluxo não possuir rótulo ou tiver o mesmo rótulo do depósito de dados, então um pacote inteiro de informação ou múltiplas instâncias do pacote inteiro estão trafegando pelo fluxo;

se o rótulo de um fluxo nomeado for diferente do rótulo do depósito, então as informações que estão trafegando são um ou mais componentes (partes) de um ou mais pacotes e estarão definidas no dicionário de dados.

Muitas vezes, diferentes sistemas compartilham uma mesma base de dados e, portanto, vários sistemas poderão estar lendo e atualizando os conteúdos de um mesmo depósito de dados. É interessante mostrar este fato explicitamente no DFD e, nesse caso, podemos notar três situações distintas:

O sistema em questão apenas lê as informações do depósito de dados, não sendo responsável por qualquer alteração de seu conteúdo. Neste caso, utilizamos a notação da figura 5.37.

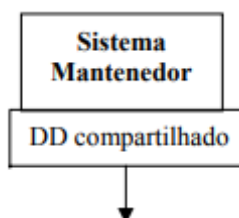


Figura 5.37 – Sistema apenas acessando depósito de dados mantido por outro sistema.

Sistema em questão apenas gera as informações que são utilizadas por outros sistemas. Representamos essa situação segundo a notação da figura 5.38

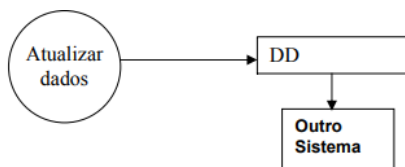


Figura 5.38 – Sistema atualizando dados utilizados por outro sistema.

Ambos os sistemas atualizam o depósito de dados. A notação para esta situação é mostrada na figura 5.39.

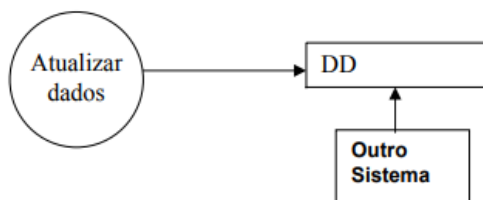


Figura 5.39 – Ambos os sistemas atualizando dados de um mesmo depósito.

Essas notações são exceções à regra de que os dados não devem fluir diretamente entre uma entidade externa e um depósito de dados, sem passar por um processo responsável pela transferência dos dados. Fora as situações anteriormente descritas, devemos observar a integridade de um depósito de dados segundo dois prismas:

Observar se todos os elementos de dados que fazem parte do depósito têm como efetivamente chegar lá, isto é, fazem parte de pelo menos um fluxo de dados que chega ao depósito.

Observar se todos os elementos de dados que fazem parte do depósito são, em algum momento, solicitados por um processo, isto é, fazem parte de pelo menos um fluxo de dados que sai do depósito.

**Entidades Externas**

Entidades externas ou terminadores são fontes ou destinos de dados do sistema. Representam os elementos do ambiente com os quais o sistema se comunica. Tipicamente, uma entidade externa é uma pessoa (p.ex. um cliente), um grupo de pessoas (p. ex. um departamento da empresa ou outras instituições) ou um outro sistema que interaja com o sistema em questão. Uma entidade externa deve ser identificada por um nome e representada por um retângulo.

Assim como no caso dos depósitos de dados, em diagramas complexos, podemos desenhar um mesmo terminador mais de uma vez, para se evitar o cruzamento de linhas de fluxos de dados ou para impedir que longas linhas de fluxos de dados saiam de um lado a outro do diagrama. Nesse caso, convencionou-se utilizar um traço diagonal no canto inferior direito do símbolo da entidade externa, como mostra a figura 5.40.



Figura 5.40 – Notações para representar entidades externas.

Ao identificarmos alguma coisa ou sistema como uma entidade externa, estamos afirmando que essa entidade está fora dos limites do sistema em questão e, portanto, fora do controle do sistema que está sendo modelado. Assim, qualquer relacionamento existente entre entidades externas não deve ser mostrado em um DFD.

Se percebermos que, em algum ponto do sistema, descrevemos algo que ocorre dentro de uma entidade externa ou relacionamentos entre entidades externas, é necessário reconhecer que a fronteira do sistema é na realidade mais ampla do que foi estabelecido inicialmente e, portanto, deve ser revista.

Uma vez que os terminadores são externos ao sistema, os fluxos de dados que os interligam aos diversos processos representam a interface entre o sistema e o mundo externo.

**Relações Entre DFDS e DERS**

Conforme discutido anteriormente, depósitos de dados são representações em um DFD para entidades e relacionamentos em um modelo ER. Entretanto, em um DFD, não há uma representação explícita dos relacionamentos entre entidades. Para indicar que o relacionamento entre entidades existe, a representação dos acessos dos processos aos depósitos de dados deve obedecer à seguinte regra geral: ao criar ou excluir um relacionamento ou uma entidade que participa de um relacionamento, mostre o acesso aos depósitos de dados que correspondem ao relacionamento e às entidades que participam do relacionamento. A figura 5.41 mostra a representação gráfica desses acessos.

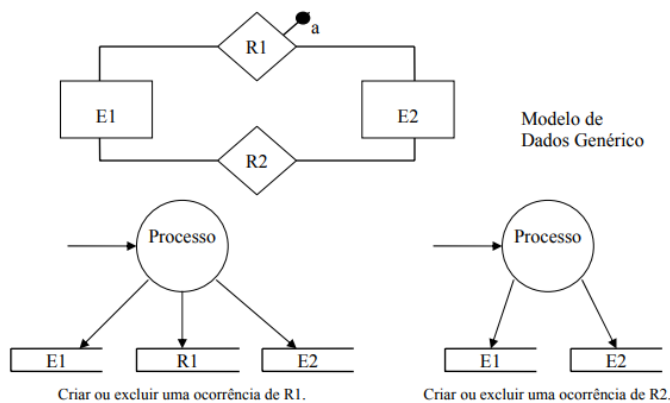


Figura 5.41 – Acessos a depósitos de dados.



No caso do relacionamento R1, como esse relacionamento tem um atributo (a), ele é representado em um DFD como sendo um depósito de dados. Assim, para criar ou excluir uma ocorrência de R1, representam-se acessos a R1, E1 e E2. Já o relacionamento R2, como esse não possui atributos, não dá origem a um depósito de dados. Para criar ou excluir uma ocorrência de R2, são representados acessos a E1 e E2.

### Construindo DFDs

Como já mencionado no estudo sobre processos, é uma boa prática manter um certo nível de complexidade nos processos representados em um DFD. Esse nível de complexidade pode ser estabelecido pelo tamanho da especificação da lógica do processo ou pelo número de processos em um diagrama. Se tal nível de complexidade for superado, devemos utilizar uma das seguintes técnicas para decompor o DFD: fissão ou explosão.

#### Fissão

Na fissão, o processo complexo deve ser substituído no próprio DFD do sistema por

Um número de processos mais simples. Por exemplo, se um processo requer 8 páginas de especificação de lógica, ele pode ser substituído por 4 processos, cada um deles tendo aproximadamente 2 páginas.

Problema na utilização desta técnica é a sobrecarga a que o diagrama poderá ficar sujeito, dificultando sua leitura.

#### Explosão

Processo original permanece no diagrama, sendo criado um novo dfd de nível inferior, consistindo de processos menos complexos. Assim, um projeto não é representado por um único dfd, mas sim por um conjunto de dfds em vários níveis de decomposição funcional.

Quando a explosão é utilizada, alguns aspectos importantes devem ser observados. O primeiro deles diz respeito ao número de níveis que devem ser esperados em um sistema. A priori, esse número não deve ser pré-fixado, mas lembre-se que o número total de processos cresce exponencialmente quando se passa de um nível para o imediatamente inferior.

Tipicamente são quatro os níveis de representação:

C – Contexto: mostra o sistema como uma “caixa-preta”, trocando informações (fluxos de dados) com entidades externas ao sistema. Define o escopo de abrangência do sistema, indicando que se está renunciando à possibilidade de examinar qualquer coisa além da fronteira definida pelas entidades externas. É parte integrante do modelo ambiental, segundo a Análise Essencial.

0 (Zero) – Geral ou de Sistema: é uma decomposição do diagrama de contexto, mostrando o funcionamento do sistema em questão, isto é, as grandes funções do sistema e as interfaces entre elas. Os processos nesse diagrama recebem os números 1, 2, 3 etc. É necessário assegurar a coerência entre os diagramas C e 0, isto é, assegurar que os fluxos de dados entrando ou saindo do diagrama 0 efetivamente reproduzem as entradas e saídas do diagrama C. No diagrama 0, devem aparecer os depósitos de dados necessários para a sincronização dos processos.

N – Detalhe: Uma diagrama de detalhe representa a decomposição de um dos processos do diagrama 0 e, portanto, é nomeado com o número e o nome do processo que está sendo detalhado. A princípio, deverão ser elaborados diagramas N para os processos do diagrama 0 que sejam complexos e, portanto, careçam de decomposição. É necessário resguardar a coerência entre o diagrama 0 e cada diagrama detalhado elaborado. Os processos em um diagrama N são numerados com o número do processo que está sendo detalhado (p. ex., 2) e um número sequencial, separados por um ponto (p. ex., 2.1, 2.2, etc.).

E – Detalhe Expandido: um diagrama deste tipo representa a decomposição de um dos processos do diagrama N. Esse nível de decomposição pode vir a ser necessário caso um processo do nível N ainda for muito complexo. Esse nível pode ser desdobrado sucessivamente até se atingir o grau necessário

de simplicidade. Entretanto, se muitos níveis forem necessários, cuidado! Provavelmente, o contexto funcional da aplicação (diagrama de contexto) está muito abrangente e merece revisão.

Fissão Ou Explosão?

Recomenda-se o uso da fissão para sistemas de pequeno a médio porte, em que a leitura do diagrama não fica prejudicada pelo aparecimento de mais alguns processos no diagrama de sistema. A fissão possui a vantagem de representar todo o sistema em um único DFD, não sendo necessário recorrer a outros diagramas para se obter um entendimento completo de suas funções. Em sistemas maiores, o uso da fissão pode se tornar inviável, sendo recomendado, então, o uso da explosão.

### Recomendações Para A Construção De DFDs

Escolha nomes significativos para todos os elementos de um DFD. Utilize termos empregados pelos usuários no domínio da aplicação.

Os processos devem ser numerados de acordo com o diagrama a que pertencem.

Evite desenhar DFDs complexos.

Cuidado com os processos sem fluxos de dados de entrada ou de saída.

Cuidado com os depósitos de dados que só possuem fluxos de dados de entrada ou de saída.

Depósitos de dados permanentes devem manter estreita relação com os conjuntos de entidades e de relacionamentos do modelo ER. Fique atento ao princípio de conservação de dados, isto é, dados que saem de um depósito devem ter sido previamente lá colocados e dados produzidos por um processo têm de ser passíveis de serem gerados por esse processo.

Quando do uso de explosão, os fluxos de dados que entram e saem em um diagrama de nível superior devem entrar e sair no nível inferior que o detalha.

Mostre um depósito de dados no nível mais alto em que ele faz a sincronização entre dois ou mais processos. Passe a representá-lo em todos os níveis inferiores que detalham os processos envolvidos.

Não represente no DFD fluxos de controle ou de material. Como o nome indica, DFDs representam fluxos de dados.

Só especifique a lógica de processos primitivos, ou seja, aqueles que não são detalhados em outros diagramas.

### Técnicas de Especificação de Processos

Quando chegamos a um nível de especificação em que os processos não são mais decomponíveis, precisamos complementar essa especificação com descrições das lógicas dos processos. A especificação de processos deve ser feita de forma que possa ser validada por analistas e usuários. Entretanto, encontramos muitos problemas na descrição de forma narrativa, entre os quais podemos citar:

Uso de expressões do tipo: mas, todavia, a menos que.

Por exemplo, qual a diferença entre as declarações abaixo?

Somar A e B, a menos que A seja menor que B, onde, neste caso, subtrair A de B.

Somar A e B. Entretanto, se A for menor que B, a resposta será a diferença entre B e A.

Somar A e B, mas subtrair A de B quando A for menor que B.

Total é a soma de B e A. Somente quando A for menor que B é que a diferença deve ser usada como o total.

Ao analisarmos essas frases, notamos que não existe diferença lógica entre elas, entretanto as formas narrativas apresentadas mascaram a semelhança existente. Se ao invés de usarmos uma forma narrativa, usarmos uma forma padrão do tipo se- então-senão, teremos maior clareza e validação.

se  $A < B$

Uso de comparativos como: Maior que / Menor que, Mais de / Menos de.

Seja a seguinte declaração: “Até 20 unidades, sem desconto. Mais de R\$20, 5% de desconto”.

E exatamente 20 unidades, que tratamento deve ser dado?

Ambiguidades do E/OU.

Seja a seguinte declaração: “Clientes que gerarem mais de um milhão de reais em negócios por ano e possuírem um bom histórico de pagamentos **ou** que estiverem conosco há mais de 20 anos, devem receber tratamento prioritário”.

Quem deverá receber tratamento prioritário? Clientes com mais de 1 milhão em negócios por ano que possuírem bom histórico de pagamentos? Clientes com mais de 20 anos? Clientes com mais de 1 milhão e (ou bom histórico, ou mais de 20 anos)?

Note que pela declaração não fica claro quando deverá ser aplicado o tratamento prioritário.

### Uso de Adjetivos Indefinidos

Na declaração do item anterior, o que é um bom histórico de pagamentos? Devemos tomar cuidado ao utilizarmos adjetivos indefinidos. Quando o fizermos, devemos tomar o cuidado de defini-los.

Para administrar os problemas oriundos da narrativa, são utilizadas técnicas de especificação de processos, entre as quais podemos citar:

Português Estruturado

Tabelas de Decisão

Árvores de Decisão

Combinação das técnicas acima

### Português Estruturado

Português Estruturado é um subconjunto do Português, cujas sentenças são organizadas segundo as três estruturas de controle introduzidas pela Programação Estruturada: sequência, seleção e repetição.

**Instruções de Sequência:** grupo de instruções a serem executadas que não tenham repetição e não sejam oriundas de processos de decisão. São escritas na forma imperativa, como no exemplo abaixo.

Obter ... Atribuir ... Armazenar ...

**Instruções de Seleção:** quando uma decisão deve ser tomada para que uma ação seja executada, utilizamos uma instrução de seleção. As instruções de seleção são expressas como uma combinação se-então-senão, conforme abaixo.

Se <condição>

Então grupo\_de\_ações\_1; senão grupo\_de\_ações\_2;

Exemplo:

Fim-se;

Se número\_de\_dependentes = 0 então salário\_família = 0;

Senão salário\_família = salário\_mínimo / 3;

Fim-se;

Quando existirem várias ações dependentes de uma mesma condição, que sejam mutuamente exclusivas, podemos utilizar uma estrutura do tipo caso, conforme abaixo.

Caso <condição> =

Valor\_1 : grupo\_de\_ações\_1; valor\_2 : grupo\_de\_ações\_2;

... ..

Valor\_n : grupo\_de\_ações-n; fim-caso;

Instruções de repetição: aplicadas quando devemos executar uma instrução, ou um grupo de instruções, repetidas vezes. A estrutura de repetição pode ser usada de três formas distintas:

Para cada "x" faça

Grupo\_de\_ações; fim-para;

Exemplo:

Para cada aluno faça

Média = (prova\_1 + prova\_2) / 2; imprima média;

Fim-para;

Enquanto <condição for verdadeira> faça grupo\_de\_ações;

Fim-enquanto;

Exemplo:

Enquanto existir notas faça ler nota;

Consistir dados; fim-enquanto;

Repita

Grupo\_de\_ações;

Até que <condição seja verdadeira>;

Exemplo:

Repita

Ler nota; consistir dados;

Até que todas as notas tenham sido processadas;

Uma especificação de processo em português estruturado deve possuir as seguintes características gerais:

Deve ser clara, concisa, completa e livre de ambiguidades;

Todos os dados citados na especificação que estejam definidos no dicionário de dados devem estar em itálico, incluindo depósitos de dados;

Os dados definidos localmente são escritos em fonte normal;

Sempre que um comando de seleção ou repetição for utilizado, os comandos do bloco interno (grupo\_de\_ações) devem estar identados, de modo a dar a clareza de que esses comandos fazem parte das ações da seleção ou repetição.

### Árvores De Decisão

Árvores de Decisão são excelentes para mostrar a estrutura de decisão de um processo. Os ramos da árvore correspondem a cada uma das possibilidades lógicas. É uma excelente ferramenta para esquematizar a estrutura lógica e para obter do usuário a confirmação de que a lógica expressa está correta. De forma clara e objetiva, permite a leitura da combinação das circunstâncias que levam a cada ação.

Como podemos notar pelo exemplo da figura 5.42, uma Árvore de Decisão é muito boa para representar a lógica decisória. Entretanto, se for necessário descrever a lógica de um processo como um conjunto de instruções, combinando decisões e ações intermediárias, a árvore de decisão deve ser preterida em favor do português estruturado ou combinada a ele.

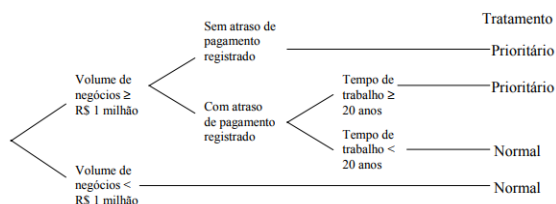


Figura 5.42 – Exemplo de Árvore de Decisão.

### Tabelas De Decisão

Tabelas de decisão são usadas em aplicações semelhantes às das árvores de decisão. As árvores de decisão são mais indicadas, quando o número de decisões for pequeno e nem todas as combinações de condições forem possíveis. As tabelas de decisão aplicam-se melhor a situações em que o número de ações é grande e ocorrem muitas combinações de condições.

Também devemos utilizar tabelas de decisão se existirem dúvidas de que a árvore de decisão não mostra toda a complexidade do problema.

formato básico de uma tabela de decisão é mostrado na figura 5.43.

Nome da Tabela	
Condições	Combinações
Ações	Regras

Figura 5.43 – Formato básico de uma Tabela de Decisão. A construção de uma tabela de decisão envolve os seguintes passos:

- Levantar as ações do processo;
- Identificar as condições que determinam estas ações;
- Identificar os estados possíveis de cada condição;
- Identificar as combinações dos estados das condições;
- Construir uma coluna para cada combinação de condições;
- Preencher cada coluna com as regras das ações correspondentes;
- Verificar se o entendimento foi correto;
- Alterar a tabela até obter total concordância dos usuários;

Se possível, compactar a tabela.

Em função do tipo das condições, temos dois tipos de tabelas:

Tabela de Entrada Limitada: os valores de uma condição se limitam a dois. Exemplos típicos deste tipo de tabelas são as tabelas cujas condições são escritas sob a forma de perguntas, de modo que as respostas sejam “sim” ou “não”, como mostra o exemplo da figura 5.44.

Tratamento de Clientes								
Volume de Negócios ≥ R\$ 1 milhão?	S	S	S	S	N	N	N	N
Atraso de pagamento registrado?	N	N	S	S	N	N	S	S
Tempo de trabalho ≥ 20 anos?	S	N	S	N	S	N	S	N
Tratamento Prioritário	X	X	X					
Tratamento Normal				X	X	X	X	X

Figura 5.44 – Tabela de Entrada Limitada.

Tabela de Entrada Ampliada: Uma condição pode ter mais de dois valores possíveis diferentes, como no exemplo da figura 5.45.

Cobrança de Fretes												
Meio de Transporte	F	F	F	F	R	R	R	R	M	M	M	M
Tipo de Entrega	R	R	N	N	R	R	N	N	R	R	N	N
Peso	L	P	L	P	L	P	L	P	L	P	L	P
R\$ 100/Kg					X				X			
R\$ 50/Kg	X					X	X			X	X	
R\$ 10/Kg		X	X	X				X				X

Meio de Transporte: Ferroviário (F), Rodoviário (R), Marítimo (M). Tipo de Entrega: Rápida (R) – até 5 dias úteis; Normal (N) – até 30 dias. Peso: Leve (L): ≤ 100kg; Pesado (P): > 100Kg

Figura 5.45 – Tabela de Entrada Ampliada.

Muitas vezes, grupos de condições levam à mesma ação. Para estes casos, podemos utilizar tabelas compactadas, como a do exemplo 5.46.

Tratamento de Clientes				
Volume de Negócios ≥ R\$ 1 milhão?	S	S	S	N
Atraso de pagamento registrado?	N	S	S	-
Tempo de trabalho ≥ 20 anos?	-	S	N	-
Tratamento Prioritário	X	X		
Tratamento Normal			X	X

Figura 5.46 – Tabela Compactada.

Quando uma única tabela se tornar muito grande ou complexa, podemos utilizar tabelas encadeadas, onde uma tabela faz referência a outras, como mostra o exemplo da figura 5.47.

Tratamento de Clientes - 1		
Volume de Negócios ≥ R\$ 1 milhão?	S	N
Tratamento de Clientes - 2	X	
Tratamento Normal		X

Tratamento de Clientes - 2				
Atraso de pagamento registrado?	N	N	S	S
Tempo de trabalho ≥ 20 anos?	S	N	S	N
Tratamento Prioritário	X	X	X	
Tratamento Normal				X

Figura 5.47 – Tabelas Encadeadas.

### Modelagem Funcional Com DFDs E A Análise Essencial

Quando empregamos a filosofia da Análise Essencial na modelagem funcional, um DFD contendo um único processo é construído para cada um dos eventos listados na lista de eventos.

Caso o evento seja complexo demais e mereça ser decomposto em outros processos, então as técnicas de fissão ou explosão devem ser aplicadas. Construídos os DFDs para os eventos específicos, os mesmos podem ser agrupados, dando origem a DFDs de nível superior, até se chegar a um DFD de nível 0 e, por fim, a um DFD de Contexto.

Contudo, é importante ressaltar que a maior parte dos eventos em uma lista de eventos pode ser simples e que representar tais eventos por meio de DFDs pode não trazer ganhos substanciais para o desenvolvimento.

Muito pelo contrário: pode gerar uma quantidade desnecessária de DFDs, aumentando muito a documentação do sistema, com pouca utilidade. Assim, recomendamos a elaboração de DFDs apenas para os eventos da lista que sejam mais complexos e que estejam intimamente ligados ao propósito do sistema. Para os demais, as descrições dos eventos providas no modelo ambiental são suficientes.

#### Projeto

projeto de software encontra-se no núcleo técnico do processo de desenvolvimento de software e é aplicado independentemente do modelo de ciclo de vida e paradigma adotados. É iniciado assim que os requisitos do software tiverem sido modelados e especificados, correspondendo à primeira dentre as três atividades técnicas – projeto, implementação e testes – requeridas para se construir e verificar um sistema de software.

Enquanto a atividade de análise pressupõe que dispomos de tecnologia perfeita (capacidade ilimitada de processamento com velocidade instantânea, capacidade ilimitada de armazenamento, custo zero e não passível de falha), a atividade de projeto envolve a modelagem de como o sistema será implementado, com a adição dos requisitos não funcionais aos modelos construídos na análise, como ilustra a figura 6.1.

Assim, o objetivo do projeto é incorporar a tecnologia aos requisitos essenciais do usuário, projetando o que será construído na implementação. Para tal, é necessário conhecer a tecnologia disponível e as facilidades do ambiente de software no qual o sistema será implementado.

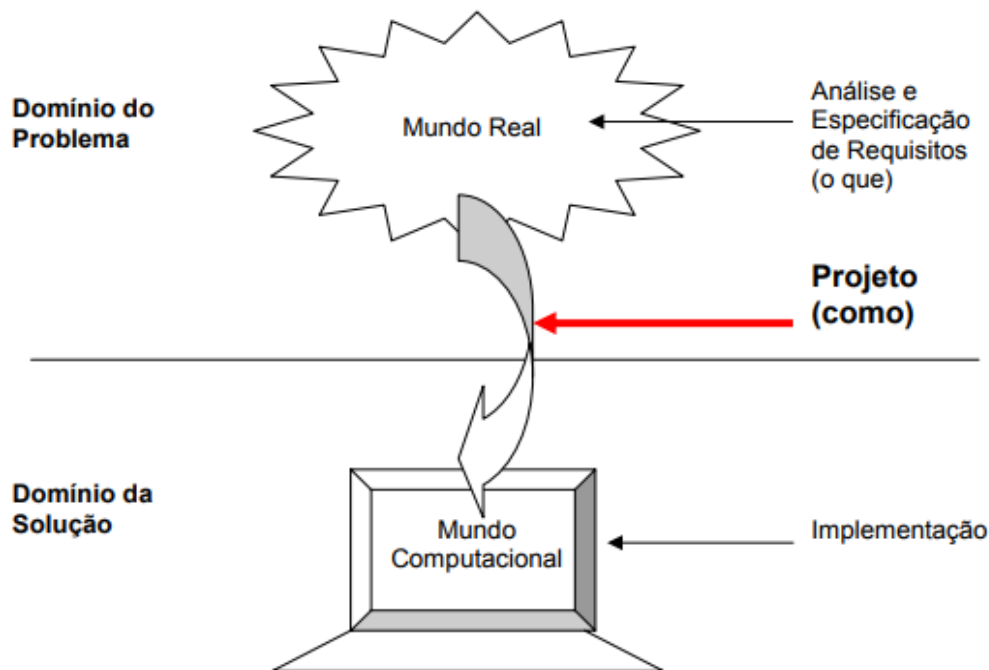


Figura 6.1 – Visão Geral da Atividade de Projeto.

projeto de software é um processo iterativo. Inicialmente, o projeto é representado em um nível alto de abstração. À medida que iterações ocorrem, os refinamentos conduzem a representações de menores níveis de abstração.

Uma especificação de projeto deve:

Contemplar todos os requisitos explícitos contidos no modelo de análise e todos os requisitos implícitos desejados pelo cliente;

Ser um guia legível e compreensível para aqueles que irão codificar, testar e manter o software.

Prover um quadro completo do software, tratando aspectos funcionais, comportamentais e de dados, segundo uma perspectiva de implementação.

No projeto de sistemas, um modelo de projeto é tipicamente gerado a partir dos modelos de análise, com o objetivo de representar o que deverá ser codificado na fase de implementação. Independentemente do paradigma adotado, o projeto deve produzir:

**Projeto da Arquitetura do Software:** visa a definir os grandes componentes estruturais do software e seus relacionamentos.

**Projeto de Dados:** tem por objetivo projetar a estrutura dos dados necessária para implementar o software.

**Projeto de Interfaces:** descreve como o software deverá se comunicar dentro dele mesmo (interfaces internas), com outros sistemas (interfaces externas) e com pessoas que o utilizam (interface com o usuário).

**Projeto Procedimental:** tem por objetivo refinar e detalhar a descrição procedimental dos componentes estruturais da arquitetura do software.

A seguir, cada uma dessas sub-atividades do projeto de sistemas é discutida à luz do paradigma estruturado.

### Projeto de Dados



Um aspecto fundamental da fase de projeto consiste em estabelecer de que forma serão armazenados os dados do sistema. Em função da plataforma de implementação, diferentes soluções de projeto devem ser adotadas. Isto é, se o software tiver de ser implementado em um banco de dados hierárquico, por exemplo, um modelo hierárquico deve ser produzido, adequando a modelagem de entidades e relacionamentos a essa plataforma de implementação.

Atualmente, a plataforma mais difundida para armazenamento de dados é a dos Bancos de Dados Relacionais e, portanto, neste texto, discutiremos apenas o projeto lógico de bancos de dados relacionais.

Em um modelo de dados relacional, os conjuntos de dados são representados por tabelas de valores. Cada tabela é bidimensional, sendo organizada em linhas e colunas.

Para se realizar o mapeamento de um modelo de entidades e relacionamentos em um modelo relacional, pode-se utilizar como ponto de partida as seguintes diretrizes:

Entidades e agregados devem dar origem a tabelas;

Uma instância de uma entidade ou de um agregado deve ser representada como uma linha da tabela correspondente;

Um atributo de uma entidade ou agregado deve ser tratado como uma coluna da tabela correspondente;

Toda tabela tem de ter uma chave primária, que pode ser um atributo determinante do conjunto de entidades ou agregado correspondente, ou uma nova coluna criada exclusivamente para este fim;

Relacionamentos devem ser mapeados através da transposição da chave primária de uma tabela para a outra.

Ainda que esse mapeamento seja amplamente aplicável, é sempre necessário avaliar requisitos não funcionais para se chegar ao melhor projeto para uma dada situação. Além disso, os relacionamentos requerem um cuidado maior e, por isso, são tratados a seguir com mais detalhes.

Relacionamentos 1 : 1

No exemplo da figura 6.2, ambas as soluções são igualmente válidas. Deve-se observar para cada caso, contudo, a melhor solução, considerando os seguintes aspectos:

Se A for total em R (todo A está associado a um B), é melhor colocar a chave de B (#B) em A, como mostra o exemplo da figura 6.3.

Se B for total em R (todo B está associado a um A), é melhor colocar a chave de A (#A) em B.

Se ambos forem totais, pode-se trabalhar com uma única tabela escolhendo uma das chaves #A ou #B como chave primária, como mostra o exemplo da figura 6.4.

Caso contrário, é melhor transpor a chave que dará origem a uma coluna mais densa, isto é, que terá menos valores nulos.

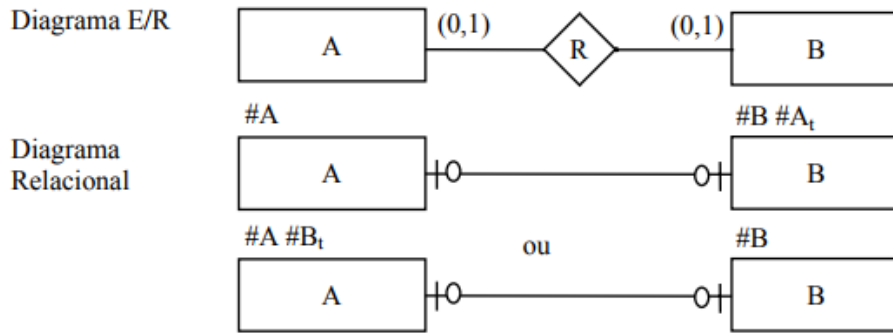


Figura 6.2 - Tradução de Relacionamentos 1:1 do Diagrama E/R para o Relacional.

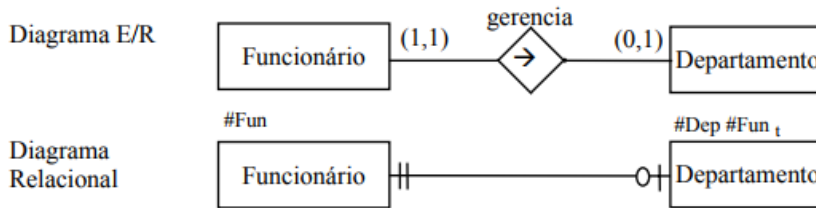
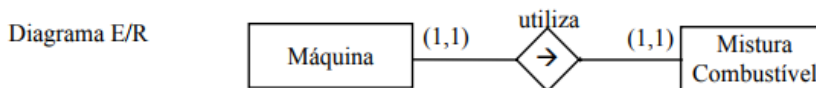


Figura 6.3 – Exemplo de Relacionamento 1:1.



Uma máquina emprega necessariamente uma mistura combustível e vice-versa.

**Diagrama Relacional**

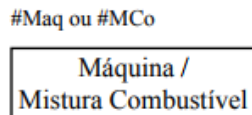


Figura 6.4 - Exemplo de um relacionamento 1:1 total em ambos os lados.

**Relacionamentos 1 : N**

Neste caso, deve-se transpor a chave da tabela correspondente à entidades de cardinalidade máxima N para a tabela que representa a entidade cuja cardinalidade máxima é 1, como mostra a figura 6.5.

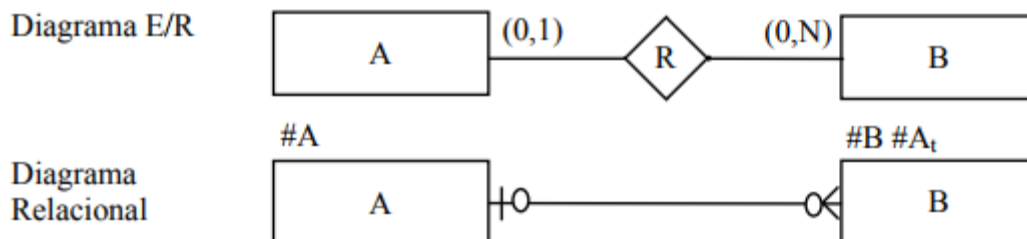


Figura 6.5 - Tradução de Relacionamentos 1:N do Diagrama E/R para o Relacional.

Um A pode estar associado a vários Bs, mas um B só pode estar associado a um A, logo se deve transpor a chave primária de A para B. A figura 6.6 mostra um exemplo desta situação.

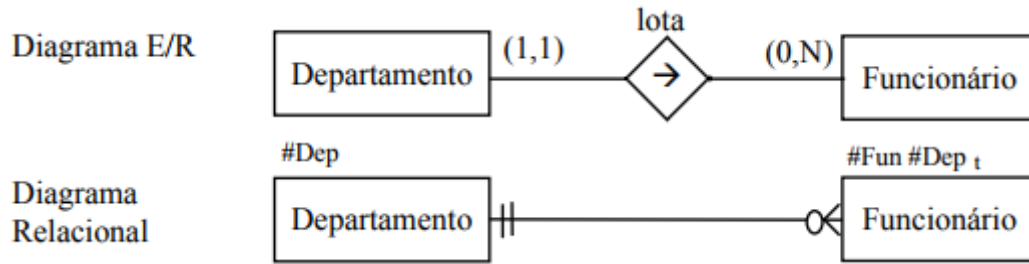


Figura 6.6 – Exemplo de um relacionamento 1:N.

**Relacionamentos N : N**

No caso de relacionamentos N:N (agregados ou não), deve-se criar uma terceira tabela, transpondo as chaves primárias das duas tabelas que participam do relacionamento N:N, como mostra a figura 6.7. Se existirem atributos do relacionamento (agregado), esses deverão ser colocados na nova tabela. Caso seja necessário, algum desses atributos pode ser designado para compor a chave primária da tabela correspondendo ao agregado, como ilustra o exemplo da figura 6.8.

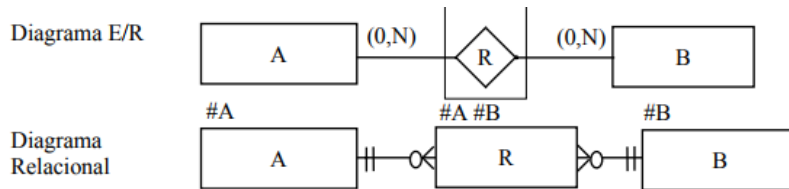


Figura 6.7 - Tradução de Relacionamentos N:N do Diagrama E/R para o Relacional.

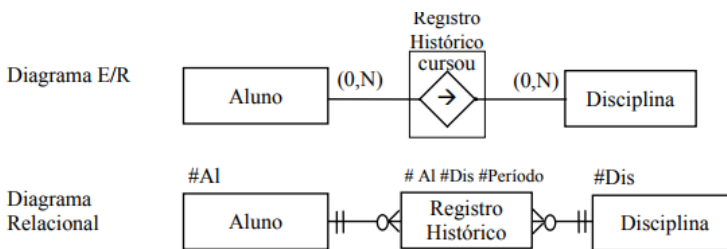


Figura 6.8 – Exemplo de relacionamento N:N.

**Auto-Relacionamentos**

Os auto-relacionamentos devem seguir as mesmas regras de tradução de relacionamentos, como mostram os exemplos das figuras 6.9 e 6.10.

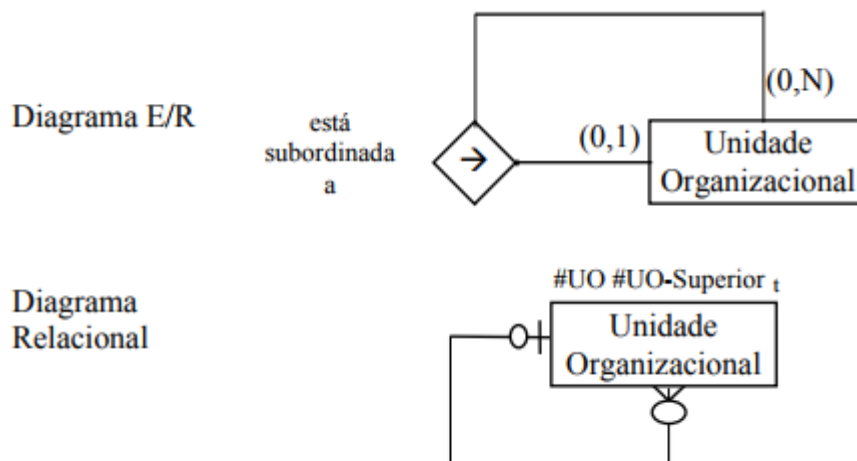


Figura 6.9 – Exemplo de auto-relacionamento 1:N.

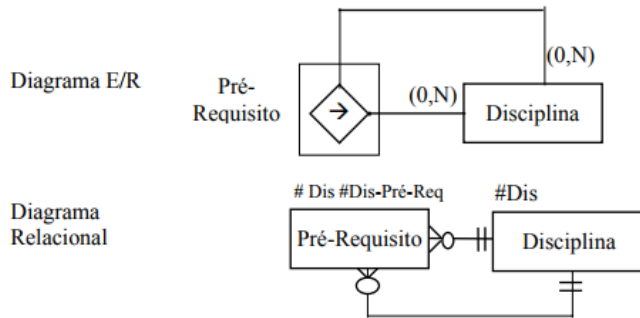


Figura 6.10 – Exemplo de auto-relacionamentos N:N.

Relacionamentos entre uma Entidade e um Agregado

Já discutimos como fazer a tradução de um agregado para o modelo relacional. Um relacionamento entre uma entidade e um agregado terá o mesmo tratamento que um relacionamento entre entidades, considerando, agora, o agregado como uma entidade. Tomemos como exemplo um relacionamento 1:N entre uma entidade e um agregado, como mostra o exemplo da figura 6.11.

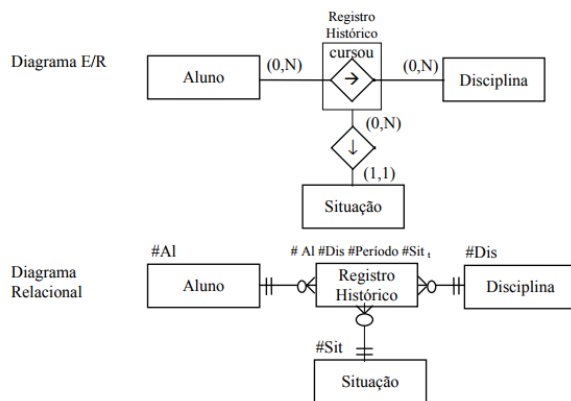


Figura 6.11 – Exemplo de relacionamento 1:N entre uma entidade e um agregado.

Relacionamento Ternário

No caso de relacionamentos ternários, deve-se criar uma nova tabela contendo as chaves das três entidades envolvidas, como mostra a figura 6.12. Assim como no caso de agregados, se existirem atributos do relacionamento, esses deverão ser colocados na nova tabela. Caso seja necessário, algum desses atributos pode ser designado para compor a chave primária da nova tabela.

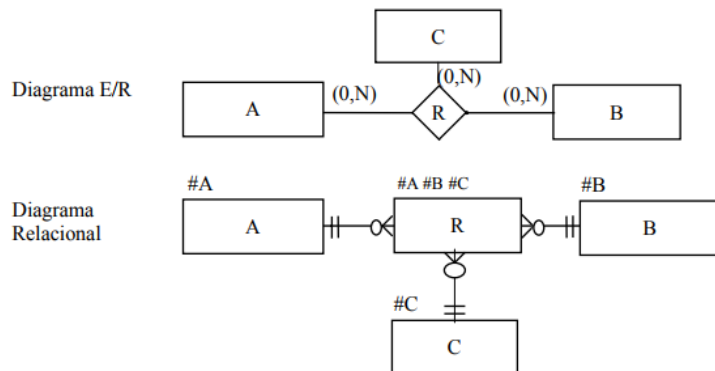


Figura 6.12 - Tradução de Relacionamentos Ternários.

**Particionamento**

No caso de particionamento de conjuntos de entidades, deve-se criar uma tabela para o super-tipo e tantas tabelas quantos forem os sub-tipos, todos com a mesma chave, como mostra a figura 6.13. Caso não haja no modelo conceitual um atributo determinante no super-tipo, uma chave primária deve ser criada para fazer a amarração com os sub-tipos.

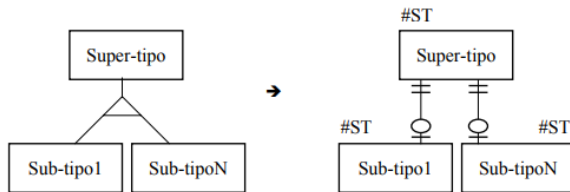


Figura 6.13 – Tradução de Particionamento.

**Atributos Multivalorados**

Segundo a propriedade do modelo relacional que nos diz que cada célula de uma tabela pode conter no máximo um único valor, não podemos representar atributos multivalorados como uma única coluna da tabela. Há algumas soluções possíveis para este problema, tal como, criar tantas colunas quantas necessárias para representar o atributo. Essa solução, contudo, pode, em muitos casos, não ser eficiente ou mesmo possível. Uma solução mais geral para este problema é criar uma tabela em separado, como mostra a figura 6.14.

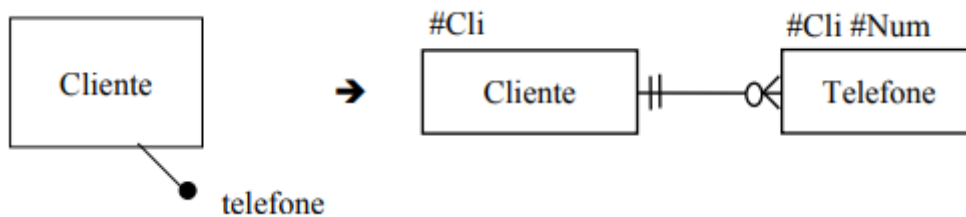


Figura 6.14 – Mapeamento Geral de Atributos Multi-valorados.

**Projeto De Interface Com O Usuário**

A maioria dos sistemas atuais é desenvolvida para ser utilizada por pessoas. Assim, um aspecto fundamental no projeto de sistemas é a interface com o usuário (IU). Nessa etapa do projeto, são definidos os formatos de janelas e relatórios, entre outros, sendo a prototipagem bastante utilizada, buscando auxiliar o desenvolvimento e a seleção dos mecanismos reais de interação. A IU capta como um usuário comandará o sistema e como o sistema apresentará as informações a ele.

Princípio básico para o projeto de interfaces com o usuário é o seguinte: “Conheça o usuário e as tarefas”. O projeto de interface com o usuário envolve não apenas aspectos de tecnologia (facilidades para interfaces gráficas, multimídia, etc), mas principalmente o estudo das pessoas. Quem é o usuário? Como ele aprende a interagir com um novo sistema? Como ele interpreta uma informação produzida pelo sistema? O que ele espera do sistema? Essas são apenas algumas das muitas questões que devem ser levantadas durante o projeto da interface com o usuário [1]. De maneira geral, o projeto de interfaces com o usuário segue o seguinte processo global, como mostra a figura 6.15:

Delinear as tarefas necessárias para obter a funcionalidade do sistema: este passo visa capturar as tarefas que as pessoas fazem normalmente no contexto do sistema e mapeá-las em um conjunto similar (mas não necessariamente idêntico) de tarefas a serem implementadas no contexto da interface homem-máquina.

Estabelecer o perfil dos usuários: A interface do sistema deve ser adequada ao nível de habilidade dos seus futuros usuários. Assim, é necessário estabelecer o perfil dos potenciais usuários e classificá-los segundo aspectos como nível de habilidade, nível na organização e membros em diferentes grupos. Uma classificação possível considera os seguintes grupos:

Usuário Novato: não conhece os mecanismos de interação requeridos para utilizar a interface eficientemente (não está habituado a usar computadores ou mecanismos específicos de interação com os sistemas computacionais) e conhece pouco a aplicação em si, isto é, entende pouco as funções e objetivos do sistema (semântica da aplicação);

Instruído, mas intermitente: possui um conhecimento razoável da semântica da aplicação, mas tem relativamente pouca lembrança das informações necessárias para utilizar bem a interface;

Instruído e frequente: possui bom conhecimento da aplicação e domina bem os mecanismos de interação. Geralmente, usuários desse tipo buscam atalhos e modos abreviados de interação.

Considerar aspectos gerais de projeto de interface, tais como tempo de resposta, facilidades de ajuda, mensagens de erro, tipos de comandos, entre outros.

Construir protótipos e, em última instância, implementar as interfaces do sistema, usando ferramentas apropriadas. A prototipagem abre espaço para uma abordagem iterativa de projeto de interface com o usuário, como mostra abaixo. Entretanto, para tal é imprescindível o suporte de ferramentas para a construção de interfaces, provendo facilidades para manipulação de janelas, menus, botões, comandos, etc...

Avaliar o resultado: Coletar dados qualitativos e quantitativos (questionários distribuídos aos usuários do protótipo).

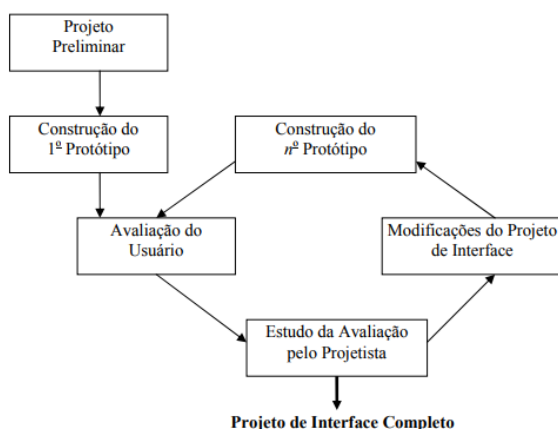


Figura 6.15 - Abordagem Iterativa para o Projeto de Interface com o Usuário.

### Projeto Modular De Programas

A tarefa de construção de sistemas computadorizados requer uma organização das idéias, de modo a se conseguir desenvolver produtos com qualidade. Programas escritos sem qualquer subdivisão são inviáveis do ponto de vista administrativo e não permitem reaproveitamento de trabalhos anteriormente executados.

Projeto Modular de Programas oferece uma coleção de orientações, técnicas, estratégicas e heurísticas capazes de conduzir a bons projetos de programas. O objetivo é desenvolver programas com menor complexidade, usando o princípio “dividir para conquistar”. Como resultados de um bom projeto de programas, tem-se:

Facilidade na leitura de programas (maior legibilidade);

Maior rapidez na depuração de programas na fase de testes;

Facilidade de modificação de programas na fase de manutenção.

Projeto estruturado de sistemas, em sua dimensão de funções, considera que o projeto de programas envolve duas grandes etapas: o projeto da arquitetura do sistema e o projeto detalhado dos módulos. Em ambos os casos, técnicas de Projeto Modular de Programas são empregadas. Apesar de usar

diferentes variações para o projeto arquitetural e para o projeto detalhado, basicamente, dois conceitos são centrais para o projeto estruturado de sistemas:

**Módulo:** Conjunto de instruções que desempenha uma função específica dentro de um programa. É definido por: entrada / saída, função, lógica e dados internos.

**Conexão entre Módulos:** Indica a forma como os módulos interagem entre si.

Bloco básico de construção de um programa estruturado é, portanto, um módulo. Assim, os modelos do projeto estruturado de programas são organizados como uma hierarquia de módulos. A idéia básica é estruturar os programas em termos de módulos e conexões entre esses módulos.

Projeto Modular de Programas considera, ainda, alguns aspectos importantes para o projeto de programas:

Procura solucionar sistemas complexos através da divisão do sistema em “caixas pretas” (os módulos) e pela organização dessas “caixas pretas” em uma hierarquia conveniente para uma implementação computadorizada.

Utiliza ferramentas gráficas, o que tornam mais fácil a compreensão.

Oferece um conjunto de estratégias para desenvolver o projeto de solução a partir de uma declaração bem definida do problema.

Oferece um conjunto de critérios para avaliação da qualidade de um determinado projeto-solução com respeito ao problema a ser resolvido.

São objetivos do Projeto Modular de Programas:

Permitir a construção de programas mais simples;

Obter módulos independentes;

Permitir testes por partes;

Ter menos código a analisar em uma manutenção;

Servir de guia para a programação estruturada;

Construir módulos com uma única função;

Permitir reutilização.

Projeto Modular procura simplificar um sistema complexo, dividindo-o em módulos e organizando esses hierarquicamente. O sistema é subdividido em caixas-pretas, que são organizadas em uma hierarquia conveniente. A vantagem do uso da caixa-preta está no fato de que não precisamos conhecer como ela trabalha, mas apenas utilizá-la. As características de uma caixa-preta são:

Sabemos como devem ser os elementos de entrada, isto é, as informações necessárias para seu processamento;

Sabemos como devem ser os elementos de saída, isto é, os resultados oriundos do seu processamento;

Conhecemos a sua função, isto é, que processamento ela faz sobre os dados de entrada para que sejam produzidos os resultados;

Não precisamos conhecer como ela realiza as operações, nem tampouco seus procedimentos internos, para podermos utilizá-la.

Sistemas compostos por caixas pretas são facilmente construídos, testados, corrigidos, entendidos e modificados. Desse modo, o primeiro passo no controle da complexidade no projeto estruturado consiste em dividir um sistema em módulos, de modo a atingir as seguintes metas:

Cada módulo deve resolver uma parte bem definida do problema;

A função de cada módulo deve ser facilmente compreendida;

Conexões entre módulos devem refletir apenas conexões entre partes do problema;

As conexões devem ser tão simples e independentes quanto possível.

**Organizando Módulos Hierarquicamente**

Antes de iniciarmos uma discussão sobre Projeto Modular de Programas, passemos a observar os exemplos das figuras 6.16, 6.17 e 6.18, que mostram três organogramas de empresas

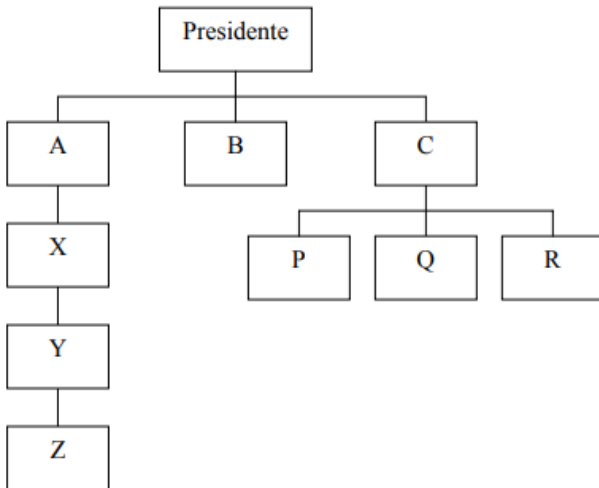


Figura 6.16 - Organograma da Empresa 1.

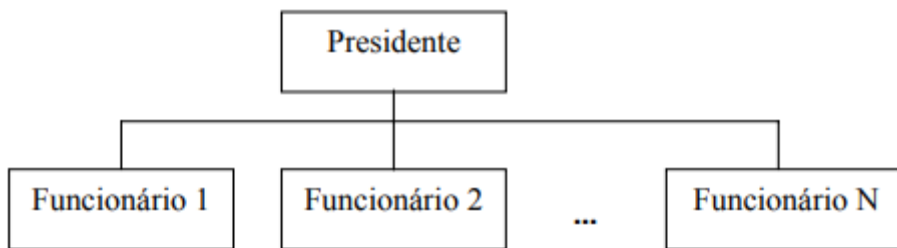


Figura 6.17 - Organograma da Empresa 2.

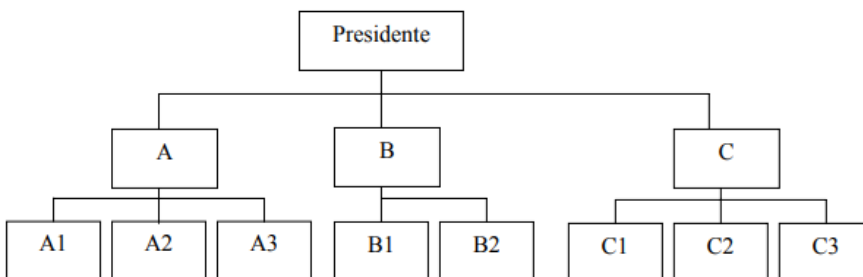


Figura 6.18 - Organograma da Empresa 3.

Como podemos notar, no organograma da empresa 1, o vice-presidente A e os gerentes X e Y, possuem tarefas triviais, pois cada um deles tem como responsabilidade gerenciar apenas um subordinado. Neste caso, todo serviço seria realizado pelo funcionário Z. Poderíamos sugerir, então, acabar com as gerências. Por outro lado, o presidente da empresa 2 está sobrecarregado, uma vez que ele



gerencia funcionários demais. A empresa 3 parece apresentar um organograma mais equilibrado, no qual cada gerente gerencia um número apropriado de subordinados.

As estruturas de um programa, ou de um sistema, podem ser discutidas de maneira análoga à questão dos organogramas. Ou seja, os módulos devem ser dispostos em uma hierarquia, de modo a, por um lado, não provocar sobrecarga de processamento e, de outro, não criar módulos apenas intermediários, sem desempenhar nenhuma função.

Há vários tipos de diagramas hierárquicos para o projeto de programas. Neste texto, serão explorados dois deles: o Diagrama Hierárquico de Funções (DHF), usado principalmente para o projeto arquitetural, e o Diagrama de Estrutura Modular (DEM), usado para o projeto detalhado de módulos.

A diferença básica entre eles é que o DHF não representa o fluxo de dados e controles entre módulos, nem aspectos relacionados com detalhes lógicos de um módulo, tais como estruturas de repetição (laços) e condição. Essas informações são capturadas em um DEM e, por isso mesmo, o DEM é empregado no projeto detalhado de módulos, enquanto o DHF é usado para o projeto da arquitetura do sistema.

### **Diagrama Hierárquico de Funções**

Um Diagrama Hierárquico de Funções (DHF) define a arquitetura global de um programa ou sistema, mostrando módulos e suas inter-relações. Cada módulo pode representar um subsistema, programa ou módulo de programa. Sua finalidade é mostrar os componentes funcionais gerais (arquitetura do sistema) e fazer referência a diagramas detalhados (tipicamente Diagramas de Estrutura Modular). Um DHF não mostra o fluxo de dados entre componentes funcionais ou qualquer informação de estruturas de controle, tais como laços (loops) ou condições.

A estrutura de um DHF tem como ponto de partida um módulo inicial, localizado no topo da hierarquia, que detém o controle sobre os demais módulos do diagrama, ditos seus módulos-filhos. Um módulo-filho, por sua vez, pode ser “pai” de outros módulos, indicando que ele detém o controle sobre esses módulos.

A construção de um DHF deve procurar espelhar a estrutura do negócio que o sistema está tratando. A descrição do escopo, com sua subdivisão em sub-sistemas e módulos, e a lista de eventos e descrições associadas devem ser a base para a construção dos DHFs.

Cada executável deve dar origem a um DHF. As funcionalidades controladas por esse executável devem ser tratadas como módulos-filhos do módulo inicial do diagrama. Funções menores que compõem uma macro-função podem ser representadas como módulos-filhos do módulo correspondente. Para sistemas de médio a grande porte, contudo, representar todas as funcionalidades em um único diagrama pode torná-lo muito complexo. Assim, novos DHFs podem ser elaborados para agrupar certas funcionalidades. Tomemos como exemplo um sistema de entrega em domicílio de refeições, cujo escopo é o seguinte:

Subsistema Controle de Cardápio, envolvendo macro-funções para: Cadastrar Refeições, Sobremesas e Bebidas. Cada uma dessas macro-funções teria funções para incluir, excluir, alterar e consultar esses diferentes tipos de itens de cardápio;

Subsistema Atendimento a Clientes, envolvendo macro-funções para Cadastrar Cliente, Controlar Pedido e Consultar Cardápio. Assim como os demais cadastros, o cadastro de clientes teria funções para incluir um novo cliente, alterar dados de cliente, consultar e excluir clientes. Já o controle de pedidos envolveria funções para efetuar um novo pedido, alterar dados de pedido, cancelar pedido, definir entregador e registrar atendimento de pedido. Por fim a consulta ao cardápio teria funções para consultar refeições, sobremesas e bebidas.

Com base nesse escopo e considerando que cada subsistema deve ser implementado como uma aplicação executável, poderíamos construir o DHF mostrado na figura 6.19. Nesse diagrama, optou-se por não representar os módulos-filhos do módulo Controlar Pedido, uma vez que ele é bastante complexo, com vários sub-módulos, o que traria uma complexidade indesejada para o DHF. Assim, além do diagrama da figura 6.19, um outro, cujo módulo inicial seria Controlar Pedido, deveria ser elaborado.

Vale ressaltar que um DHF pode ser usado como um guia para o projeto das interfaces com o usuário, apoiando a definição de janelas, estruturas de menu, etc.

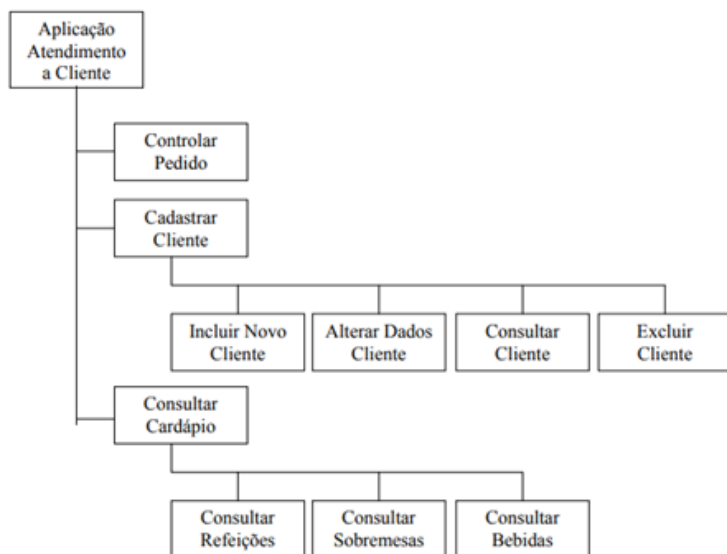


Figura 6.19 – DHF para o subsistema de Atendimento a Clientes do Sistema de Entrega de Refeições.

#### Diagrama De Estrutura Modular

Em um Diagrama de Estrutura Modular (DEM), um programa é representado como um conjunto de módulos organizados hierarquicamente, de modo que os módulos que executam tarefas de alto nível no programa são colocados nos níveis superiores da hierarquia, enquanto os módulos que executam tarefas detalhadas, de nível mais baixo, aparecem nos níveis inferiores. Observando a hierarquia, os módulos a cada nível sucessivo contêm tarefas que definem as tarefas realizadas no nível precedente.

Um módulo é definido como uma coleção de instruções de programa com quatro atributos básicos: entradas e saídas, função, lógica e dados internos. Entradas e saídas são, respectivamente, as informações que um módulo necessita e fornece. A função de um módulo é o que ele faz para produzir, a partir da informação de entrada, os resultados da saída. Entradas, saídas e função fornecem a visão externa do módulo e, portanto, apenas esses aspectos são representados no diagrama de estrutura modular.

A lógica de um módulo é a descrição dos algoritmos que executam a função. Dados internos são aqueles referenciados apenas dentro do módulo. Lógica e dados internos representam a visão interna do módulo e são descritos por uma técnica de especificação de programas, tal como português estruturado, tabelas de decisão e árvores de decisão, discutidos no capítulo 5.

Assim sendo, um DEM mostra:

A divisão de um programa em módulos;

A hierarquia e a organização dos módulos;

As interfaces de comunicação entre módulos (entrada/saída);

As funções dos módulos, dadas por seus nomes;

Estruturas de controle entre módulos, tais como condição de execução de um módulo, laços de repetição de módulos (iteração), dentre outras.

Um DEM não mostra a lógica e os dados internos dos módulos e, por isso, deve ser acompanhado de uma descrição dos módulos, mostrando os detalhes internos dos procedimentos das caixas pretas.

#### Notação Utilizada na Elaboração de DEMs

A seguir, são apresentadas as principais notações utilizadas para elaborar Diagramas de Estrutura Modular:

**Módulo:** Em um DEM, um módulo é representado por um retângulo, dentro do qual está contido seu nome, como mostra a figura 6.20. Um módulo pré-definido é aquele que já existe em uma biblioteca de módulos e, portanto, não precisa ser descrito ou detalhado.

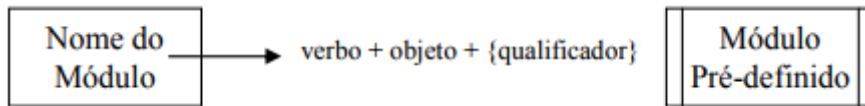


Figura 6.20 – Simbologia para Módulos em um DEM.

**Conexão entre módulos:** Um sistema é um conjunto de módulos organizados dentro de uma hierarquia, cooperando e se comunicando para realizar um trabalho. A hierarquia mostra “quem chama quem”. Portanto, módulos devem estar conectados. No exemplo da figura 6.21, o módulo A chama o módulo B passando, como parâmetros, os dados X e Y. O módulo B executa, então, sua função e retorna o controle para A, no ponto imediatamente após à chamada de B, passando como resultado o dado Z. A ordem de chamada é sempre de cima para baixo, da esquerda para a direita.

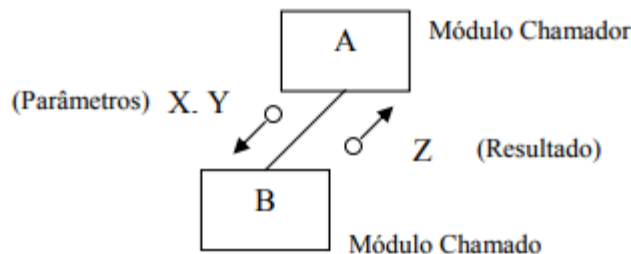


Figura 6.21 – Conexão entre módulos.

**Comunicação entre módulos:** Módulos conectados estão se comunicando, logo existem informações trafegando entre eles. Estas informações podem ser dados ou controles (descrevem uma situação ocorrida durante a execução do módulo). A figura

6.22 mostra a convenção utilizada para se determinar se a informação que está sendo passada entre módulos é um dado ou um controle, juntamente com um exemplo.

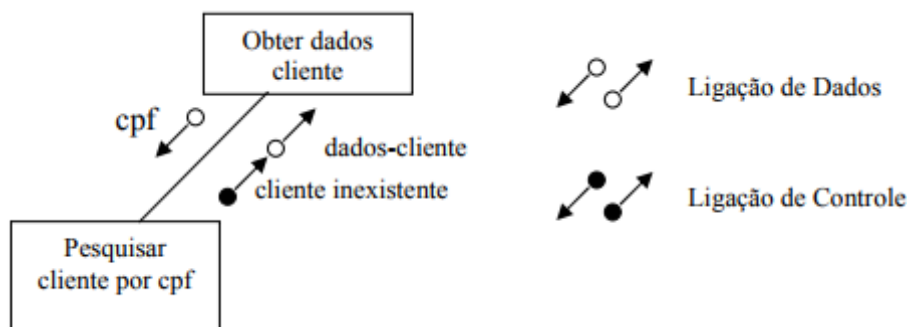


Figura 6.22 – Comunicação entre módulos.

**Chamadas Condicionais:** Em muitos casos, um módulo só será ativado se uma condição for satisfeita. Nestes casos, temos chamadas condicionais, cuja notação é mostrada na figura 6.23. No exemplo à esquerda, o módulo A pode ou não chamar o módulo B. No exemplo à direita, o módulo A pode chamar um dos módulos B ou C.

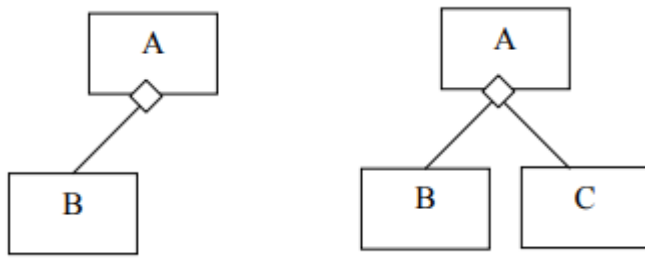
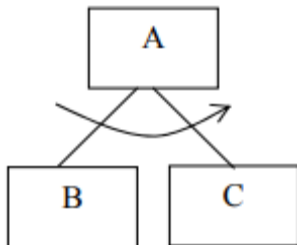


Figura 6.23 – Chamada Condicional.

Chamadas Iterativas: Algumas vezes, nos deparamos com situações nas quais um módulo (ou um conjunto de módulos) é chamado várias vezes, caracterizando chamadas iterativas ou repetidas, cuja notação é mostrada na figura 6.24. No exemplo, os módulos B e C são chamados repetidas vezes pelo módulo A.



Conectores: Algumas vezes, um mesmo módulo é chamado por mais de um módulo, às vezes em diagramas diferentes. Outras, o diagrama está complexo demais e deseja-se continuá-lo em outra página. Nestas situações, conectores podem ser utilizados, como ilustra a figura 6.25.

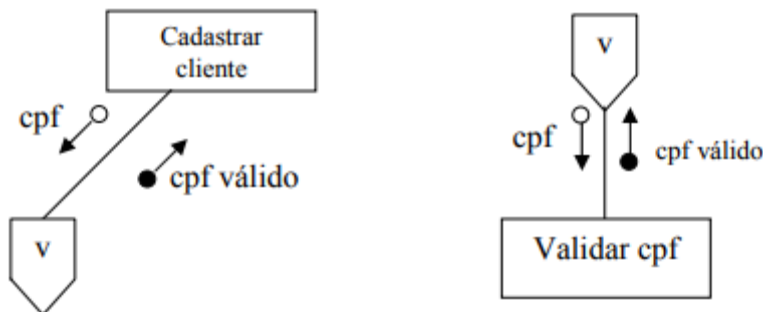


Figura 6.25 – Conectores

### Técnicas de Desenho

Para elaborar um diagrama de estrutura modular, devemos observar as seguintes orientações:

Os módulos devem ser desenhados na ordem de execução, da esquerda para a direita.

Cada módulo só deve aparecer uma única vez no diagrama. Para se evitar cruzamento de linhas, deve-se usar conectores.

Não segmentar demais.

Além dessas orientações, o projeto estruturado fornece duas estratégias de projeto para guiar a elaboração de DEMs: a análise de transformação e a análise de transação. Essas duas estratégias fornecem dois modelos de estrutura que podem ser usados isoladamente ou em combinação para derivar um projeto estruturado.

A análise de transformação é um modelo de fluxo de informações centrado na filosofia entrada-processamento-saída. Assim, o DEM correspondente tende a espelhar esta mesma estrutura, podendo ser decomposto em três grandes ramos, como mostra a figura 6.26.

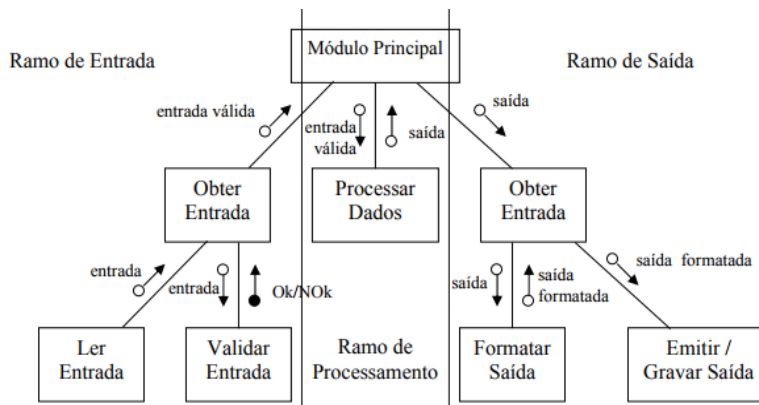
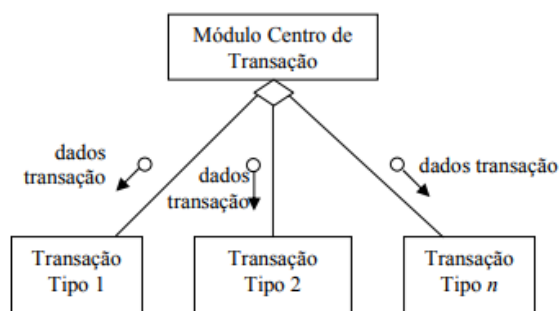


Figura 6.26 – Análise de Transformação

Ramo de entrada contém os módulos que tratam da leitura e validação dos dados de entrada, bem como de uma eventual transformação para um formato adequado para o processamento. O ramo de processamento contém o processamento essencial e deve ser independente de considerações físicas de entrada e saída. Finalmente, o ramo de saída trata da transformação dos dados de saída de um formato interno para um formato adequado para o seu registro (p.ex., uma interface com o usuário ou um registro em bancos de dados).

A análise de transação é uma estratégia de projeto alternativa para a análise de transformação. Ela é útil no projeto de programas de processamento de transações. O DEM geral para a análise de transação é mostrado na figura 6.27. No topo do diagrama está um módulo centro de transação, que é responsável pela determinação do tipo de transação e pela chamada do módulo de transação apropriado. Abaixo dele, estão os vários módulos de transação. Há um módulo de transação para cada tipo de transação.



**Critérios De Qualidade De Projetos De Programa**

Objetivo maior do projeto modular de programas é permitir que um sistema complexo seja dividido em módulos simples. No entanto, é vital que essa partição seja feita de tal forma que os módulos sejam tão independentes quanto possível e que cada um deles execute uma única função. Critérios que tratam desses aspectos são, respectivamente, acoplamento e coesão.

Acoplamento diz respeito ao grau de interdependência entre dois módulos. O objetivo é minimizar o acoplamento, isto é, tornar os módulos tão independentes quanto possível. Um baixo acoplamento pode ser obtido:

Eliminando relações desnecessárias;

Enfraquecendo a dependência das relações necessárias. Podemos citar como razões para se minimizar o acoplamento:

Quanto menos conexões houver entre dois módulos, menor será a chance de um problema ocorrido em um deles se refletir em outros.

Uma alteração deve afetar o menor número de módulos possível, isto é, uma alteração em um módulo não deve implicar em alterações em outros módulos.

Ao dar manutenção em um módulo, não devemos nos preocupar com detalhes de codificação de outros módulos.

Acoplamento envolve três aspectos principais: tipo da conexão, tamanho da conexão e o que é comunicado através da conexão. O tipo da conexão diz respeito à forma como uma conexão é estabelecida. O ideal é que a comunicação se dê através de chamadas a módulos, cada um deles fazendo uso apenas de variáveis locais. Qualquer informação externa necessária deve ser passada como parâmetro. Assim, cada módulo deve possuir seu escopo próprio de variáveis, evitando-se utilizar uma variável definida em outro módulo.

Com relação ao tamanho da conexão, quanto menor o número de informações trafegando de um módulo para outro, menor será o acoplamento. Entretanto, vale a pena ressaltar que é importante manter-se a clareza da conexão. Não devemos mascarar as informações que fluem.

Finalmente, no que tange ao que é comunicado entre módulos, o ideal é que se busque acoplamento apenas de dados. Entretanto, quando se fizer necessária a comunicação de fluxos de controle, devemos fazê-la sem máscaras. Seja o exemplo da figura 6.28.

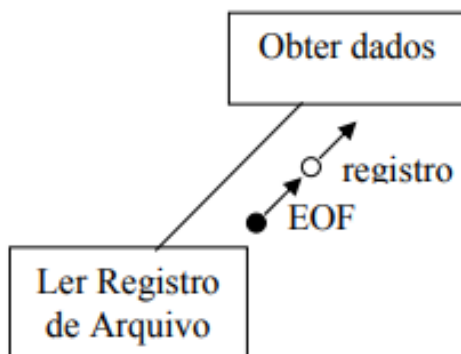


Figura 6.28 – Clareza na comunicação.

Neste caso, não é indicado mover brancos para o registro e se o registro estiver em branco é porque acabou o arquivo (EOF). Com esse artifício, estar-se-ia mascarando o fluxo de controle.

De maneira geral, não adianta melhorar dois desses aspectos se estivermos piorando o terceiro. Muitas vezes, o acoplamento resultante poderá ser maior. Só devemos fazer alterações que melhorem um dos aspectos sem afetar os demais. As seguintes orientações podem servir para apoiar a tomada de decisão:

O módulo que chama não deve nunca enviar um controle ao módulo chamado: isso significa que o módulo que chama está dizendo o que o módulo chamado deve fazer, caracterizando, portanto, que o módulo chamado não trata de uma única função.

Só utilizar fluxos de controle de baixo para cima: O módulo chamado avisa que não conseguiu executar sua função, mas não deve dizer ao chamador o que fazer.

Evitar o uso de variáveis globais: Sempre que possível, utilizar variáveis locais.

É inadmissível que um módulo se refira a uma parte interna de outro. Em suma, para minimizar o acoplamento, devemos:

Passar o menor número possível de parâmetros e, de preferência, apenas dados. Quando for necessário passar fluxos de controle, fazê-lo apenas de baixo para cima.

Ter pontos únicos de entrada e saída em um módulo.

Sempre que possível, utilizar programas compilados separadamente.

Coesão define como as atividades de um módulo estão relacionadas umas com as outras. Vale a pena ressaltar que coesão e acoplamento são interdependentes e, portanto, uma boa coesão deve nos levar a um pequeno acoplamento. A figura 6.29 procura mostrar este fato.

No projeto modular de programas, os módulos devem ter alta coesão, isto é, seus elementos internos devem estar fortemente relacionados uns com os outros.

grau de coesão de um módulo tem um impacto direto na qualidade do software produzido, sobretudo no que tange a manutenibilidade, legibilidade e capacidade de reutilização. O ideal é que tenhamos apenas coesão funcional, isto é, que todos os elementos de um módulo estejam contribuindo para a execução de uma e somente uma função do sistema.

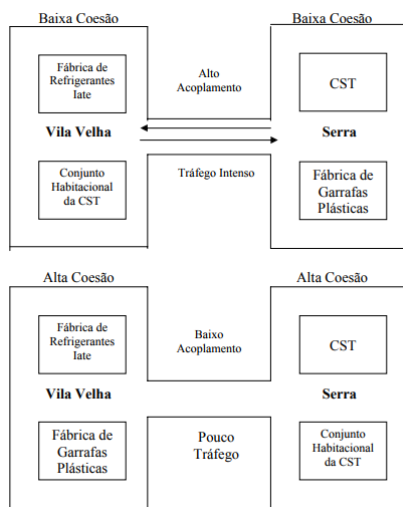


Figura 6.29 – Coesão e Acoplamento.

### Implementação E Testes

Uma vez projetado o sistema, é necessário escrever os programas que implementem esse projeto e testá-los.

#### Implementação

Ainda que um projeto bem elaborado facilite sobremaneira a implementação, essa tarefa não é necessariamente fácil. Muitas vezes, os projetistas não conhecem em detalhes a plataforma de implementação e, portanto, não são capazes de (ou não desejam) chegar a um projeto algorítmico passível de implementação direta. Além disso, questões relacionadas à legibilidade, alterabilidade e reutilização têm de ser levadas em conta.

Deve-se considerar, ainda, que programadores, geralmente, trabalham em equipe, necessitando integrar, testar e alterar código produzido por outros. Assim, é muito importante que haja padrões organizacionais para a fase de implementação. Esses padrões devem ser seguidos por todos os programadores e devem estabelecer, dentre outros, padrões de nomes de variáveis, formato de cabeçalhos de programas e formato de comentários, recuos e espaçamento, de modo que o código e a documentação a ele associada sejam claros para quaisquer membros da organização.

Padrões para cabeçalho, por exemplo, podem informar o que o código (programa, módulo ou componente) faz, quem o escreveu, como ele se encaixa no projeto geral do sistema, quando foi escrito e revisado, apoios para teste, entrada e saída esperadas etc. Essas informações são de grande valia para a integração, testes, manutenção e reutilização.

Além dos comentários feitos no cabeçalho dos programas, comentários adicionais ao longo do código são também importantes, ajudando a compreender como o componente é implementado.

Por fim, o uso de nomes significativos para variáveis, indicando sua utilização e significado, é imprescindível, bem como o uso adequado de recuo e espaçamento entre linhas de código, que ajudam a visualizar a estrutura de controle do programa.

Além da documentação interna, escrita no próprio código, é importante que o código de um sistema possua também uma documentação externa, incluindo uma visão geral dos componentes do sistema, dos diversos grupos de componentes e da inter-relação entre eles.

Ainda que padrões sejam muito importantes, deve-se ressaltar que a correspondência entre os componentes do projeto e o código é fundamental, caracterizando-se como a mais importante questão a ser tratada. O projeto é o guia para a implementação, ainda que o programador deva ter certa flexibilidade para implementá-lo como código.

Como resultado de uma implementação bem-sucedida, as unidades de software devem ser codificadas e critérios de verificação das mesmas devem ser definidos.

### Testes

Uma vez implementado o código de uma aplicação, o mesmo deve ser testado para descobrir tantos defeitos quanto possível, antes da entrega do produto de software ao seu cliente.

Conforme discutido no capítulo 4, o teste é uma atividade de verificação e validação do software e consiste na análise dinâmica do mesmo, isto é, na execução do produto de software com o objetivo de verificar a presença de defeitos no produto e aumentar a confiança de que o mesmo está correto.

Vale ressaltar que, mesmo se um teste não detectar defeitos, isso não quer dizer necessariamente que o produto é um produto de boa qualidade. Muitas vezes, a atividade de teste empregada pode ter sido conduzida sem planejamento, sem critérios e sem uma sistemática bem definida, sendo, portanto, os testes de baixa qualidade.

Assim, o objetivo é projetar testes que potencialmente descubram diferentes classes de erros e fazê-lo com uma quantidade mínima de esforço. Ainda que os testes não possam demonstrar a ausência de defeitos, como benefício secundário, podem indicar que as funções do software parecem estar funcionando de acordo com o especificado.

A idéia básica dos testes é que os defeitos podem se manifestar por meio de falhas observadas durante a execução do software. Essas falhas podem ser resultado de uma especificação errada ou falta de requisito, de um requisito impossível de implementar considerando o hardware e o software estabelecidos, o projeto pode conter defeitos ou o código pode estar errado. Assim, uma falha é o resultado de um ou mais defeitos.

São importantes princípios de testes a serem observados:

Teste completo não é possível, ou seja, mesmo para sistemas de tamanho moderado, pode ser impossível executar todas as combinações de caminhos durante o teste.

Teste envolve vários estágios. Geralmente, primeiro, cada módulo é testado isoladamente dos demais módulos do sistema (teste de unidade). À medida que os testes progridem, o foco se desloca para a integração dos módulos (teste de integração), até se chegar ao sistema como um todo (teste de sistema).

Teste deve ser conduzido por terceiros. Os testes conduzidos por outras pessoas que não aquelas que produziram o código têm maior probabilidade de encontrar defeitos. O desenvolvedor que produziu o código pode estar muito envolvido com ele para poder detectar defeitos mais sutis.

Testes devem ser planejados bem antes de serem realizados. Um plano de testes deve ser utilizado para guiar todas as atividades de teste e deve incluir objetivos do teste, abordando cada tipo (unidade, integração e sistema), como serão executados e quais critérios a serem utilizados para determinar quando o teste está completo. Uma vez que os testes estão relacionados aos requisitos dos clientes e usuários, o planejamento dos testes pode começar tão logo a especificação de requisitos tenha sido elaborada. À medida que o processo de desenvolvimento avança (análise, projeto e implementação), novos testes vão sendo planejados e incorporados ao plano de testes.



processo de teste envolve quatro atividades principais:

**Planejamento de Testes:** trata da definição das atividades de teste, das estimativas dos recursos necessários para realizá-las, dos objetivos, estratégias e técnicas de teste a serem adotadas e dos critérios para determinar quando uma atividade de teste está completa.

**Projeto de Casos de Testes:** é a atividade chave para um teste bem-sucedido, ou seja, para se descobrir a maior quantidade de defeitos com o menor esforço possível. Os casos de teste devem ser cuidadosamente projetados e avaliados para tentar se obter um conjunto de casos de teste que seja representativo e envolva as várias possibilidades de exercício das funções do software (cobertura dos testes). Existe uma grande quantidade de técnicas de teste para apoiar os testadores a projetar casos de teste, oferecendo uma abordagem sistemática para o teste de software.

**Execução dos testes:** consiste na execução dos casos de teste e registro de seus resultados.

**Avaliação dos resultados:** detectadas falhas, os defeitos deverão ser procurados. Não detectadas falhas, deve-se fazer uma avaliação final da qualidade dos casos de teste e definir pelo encerramento ou não de uma atividade de teste.

### **Técnicas de Teste**

Para testar um módulo, é necessário definir um caso de teste, executar o módulo com os dados de entrada definidos por esse caso de teste e analisar a saída. Um teste é um conjunto limitado de casos de teste, definido a partir do objetivo do teste.

Diversas técnicas de teste têm sido propostas visando apoiar o projeto de casos de teste. Essas técnicas podem ser classificadas, segundo a origem das informações utilizadas para estabelecer os objetivos de teste, em, dentre outras categorias, técnicas funcional, estrutural ou baseadas em máquinas de estado.

Os testes funcionais ou caixa-preta utilizam as especificações (de requisitos, análise e projeto) para definir os objetivos do teste e, portanto, para guiar o projeto de casos de teste. O conhecimento sobre uma determinada implementação não é usado. Assim, os testes são conduzidos na interface do software. Os testes caixa-preta são empregados para demonstrar que as funções do software estão operacionais, que a entrada é adequadamente aceita e a saída é corretamente produzida e que a integridade da informação externa (uma base de dados, por exemplo) é mantida.

Os testes estruturais ou caixa-branca estabelecem os objetivos do teste com base em uma determinada implementação, verificando detalhes do código. Caminhos lógicos internos são testados, definindo casos de testes que exercitem conjuntos específicos de condições ou laço.

Os testes baseados em máquinas de estado são projetados utilizando o conhecimento subjacente à estrutura de uma máquina de estados para determinar os objetivos do teste.

É importante ressaltar que técnicas de teste devem ser utilizadas de forma complementar, já que elas têm propósitos distintos e detectam categorias de erros distintas. À primeira vista, pode parecer que realizando testes caixa branca rigorosos poderíamos chegar a programas corretos. Contudo, conforme anteriormente mencionado, isso não é prático, uma vez que todas as combinações possíveis de caminhos e valores de variáveis teriam de ser exercitadas, o que é impossível. Isso não quer dizer, entretanto, que os testes caixa-branca não são úteis. Testes caixa-branca podem ser usados, por exemplo, para garantir que todos os caminhos independentes<sup>1</sup> de um módulo tenham sido exercitados pelo menos uma vez.

Há diversas técnicas de testes caixa-branca, cada uma delas procurando apoiar o projeto de casos de teste focando em algum ou vários aspectos da implementação. Dentre elas, podem ser citadas:

**Testes de estrutura de controle:** como o próprio nome diz, enfocam as estruturas de controle de um módulo, tais como comandos, condições e laços. Teste de condição é um tipo de teste de estrutura de controle que exercita as condições lógicas contidas em um módulo. Um teste de fluxo de dados, por sua vez, seleciona caminhos de teste tomando por base a localização das definições e dos usos das variáveis nos módulos. Testes de ciclo ou laço focalizam exclusivamente os laços (loops).

Teste de caminho básico: define uma medida de complexidade lógica de um módulo e usa essa medida como guia para definir um conjunto básico de caminhos de execução.

Assim como há diversas técnicas de teste caixa-branca, o mesmo acontece em relação ao teste caixa-preta. Dentre as diversas técnicas de teste caixa-preta, podem ser citadas:

Particionamento de equivalência: divide o domínio de entrada de um módulo em classes de equivalência, a partir das quais casos de teste são derivados. A meta é minimizar o número de casos de teste, ficando apenas com um caso de teste para cada classe, uma vez que, a princípio, todos os elementos de uma mesma classe devem se comportar de maneira equivalente.

Análise de valor limite: a prática mostra que um grande número de erros tende a ocorrer nas fronteiras do domínio de entrada de um módulo. Tendo isso em mente, a análise de valor limite leva à seleção de casos de teste que exercitem os valores limítrofes.

### **Estratégias de Teste**

projeto efetivo de casos de teste é importante, mas não suficiente para o sucesso da atividade de testes. A estratégia, isto é, a série planejada de realização dos testes, é também crucial. Basicamente, há três grandes fases de teste:

Teste de Unidade: tem por objetivo testar a menor unidade do projeto (um componente de software que não pode ser subdividido), procurando identificar erros de lógica e de implementação em cada módulo separadamente. No paradigma estruturado, a menor unidade refere-se a um procedimento ou função.

Teste de Integração: visa a descobrir erros associados às interfaces entre os módulos quando esses são integrados para formar estrutura do produto de software.

Teste de Sistema: tem por objetivo identificar erros de funções (requisitos funcionais) e características de desempenho (requisito não funcional) que não estejam de acordo com as especificações.

Tomando por base essas fases, a atividade de teste pode ser estruturada de modo que, em cada fase, diferentes tipos de erros e aspectos do software sejam considerados. Tipicamente, os primeiros testes focalizam componentes individuais e aplicam testes caixa-branca e caixa-preta para descobrir erros. Após os componentes individuais terem sido testados, eles precisam ser integrados, até se obter o sistema por inteiro. Na integração, o foco é o projeto e a arquitetura do sistema. Finalmente, uma série de testes de alto nível é executada quando o sistema estiver operacional, visando a descobrir erros nos requisitos.

No teste de unidade, faz-se necessário construir pequenos componentes para permitir testar os módulos individualmente, os ditos drivers e stubs. Um driver é um programa responsável pela ativação e coordenação do teste de uma unidade. Ele é responsável por receber os dados de teste fornecidos pelo testador, passar esses dados para a unidade sendo testada, obter os resultados produzidos por essa unidade e apresentá-los ao testador. Um stub é uma unidade que substitui, na hora do teste, uma outra unidade chamada pela unidade que está sendo testada. Em geral, um stub simula o comportamento da unidade chamada com o mínimo de computação ou manipulação de dados.

A abordagem de integração de módulos pode ter impacto na quantidade de drivers e stubs a ser construída. Sejam as seguintes abordagens:

Integração ascendente ou bottom-up: Nessa abordagem, primeiramente, cada módulo no nível inferior da hierarquia do sistema é testado individualmente. A seguir, são testados os módulos que chamam esses módulos previamente testados. Esse procedimento é repetido até que todos os módulos tenham sido testados [3]. Neste caso, apenas drivers são necessários. Seja o exemplo da figura 7.1. Usando a abordagem de integração ascendente, os módulos seriam testados da seguinte forma. Inicialmente, seriam testados os módulos do nível inferior (E, F e G). Para cada um desses testes, um driver teria de ser construído. Concluídos esses testes, passaríamos ao nível imediatamente acima, testando seus módulos (B, C e D) combinados com os módulos por eles chamados. Neste caso, testamos juntos B, E e F bem como C e G. Novamente, três drivers seriam necessários. Por fim, testaríamos todos os módulos juntos.

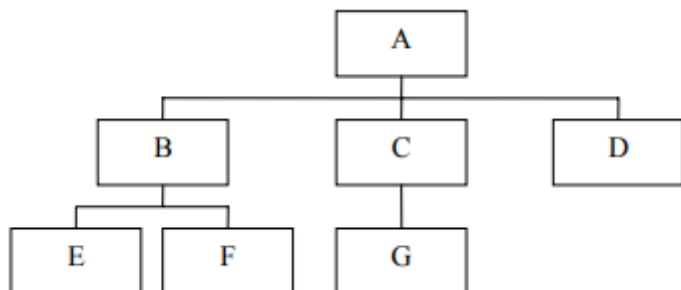


Figura 7.1 – Exemplo de uma hierarquia de módulos.

**Integração descendente ou top-down:** A abordagem, neste caso, é precisamente o contrário da anterior. Inicialmente, o nível superior (geralmente um módulo de controle) é testado sozinho. Em seguida, todos os módulos chamados por pelo módulo testado são combinados e testados como uma grande unidade. Essa abordagem é repetida até que todos os módulos tenham sido incorporados.

Neste caso, apenas stubs são necessários. Tomando o exemplo da figura 7.1, o teste iniciaria pelo módulo A e três stubs (para B, C e D) seriam necessários. Na sequência seriam testados juntos A, B, C e D, sendo necessários stubs para E, F e G. Por fim, o sistema inteiro seria testado.

Muitas outras abordagens, algumas usando as apresentadas anteriormente, podem ser adotadas, tal como a integração sanduíche, que considera uma camada alvo no meio da hierarquia e utiliza as abordagens ascendente e descendente, respectivamente para as camadas localizadas abaixo e acima da camada alvo. Outra possibilidade é testar individualmente cada módulo e só depois de testados individualmente integrá-los (teste big-band). Neste caso, tanto drivers quanto stubs têm de ser construídos para cada módulo, o que leva a muito mais codificação e problemas em potencial.

Uma vez integrados todos os módulos do sistema, parte-se para os testes de sistema, quando se busca observar se o software funciona conforme esperado pelo cliente. Por isso mesmo, muitas vezes, os testes de sistema são chamados de testes de validação. Os testes de sistema incluem diversos tipos de teste, realizados na seguinte ordem:

Teste funcional: verifica se o sistema integrado realiza as funções especificadas nos requisitos;

Teste de desempenho: verifica se o sistema integrado atende os requisitos não funcionais do sistema (eficiência, segurança, confiabilidade etc);

Teste de aceitação: os testes funcional e de desempenho são ainda realizados por desenvolvedores, entretanto é necessário que o sistema seja testado pelos clientes. No teste de aceitação, os clientes testam o sistema a fim de garantir que o mesmo satisfaz suas necessidades. Vale destacar que o que foi especificado pelos desenvolvedores pode ser diferente do que queria o cliente. Assim, o teste de aceitação assegura que o sistema solicitado é o que foi construído.

Teste de instalação: algumas vezes o teste de aceitação é feito no ambiente real de funcionamento, outras não. Quando o teste de aceitação for feito em um ambiente de teste diferente do local em que será instalado, é necessário realizar testes de instalação.

### Entrega E Manutenção

Concluídos os testes, sistema aceito e instalado, estamos chegando ao fim do processo de desenvolvimento de software. A entrega é a última etapa desse processo. Uma vez entregue, o sistema passa a estar em operação e eventuais mudanças, sejam de caráter corretivo, sejam de caráter de evolução, caracterizam-se como uma manutenção.

### Entrega

A entrega não é meramente uma formalidade. No momento em que o sistema é instalado no local de operação e devidamente aceito, é necessário, ainda, ajudar os usuários a entenderem e a se sentirem mais familiarizados com o sistema. Neste momento, duas questões são cruciais para uma transferência bem-sucedida: treinamento e documentação.

A operação do sistema é extremamente dependente de pessoal com conhecimento e qualificação. Portanto, é essencial que o treinamento de pessoal seja realizado para que os usuários e operadores possam operar o sistema adequadamente.

A documentação que acompanha o sistema também tem papel crucial na entrega, afinal ela será utilizada como material de referência para a solução de problemas ou como informações adicionais. Essa documentação inclui, dentre outros, manuais do usuário e do operador, guia geral do sistema, tutoriais, ajuda (help), preferencialmente on-line e guias de referência rápida.

### **Manutenção**

Desenvolvimento de um sistema termina quando o produto é entregue para o cliente e entra em operação. A partir daí, deve-se garantir que o sistema continuará a ser útil e atendendo às necessidades do usuário, o que pode demandar alterações no mesmo. Começa, então, a fase de manutenção.

Há muitas causas para a manutenção, dentre elas: falhas no processamento devido a erros no software, falhas de desempenho, alterações no ambiente de dados, alterações no ambiente de processamento, necessidade de modificações em funções existentes e necessidade de inclusão de novas capacidades.

Ao contrário do que podemos pensar, a manutenção não é uma atividade trivial nem de pouca relevância. Ela é uma atividade importantíssima e de intensa necessidade de conhecimento. O mantenedor precisa conhecer o sistema, o domínio de aplicação, os requisitos do sistema, a organização que utiliza o mesmo, práticas de engenharia de software passadas e atuais, a arquitetura do sistema, algoritmos usados etc.

Processo de manutenção é semelhante, mas não igual, ao processo de desenvolvimento e pode envolver atividades de levantamento de requisitos, análise, projeto, implementação e testes, agora no contexto de um software existente. Essa semelhança pode ser maior ou menor, dependendo do tipo de manutenção a ser realizada.

Pfleeger aponta os seguintes tipos de manutenção:

**Manutenção corretiva:** trata de problemas decorrentes de defeitos. À medida que falhas ocorrem, elas são relatadas à equipe de manutenção, que se encarrega de encontrar o defeito que causou a falha e faz as correções (nos requisitos, análise, projeto ou implementação), conforme o necessário. Esse reparo inicial pode ser temporário, visando manter o sistema funcionando. Quando esse for o caso, mudanças mais complexas podem ser implementadas posteriormente.

**Manutenção adaptativa:** às vezes, uma mudança no ambiente do sistema, incluindo hardware e software de apoio, pode implicar em uma necessidade de adaptação.

**Manutenção perfectiva:** consiste em realizar mudanças para melhorar algum aspecto do sistema, mesmo quando nenhuma das mudanças for consequência de defeitos. Isso inclui a adição de novas capacidades bem como ampliações gerais.

**Manutenção preventiva:** consiste em realizar mudanças a fim de prevenir falhas. Geralmente ocorre quando um mantenedor descobre um defeito que ainda não causou falha e decide corrigi-lo antes que ele gere uma falha

---

---

---

---

---

---

---

---

---

---

## Processo Unificado

A **UML** (Unified Modeling Language), que significa Linguagem Unificada de Modelagem é uma linguagem padrão para modelagem orientada a objetos. Ela surgiu da fusão de três grandes métodos, do BOOCH, OMT (Rumbaugh) e OOSE (Jacobson). Esta linguagem de modelagem não proprietária de terceira geração, não é um método de desenvolvimento. Têm como papel auxiliar a visualizar o desenho e a comunicação entre objetos. Ela permite que desenvolvedores visualizem os produtos de seu trabalho em **diagramas** padronizados, e é muito usada para criar modelos de sistemas de software.

Além de fornecer a tecnologia necessária para apoiar a prática de **engenharia de software** orientada a objetos, a UML poderá ser a linguagem de modelagem padrão para modelar sistemas concorrentes e distribuídos. Utiliza-se de um conjunto de técnicas de notação gráfica para criar modelos visuais de software de sistemas intensivos, combinando as melhores técnicas de modelagem de dados, negócios, objetos e componentes. É uma linguagem de modelagem única, comum e amplamente utilizável.

Embora com a UML seja possível representar o software através de modelos orientados a objetos, ela não demonstra que tipo de trabalho deve ser feito, ou seja, não possui um processo que define como o trabalho tem que ser desenvolvido. O objetivo então é descrever "o que fazer", "como fazer", "quando fazer" e "porque deve ser feito". É necessária a elaboração completa de um dicionário de dados, para descrever todas as entidades envolvidas, refinando, com isso, os requisitos funcionais do software.

A Linguagem Unificada de Modelagem possui diagramas (representações gráficas do modelo parcial de um sistema) que são usados em combinação, com a finalidade de obter todas as visões e aspectos do sistema.

Os Diagramas da UML estão divididos em Estruturais e Comportamentais.

### Diagramas Estruturais

**De Classe:** Este diagrama é fundamental e o mais utilizado na UML e serve de apoio aos outros diagramas. O Diagrama de Classe mostra o conjunto de classes com seus atributos e métodos e os relacionamentos entre classes.

**De Objeto:** O diagrama de objeto esta relacionado com o diagrama de classes e, é praticamente um complemento dele. Fornece uma visão dos valores armazenados pelos objetos de um Diagrama de Classe em um determinado momento da execução do processo do software.

**De Componentes:** Está associado à linguagem de programação e tem por finalidade indicar os componentes do software e seus relacionamentos.

**De implantação:** Determina as necessidades de hardware e características físicas do Sistema.

**De Pacotes:** Representa os subsistemas englobados de forma a determinar partes que o compõem.

**De Estrutura:** Descreve a estrutura interna de um classificador.

### Diagramas Comportamentais

**De Caso de Uso (Use Case):** Geral e informal para fases de levantamento e análise de Requisitos do Sistema.

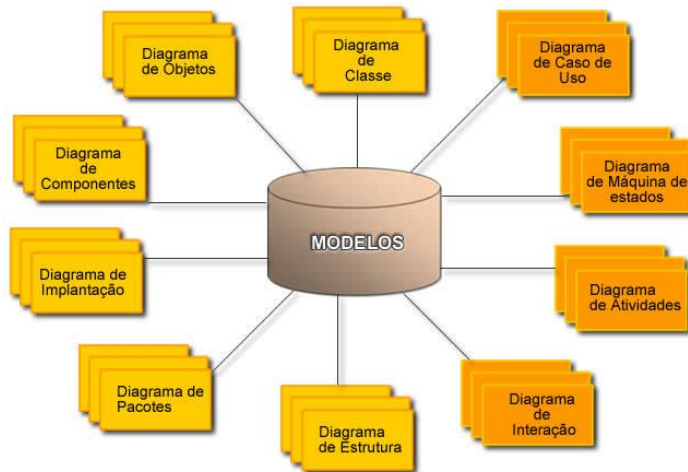
**De Máquina de Estados:** Procura acompanhar as mudanças sofridas por um objeto dentro de um processo.

**De Atividades:** Descreve os passos a serem percorridos para a conclusão de uma atividade.

**De Interação:** Dividem-se em:

1. De Sequência: Descreve a ordem temporal em que as mensagens são trocadas entre os objetos.

2. Geral interação: Variação dos diagramas de atividades que fornece visão geral dentro do sistema ou processo do negócio.
3. De comunicação: Associado ao diagrama de Seqüência, complementando-o e concentrando-se em como os objetos estão vinculados.
4. De tempo: Descreve a mudança de estado ou condição de uma instância de uma classe ou seu papel durante o tempo.



### Introdução a UML e seus Diagramas

A Unified Modelling Language (UML) é uma linguagem ou notação de diagramas para especificar, visualizar e documentar modelos de 'software' orientados por objetos. O UML não é um método de desenvolvimento, o que significa que não lhe diz o que fazer primeiro ou o que fazer depois ou como desenhar o seu sistema, mas ajuda-o a visualizar o seu desenho e a comunicar com os outros. O UML é controlado pelo Object Management Group (OMG) e é a norma da indústria para descrever graficamente o 'software'.

O UML está desenhado para o desenho de 'software' orientado por objetos e tem uma utilização limitada para outros paradigmas de programação.

A UML é composta por muitos elementos de modelo que representam as diferentes partes de um sistema de software. Os elementos UML são usados para criar diagramas, que representam uma determinada parte, ou um ponto de vista do sistema.

#### Objetivos

Os objetivos da UML são: especificação, documentação, estruturação para sub-visualização e maior visualização lógica do desenvolvimento completo de um sistema de informação. A UML é um modo de padronizar as formas de modelagem.

#### Para que usar UML?

Ajudar a conceber nossas idéias, em relação ao sistema que estivermos projetando – Pensar antes de codificar;

Apresentar nossas idéias ao grupo de forma que todos possam interagir e discutir um determinado ponto – Aumentar a participação e envolvimento do time;

Documentar nossas idéias quando elas já estiverem bem consolidadas para que novos integrantes e novos colaboradores possam acelerar sua compreensão dos sistemas desenvolvidos pelo grupo;

#### Tipos de Diagramas

Diagramas servem para capturar diferentes visões do sistema:

**Estrutural: estática**

Diagrama de Classes

Diagrama de Objetos

Diagrama de Componentes

Diagrama de Implantação

**Comportamental: dinâmica**

Diagrama de Casos de Uso

Diagrama de Seqüência

Diagrama de Atividades

Diagrama de Estados

Diagrama de Colaboração

Diagramas Estáticos

Diagrama de Classes

Diagramas de classe mostram as diferentes classes que fazem um sistema e como elas se relacionam. Os diagramas de classe são chamados diagramas estáticos porque mostram as classes, com seus métodos e atributos bem como os relacionamentos estáticos entre elas: quais classes conhecem quais classes ou quais classes são parte de outras classes, mas não mostram a troca de mensagens entre elas.

Na UML, atributos são mostrados com pelo menos seu nome, e podem também mostrar seu tipo, valor inicial e outras propriedades. Atributos podem também ser exibidos com sua visibilidade:

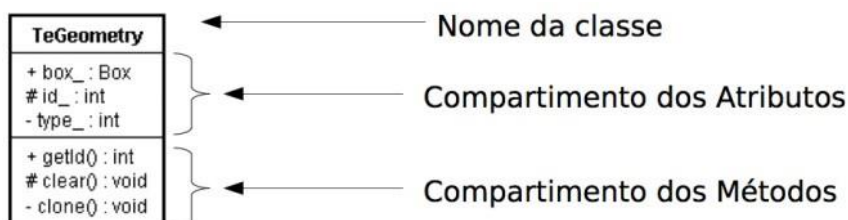
+ indica atributos ou métodos públicos (significa que ele pode ser acessado de qualquer lugar da classe ou sub-classe);

# indica atributos ou métodos protegidos (significa que ele só pode ser acessado de dentro da própria classe ou em suas classes-filha);

- indica atributos ou métodos privados (significa que ele só pode ser acessado de dentro da própria classe.);

~ pacote.

As operações também são exibidas com pelo menos seu nome, e podem também mostrar seus parâmetros e valores de retorno.



Uma observação a ser feita é que quando se tem um atributo ou método estáticos, eles irão aparecer sublinhados no diagrama. Já os métodos abstratos irão aparecer em itálico.

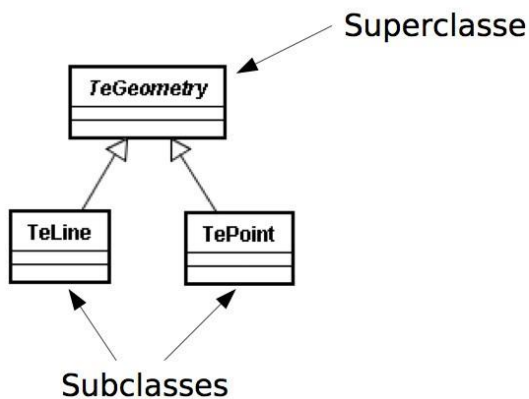
Classes podem ter modelos, um valor que é usado para uma classe ou tipo não especificado. O tipo de modelo é especificado quando uma classe é iniciada (isto é um objeto é criado). Modelos existem no C ++ moderno e foram introduzidos no Java 1.5 onde eles são chamados de genéricos.

Existem alguns tipos de relacionamentos no diagrama de classes que veremos abaixo:

**Generalização**

A herança é um dos conceitos fundamentais da programação orientada por objetos, nos quais uma classe “ganha” todos os atributos e operações da classe que herda, podendo sobrepor ou modificar algumas delas, assim como adicionar mais atributos ou operações próprias.

EM UML, uma associação Generalização entre duas classes coloca-as numa hierarquia representando o conceito de herança de uma classe derivada de uma classe base. Em UML, Generalizações são representadas por uma linha conectando duas classes, com uma seta no lado da classe base.



Generalização

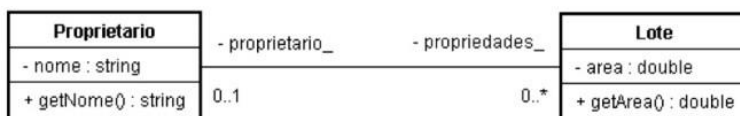
**Associação**

Um associação representa um relacionamento entre classes, e fornece a semântica comum e a estrutura para muitos tipos de “conexões” entre objetos.

Associações são o mecanismo que permite objetos comunicarem-se entre si. Elas descrevem a conexão entre diferentes classes (a conexão entre os objetos atuais é chamada conexão do objeto, ou link).

Associações podem ter um regra que especifica o propósito da associação e pode ser uni ou bidirecional (indicando se os dois objetos participantes do relacionamento podem mandar mensagens para o outro, ou se apenas um deles sabe sobre o outro). Cada ponta da associação também possui um valor de multiplicidade, que dita como muitos objetos neste lado da associação pode relacionar-se com o outro lado.

Em UML, associações são representadas como linhas conectando as classes participantes do relacionamento, e podem também mostrar a regra e a multiplicidade de cada um dos participantes. A multiplicidade é exibida como um intervalo [min...máx] de valores não negativos, com uma estrela (\*) no lado máximo representando infinito.



Associação

**Agregação**

Agregações são um tipo especial de associação no qual as duas classes participantes não possuem em nível igual, mas fazem um relacionamento “todo-parte”. Uma Agregação descreve como a classe



que possui a regra do todo, é composta (tem) de outras classes, que possuem a regra das partes. Para Agregações, a classe que age como o todo sempre tem uma multiplicidade de um.

Em UML, Agregações são representadas por uma associação que mostra um romboide no lado do todo.



Agregação

### Composição

Composições são associações que representam agregações muito fortes. Isto significa que Composições formam relacionamentos todo-parte também, mas o relacionamento é tão forte que as partes não pode existir independentes. Elas existem somente dentro do todo, e se o todo é destruído as partes morrem também.

Em UML, Composições são representadas por um romboide sólido no lado do todo.



Composição

### Interfaces

Interfaces são classes abstratas que significam instâncias que não podem ser diretamente criadas delas. Elas podem conter operações mas não podem conter atributos. Classes podem derivar de interfaces (através da realização de uma associação) e instâncias podem então ser feitas destes diagramas.

### Tipos de Dados

Tipos de dados são primitivos uma vez que são tipicamente construídos numa linguagem de programação. Exemplos comuns são inteiros e lógicos. Eles não podem ser relacionados à classes mas classes pode se relacionar com eles.

### Enumerações

Enumerações são uma lista simples de valores. Um exemplo típico é uma enumeração para dias da semana. As opções de uma enumeração são chamadas Literais de Enumeração. Como tipos de dados, elas não podem ter relacionamentos para classes mas classes podem relacionar-se com elas.

### Pacotes

Pacotes representam um espaço de nomes numa linguagem de programação. Num diagrama eles são usados para representar partes de um sistema que contém mais de uma classe, talvez centenas de classes.

### Diagrama de Objetos

O diagrama de objetos é uma variação do diagrama de classes e utiliza quase a mesma notação. A diferença é que o diagrama de objetos mostra os objetos que foram instanciados das classes. O diagrama de objetos é como se fosse o perfil do sistema em um certo momento de sua execução.

A mesma notação do diagrama de classes é utilizada com duas exceções: os objetos são escritos com seus nomes sublinhados e todas as instâncias num relacionamento são mostradas. Os diagramas de objetos não são tão importantes como os diagramas de classes, mas eles são muito úteis para exemplificar diagramas complexos de classes ajudando muito em sua compreensão. Diagramas de objetos também são usados como parte dos diagramas de colaboração (passou a se

chamar comunicação na uml 2.0), onde a colaboração dinâmica entre os objetos do sistema são mostrados.

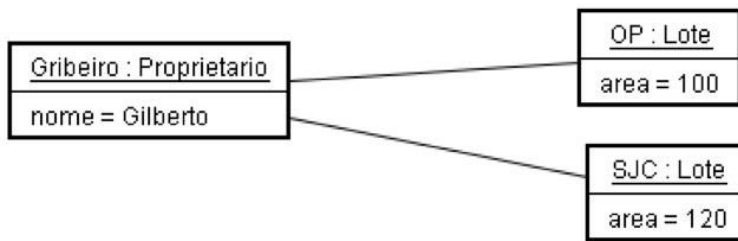


Diagrama de Objetos

Diagrama de Componentes

Diagramas de componente mostram os componentes do software (sejam componentes de tecnologias como KParts, componentes CORBA ou Java Beans ou apenas seções do sistema que são claramente distintas) e os artefatos de que eles são feitos como arquivos de código-fonte, bibliotecas de programação ou tabelas de bancos de dados relacionais.

Componentes podem possuir interfaces (isto é classes abstratas com operações) que permitem associações entre componentes.

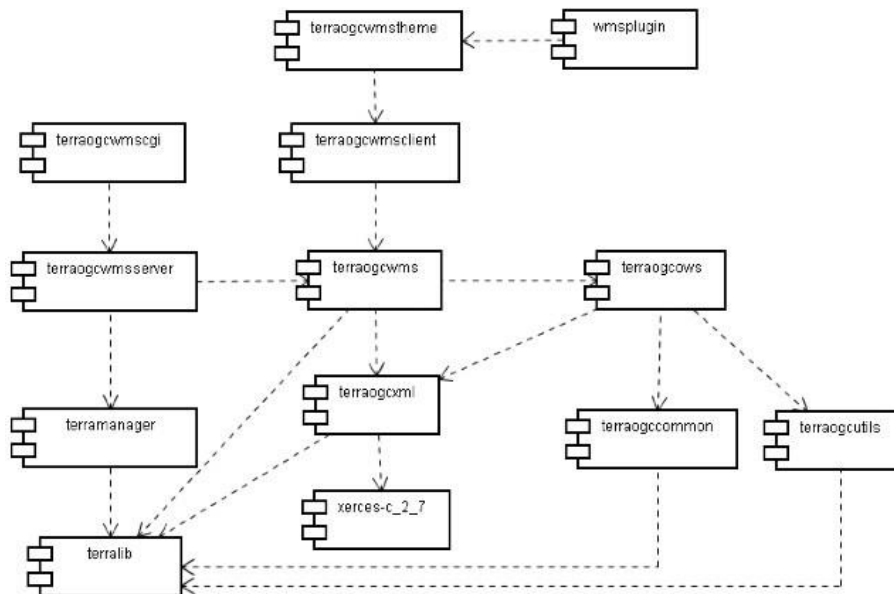


Diagrama de Componentes

Diagrama de Implantação

O Diagrama de instalação/implantação é definido pela Linguagem de Modelagem Unificada (Unified Modeling Language – UML), descreve os componentes de hardware e software e sua interação com outros elementos de suporte ao processamento. Representa a configuração e a arquitetura de um sistema em que estarão ligados seus respectivos componentes, sendo representado pela arquitetura física de hardware, processadores etc.

**Nó:** Representa uma peça física de equipamento na qual o sistema será implantado.

**Artefatos:** Qualquer pedaço físico de informação usada ou produzida por um sistema.

**Especificação de implantação:** Especifica um conjunto de propriedades que determina os parâmetros de execução de um artefato que está instalado em um nó.

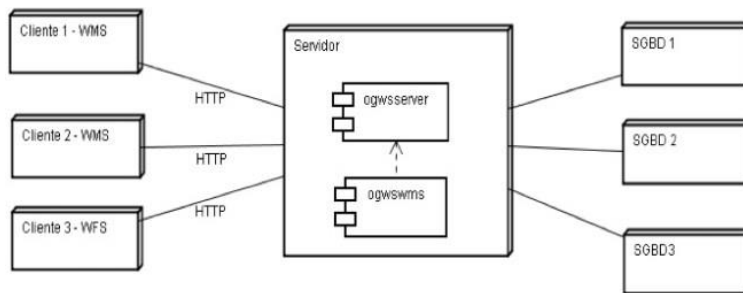


Diagrama de Implantação

Diagramas Dinâmicos

Diagrama de Casos de Uso

Diagramas de Caso de Uso descrevem relacionamentos e dependências entre um grupo de Caso de Uso e os Atores participantes no processo.

É importante observar que Diagramas de Caso de Uso não são adequados para representar o desenho, e não podem descrever os mecanismos internos de um sistema. Diagramas de Caso de Uso são feitos para facilitar a comunicação com os futuros usuários do sistema, e com o cliente, e são especialmente úteis para determinar os recursos necessários que o sistema deve ter. Diagramas de Caso de Uso dizem o quê o sistema deve fazer, mas não fazem — e não podem — especificar como isto será conseguido.

Um Caso de Uso descreve — do ponto de vista dos atores — um grupo de atividades num sistema que produz um resultado concreto e tangível.

Casos de Uso são descrições de interações típicas entre os usuários de um sistema e o sistema propriamente dito. Eles representam a interface externa do sistema e especificam um conjunto de exigências do que o sistema deve fazer (lembre-se: somente o quê, não como).

Quando trabalhar com Casos de Uso, é importante lembrar-se de algumas regras simples

Cada Caso de Uso está relacionado com no mínimo um ator;

Cada Caso de Uso possui um iniciador (isto é um ator);

Cada Caso de Uso liga-se a um resultado relevante (um resultado com “valor de negócio”).

Casos de Uso também podem ter relacionamentos com outros Casos de Uso. Os três tipos mais comuns de relacionamento entre Casos de Uso são:

**<<inclui-se>>** que especifica que um Caso de Uso toma lugar dentro de outro Caso de Uso

**<<estende>>** que especifica que em determinadas situações, ou em algum ponto (chamado um ponto de extensão) um Caso de Uso será estendido por outro.

**Generalização** especifica que um Caso de Uso herda as características do “Super” Caso de Uso, e pode sobrepor algumas delas ou adicionar novas de maneira semelhante a herança entre classes.

### Ator

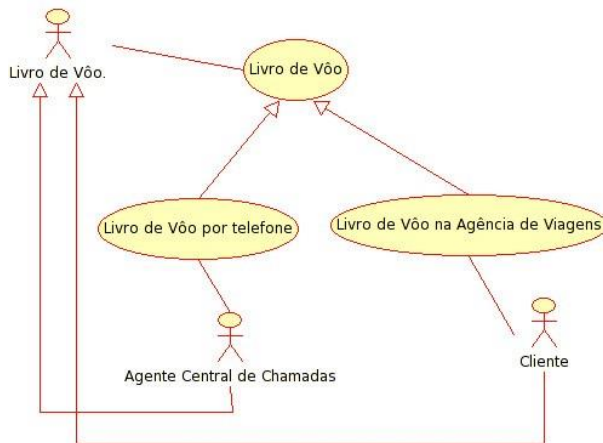
Um ator é uma entidade externa (fora do sistema) que interage com o sistema participando (e frequentemente iniciando) um Caso de Uso. Atores podem ser pessoas reais (por exemplo usuários do sistema), outro sistema de computador ou eventos externos.

Atores não representam as pessoa física ou sistemas, mas sua regra. Isto significa que quando uma pessoa interage com o sistema de diferentes maneiras (assumindo diferentes regras) ela será representada por diversos atores. Por exemplo um pessoa que fornece suporte ao cliente por telefone

e recebe ordens do cliente para o sistema pode ser representado por um ator da “Equipe de Suporte” e um ator “Representante de Vendas”

**Descrição do Caso de Uso**

Descrição do Caso de Uso são narrativas de texto do Caso de Uso. Elas usualmente tomam a forma de uma nota ou um documento que é de alguma maneira ligado ao Caso de Uso, e explana o processo ou atividades que tomarão lugar no Caso de Uso.



**Caso de Uso**

**Diagrama de Sequência**

É usado para mostrar uma seqüência de atividades. Mostra o fluxo de trabalho (workflow) a partir de um ponto inicial até um ponto final, detalhando as decisões do caminho tomado durante a execução das tarefas. Este diagrama possui várias aplicações, desde a definição do fluxo básico de um programa até a definição de um processo com as suas tomadas de decisões e ações.

Diagramas de Sequência mostram a troca de mensagens (isto é chamada de método) entre diversos Objetos, numa situação específica e delimitada no tempo. Objetos são instâncias de classes. Diagramas de Sequência colocam ênfase especial na ordem e nos momentos nos quais mensagens para os objetos são enviadas.

Em Diagramas de Sequência objetos são representados através de linhas verticais tracejadas, com o nome do Objeto no topo. O eixo do tempo é também vertical, aumentando para baixo, de modo que as mensagens são enviadas de um Objeto para outro na forma de setas com a operação e os nomes dos parâmetros.



**Diagrama de Sequência**

**Diagramas de Atividade**

O Diagrama de Atividade descreve a sequência de atividades num sistema com a ajuda as Atividades. Diagramas de Atividade são uma forma especial de Diagramas de Estado, que somente (ou principalmente) contém Atividades.

Os diagramas de atividade não são importantes somente para a modelagem de aspectos dinâmicos de um sistema ou um fluxograma, mas também para a construção de sistemas executáveis por meio de engenharia de produção reversa.

Alguns conceitos:

**Atividades:** Comportamento a ser realizado.

**Sub-atividade:** Execução de uma sequência não atômica de atividades.

**Transição:** Fluxo de uma atividade para outra.

**Ação:** Transformação.

**Decisão:** Dependendo de uma condição, mostra as diferentes transições.

**Raia:** Diferenciação de unidades organizacionais.

**Bifurcação (Fork):** Separa uma transição em várias transições executadas ao mesmo tempo.

**Sincronização (Join):** Concatenação de transições vindas do Fork.

**Objecto:** O objecto da atividade.

**Envio de sinal:** Transição pra um meio externo, por exemplo, um hardware.

**Recepção de sinal:** Recepção do envio.

**Região:** Agrupamento de uma ou mais atividades.

**Exceção:** Atividades que ocorrerem em decorrência de uma exceção.

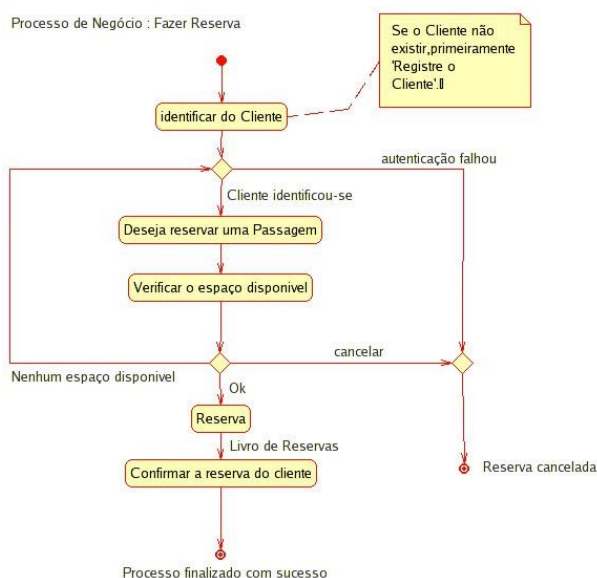


Diagrama de Atividades

Diagramas de Estado

Modela o comportamento de um objeto individual. Especifica as seqüências de estados pelos quais um objeto passa durante seu tempo de vida em resposta a eventos.

Diagramas de Estado mostram os diferentes estados de um Objeto durante sua vida, e o estímulo que faz com que o Objeto mude seu estado.

Diagramas de Estado veem Objetos como máquinas de estado ou automatismos finitos que podem ser um de um conjunto de estados finitos e que podem mudar seu estado através de um de um conjunto finito de estímulos. Por exemplo um tipo de Objeto ServidorRede pode estar em um dos seguintes estados durante sua vida:

Pronto

Ouvindo

Trabalhando

Parado

e os eventos que podem fazer com que o Objeto mude de estado são

Objeto é criado

Objeto recebe mensagem ouvir

Um Cliente solicita uma conexão através da rede

Um Cliente termina um pedido

O pedido é executado e terminado

Objeto recebe mensagem parar

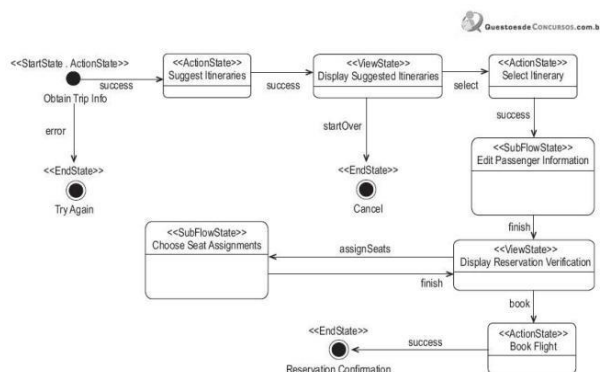
etc

### Estado

Estados são os blocos construídos dos Diagramas de Estado. Um Estado pertence a exatamente uma classe e representa um resumo dos valores dos atributos que uma classe pode tomar. Um Estado UML descreve o estado interno de um objeto para uma classe em particular

Observe que nem toda mudança em um dos atributos de um objeto pode ser representada por um Estado mas somente aquelas mudanças que podem afetar significativamente o trabalho do objeto

Existem dois tipos especiais de Estados: Inicial e Final. Eles são especiais porque nenhum evento pode fazer com que um Objeto retorne para seu estado Inicial, e da mesma maneira nenhum evento pode tirar um Objeto de seu estado Final uma vez que ele já o tenha alcançado.

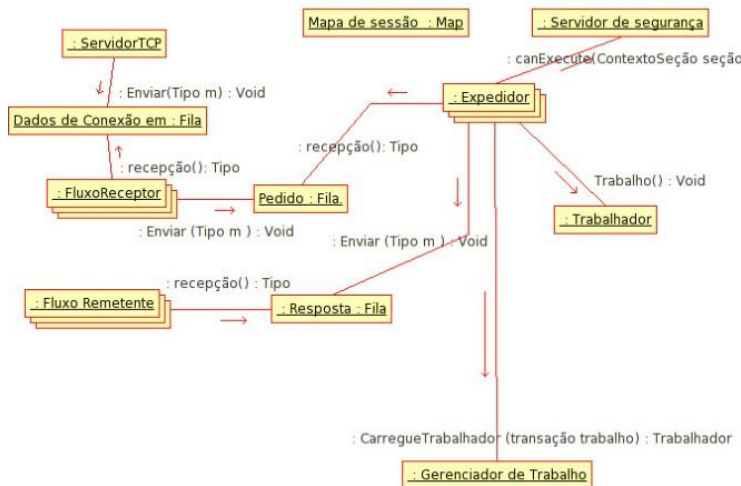


### Diagrama de Estados

### Diagramas de Colaboração

Diagramas de Colaboração mostram as interações que ocorrem entre os objetos participantes numa situação específica. Isto é mais ou menos a mesma informação mostrada pelos Diagramas de Sequência, mas neste a ênfase é colocada em como as interações ocorrem no tempo, enquanto os Diagramas de Colaboração colocam os relacionamentos entre os objetos e sua topologia em destaque.

Em Diagramas de Colaboração as mensagens enviadas de um objeto para outro são representadas por setas, mostrando o nome da mensagem, parâmetros, e a sequência da mensagem. Diagramas de Colaboração são especialmente indicados para mostrar um fluxo ou situação específica do programa e são um dos melhores tipos de diagrama para rapidamente demonstrar ou explicar um processo na lógica do programa.



### O que é UML?

A Linguagem de modelagem unificada (UML) foi criada para estabelecer uma linguagem de modelagem visual comum, semanticamente e sintaticamente rica, para arquitetura, design e implementação de sistemas de software complexos, tanto estruturalmente quanto para comportamentos. Além do desenvolvimento de software, a UML tem aplicações em fluxos do processo na fabricação.

É análoga aos modelos utilizados em outros campos, e é composta por diferentes tipos de diagramas. De modo geral, diagramas UML descrevem o limite, a estrutura e o comportamento do sistema e os objetos nele contidos.

A UML não é uma linguagem de programação, mas existem ferramentas que podem ser usadas para gerar código em várias linguagens por meio de diagramas UML. A UML tem uma relação direta com a análise e o design orientados a objetos.

### A UML e seu Papel na Modelagem e no Design Orientados a Objetos

Existem muitos paradigmas ou modelos de resolução de problemas na ciência da computação, que é o estudo de algoritmos e dados. Há quatro categorias modelo de resolução de problemas: linguagens imperativas, funcionais, declarativas e orientadas a objetos (POO). Em linguagens orientadas a objetos, os algoritmos são expressos através da definição de 'objetos', e por meio da interação dos objetos uns com os outros. Esses objetos são coisas a serem manipulados, e eles existem no mundo real. Eles podem ser edifícios, widgets em um desktop ou seres humanos.

Linguagens orientadas a objetos dominam o mundo da programação porque elas modelam objetos do mundo real. A UML é uma combinação de várias notações orientadas a objetos: design orientado a objetos, técnica de modelagem de objetos e engenharia de software orientada a objetos.

A UML usa os pontos fortes destas três abordagens para apresentar uma metodologia mais consistente e mais fácil de usar. A UML representa as melhores práticas para desenvolver e documentar aspectos diferentes da modelagem de software e sistemas de negócios.

## A História e as Origens da UML

'Os Três Amigos' da engenharia de software, como eram conhecidos, desenvolveram ainda outras metodologias. Uniram-se para proporcionar maior clareza aos programadores por meio da criação de novos padrões. A colaboração entre Grady, Booch e Rumbaugh fortaleceu os três métodos e melhorou o produto final.

Em 1996, os grandes esforços destes pensadores resultaram no lançamento dos documentos UML 0.9 e 0.91. Logo ficou claro que muitas organizações, incluindo Microsoft, Oracle e IBM, passaram a considerar a UML como algo crucial para seu próprio desenvolvimento de negócios. Juntamente com muitos outros indivíduos e empresas, as organizações estabeleceram recursos que poderiam desenvolver uma linguagem de modelagem plena. Em 1999, Os Três Amigos publicaram "UML: guia do usuário", e uma atualização que inclui informações sobre UML 2.0, em sua segunda edição, de 2005.

### OMG: tem outro significado

Segundo o site, The Object Management Group® (OMG®) (Grupo de gerenciamento de objetos, em tradução livre) é uma organização internacional voltada para aprovações de padrões tecnológicos, aberta a membros, sem fins lucrativos e fundada em 1989. Os padrões OMG são implementados por fornecedores, utilizadores finais, instituições acadêmicas e agências governamentais. Forças-tarefas OMG desenvolvem padrões de integração empresarial para muitas tecnologias e um grande número de indústrias. Os padrões de modelagem OMG, incluindo UML e Model Driven Architecture® (MDA®), possibilitam uma criação de design visualmente poderoso e a execução e manutenção de software, entre outros processos.

A OMG supervisiona a definição e manutenção de especificações UML. Esta supervisão permite aos engenheiros e programadores usarem uma única linguagem para diversas finalidades durante todas as fases do ciclo de vida do software e para todos os tamanhos de sistemas.

### O objetivo da UML de acordo com a OMG

O propósito da UML de acordo com a OMG:

Fornecer a arquitetos de sistemas, engenheiros de software e desenvolvedores de software ferramentas de análise, design e implementação para sistemas baseados em software, bem como para a modelagem de processos de negócios e similares.

Desenvolver as condições gerais da indústria ao permitir a interoperabilidade de ferramentas de modelagem visual de objetos. No entanto, para permitir a troca significativa de informações de modelos entre as ferramentas, é necessário um acordo sobre semântica e notação.

A UML atende aos seguintes requisitos:

Estabelecer uma definição formal de um metamodelo baseado em meta-objetos (MOF, em inglês) comuns que especifica a sintaxe abstrata da UML. A sintaxe abstrata define o conjunto de conceitos de modelagem UML, seus atributos e relacionamentos, bem como as regras para combinar estes conceitos para construir modelos UML parciais ou completos.

Fornecer uma explicação detalhada da semântica de cada conceito de modelagem UML. A semântica define, de forma independente da tecnologia, como os conceitos UML devem ser realizados por computadores.

Especificar os elementos de notação legíveis para humanos para representar os conceitos de modelagem UML individuais, bem como regras para combiná-los em uma variedade de tipos de diagramas correspondentes a diferentes aspectos dos sistemas modelados.

Definir maneiras pelas quais ferramentas UML podem entrar em conformidade com esta especificação. Isto é suportado (em uma especificação separada) por uma especificação baseada em XML de formatos de modelos de intercâmbio correspondentes (XMI) que devem ser realizados por ferramentas compatíveis.





## **Métricas e Estimativas de Software**

Imagine que você faça parte de uma equipe de Rally, e que você e sua equipe tenham que atravessar um deserto enorme e cheio de obstáculos e que 50% dos fatores críticos de sucesso dependam do seu navegador para estimar o tempo do percurso e a distância. Agora imagine-se diante de um projeto de Sistemas de Informação onde 50% dos fatores críticos de sucesso estão nos prazos e custos. Certamente você terá que fazer uma eficiente métrica e estimativa de software.

As métricas e estimativas de software vem se tornando um dos principais tópicos na Engenharia da Informação com a crescente exigência de seus consumidores pela qualidade, rapidez, comodidade e baixo custo de implantação e manutenção, é impossível não enxergar tais técnicas como alavanca para um produto de melhor qualidade, com custos adequados. Mas existem ainda muitas barreiras que impedem os profissionais da área de utilizarem tais técnicas ou de o fazerem de maneira errônea, embora a literatura disponível atualmente sobre engenharia da informação seja relativamente ampla e variada, o que nos leva a questionar: Por que as métricas e estimativas de software propostas para o desenvolvimento de sistemas não são fiéis à realidade e à dimensão do problema? Tais técnicas acompanharam a rápida evolução do setor?

Tais questões nos levam a crer que algumas métricas e estimativas de software ficaram obsoletas e outras ganharam vigor nas épocas atuais, quando a busca da qualidade de software vem crescendo com grande rapidez.

Acredito que o ato de medir e estimar é a parte mais importante de um projeto de sistema bem-sucedido e alguns fatos como: a falta de maturidade, o desinteresse das empresas de desenvolvimento de sistemas e a baixa popularidade deste assunto entre os profissionais da área de informática são algumas das principais causas para o insucesso e o alto custo dos sistemas de informação.

O termo métrica de software refere-se à mensuração dos indicadores quantitativos do tamanho e complexidade de um sistema. Estes indicadores são, por sua vez, utilizados para correlatar contra os desempenhos observados no passado afim de derivar previsões de desempenho futuro.

A métrica de software tem como princípios especificar as funções de coleta de dados de avaliação e desempenho, atribuir essas responsabilidades a toda a equipe envolvida no projeto, reunir dados de desempenho pertencentes à complementação do software, analisar os históricos dos projetos anteriores para determinar o efeito desses fatores e utilizar esses efeitos para pesar as previsões futuras. Estes princípios nos permitem prever o resto do processo, avaliar o progresso e reduzir a complexidade, como numa prova de rally, onde a cada corrida ficamos mais esclarecidos das condições e limites da equipe.

### **As Métricas Orientadas ao Tamanho**

Métricas de software orientadas ao tamanho são medidas diretas do software e do processo por meio do qual ele é desenvolvido. Se uma organização de software mantiver registros simples, uma tabela de dados orientada ao tamanho poderá ser criada. A tabela relaciona cada projeto de desenvolvimento de software que foi incluído no decorrer dos últimos anos aos correspondentes dados orientados ao tamanho deste projeto. A partir dos dados brutos contidos na tabela, um conjunto de métricas de qualidade e de produtividade orientadas ao tamanho pode ser desenvolvido para cada projeto. Médias podem ser computadas levando-se em consideração todos os projetos.

As métricas orientadas ao tamanho provocam controvérsias e não são universalmente aceitas como a melhor maneira de se medir o processo de desenvolvimento de software. A maior parte da controvérsia gira em torno do uso das linhas de código (LOC) como uma medida-chave.

Os proponentes da afeição de linhas de código afirmam que as mesmas são o "artefato" de todos os projetos de desenvolvimento de software que podem ser facilmente contados, que muitos modelos existentes usam LOC ou KLOC (milhares de linhas de código) como entrada-chave e que já existe um grande volume de literatura e de dados baseados nas linhas de código. Por outro lado, os opositores afirmam que as medidas LOC são dependentes da linguagem de programação utilizada na codificação do projeto, que elas penalizam programas bem projetados, porém mais curtos, que elas não podem acomodar facilmente linguagens não-procedurais e que seu uso em estimativas requer um nível de detalhes que pode ser difícil de conseguir (isto é, o planejador deve estimar as linhas de código a ser produzidas muito antes que a análise e o projeto tenham sido construídos).

Essa forma de medida foi uma herança do modelo de manufatura em que os CIO'S podiam determinar os recursos necessários para uma "corrida", contando o número de produtos manufaturados necessários. Essa métrica não leva em consideração o fato de que o desenvolvimento envolve um custo relativo ao ambiente ou linguagem de programação utilizada. Por exemplo, em orientação a objeto (OO), a flexibilidade da ferramenta no uso de mecanismos de herança dilui o resultado final da contagem de linhas.

A contagem de linhas de código pode ser uma medida do que foi feito, e não uma medida a ser utilizada para previsão.

### **Métricas Orientadas à Função**

Consiste em um método para medição de software do ponto de vista do usuário, que determina de forma consistente o tamanho e complexidade de um software, sob a perspectiva do usuário. Ele dimensiona um software, quantificando a funcionalidade proporcionada ao usuário a partir do seu desenho lógico. Ou seja, são medidas indiretas do software e do processo por meio do qual ele é desenvolvido. Em vez de contar linhas de código, a métrica orientada à função concentra-se na funcionalidade ou utilidade do programa. Uma abordagem foi sugerida baseada nesta proposta chamada de pontos-por-função (function point). Os pontos-por-função (FP's) são derivados usando-se uma relação empírica baseada em medidas de informações e complexidade do software.

Um dos princípios da análise de pontos-por-função focaliza-se na perspectiva de como os usuários "enxergam" os resultados que um sistema produz. A análise considera as várias formas com que os usuários interagem com o sistema, com os seguintes objetivos:

1. Fornecer medidas consistentes;
2. Medir funcionalidades que o usuário solicita ou recebe;
3. Independência da tecnologia;
4. Método simples.

As métricas orientadas à função apresentam vários benefícios, dentre eles podemos citar o seguintes:

1. Uma ferramenta para dimensionar aplicações;
2. Um veículo para quantificar custo, esforço e tempo;
3. Um veículo para calcular índices de produtividade e qualidade;
4. Um fator de normalização para comparar software.

Tal métrica parece ser útil e funcional para o desenvolvimento tradicional, mas apresenta algumas falhas com o modelo de desenvolvimento em orientação a objeto (OO), pois alguns atributos do design em OO invalidam o cálculo de alguns pontos-por-função. As características fundamentais de OO têm efeito de reduzir a validade da contagem de funções para a avaliação de esforço e recursos necessários para a execução de um projeto.

A métrica de pontos por função foi originalmente projetada para sistemas de informação comerciais. Para acomodar estas aplicações, a dimensão dos dados foi enfatizada para a exclusão de dimensões funcionais e de controle. Por esta razão, a medida de pontos por função era adequada para muitos sistemas de engenharia. Um número de extensões para a medida básica de pontos por função tem sido proposto para remediar esta situação.

Uma extensão de pontos por função chamada "feature points" (ou, pontos característicos) é uma evolução da medida de pontos por função que pode ser aplicada a sistemas e aplicações de engenharia de software. A medida "feature points" acomoda aplicações em que a complexidade algorítmica é alta. Sistemas real-time de controle de processos e outros apresentam alta complexidade algorítmica, e são receptivos a métrica de "feature points".

Para computar o "feature point", valores do domínio são contados e ponderados. A métrica "feature point" conta uma nova característica de software, os algoritmos. Um algoritmo é definido como "um problema computacional que é incluído com um programa de computador específico". Inverte uma matriz, decodificar um bit de string ou manusear uma interrupção são exemplos de algoritmos.

Outra extensão de pontos por função para sistemas real-time e produtos de engenharia tem sido desenvolvido por Boeing. A aproximação de Boeing integra a dimensão dos dados de software com dimensões funcionais e de controle para obter uma medida orientada à função, chamada pontos por função 3D, que é receptiva a aplicações que enfatizem capacidades de função e controle. Características de todas as três dimensões dão "contadas, quantificadas e transformadas" em uma medida que fornece uma indicação da funcionalidade fornecida pelo software.

Contagem de dados retidos (a estrutura de dados interna do programa, isto é, arquivos) e dados externos (entradas, saídas e referências externas) são usados com medidas de complexidade para derivar uma contagem da dimensão de dados.

A dimensão funcional é medida considerando "o número de informações internas requeridas para transformar entradas em dados de saída". Para os propósitos da computação de pontos por função 3D, uma transformação é vista como uma série de passos de processamento que são limitados por regras semânticas estabelecidas. Como uma regra geral, a transformação é concluída com um algoritmo que resulta em uma mudança fundamental para dados de entrada como são processados para se transformarem em dados de saída. Passos de processamento que adquirem dados de um arquivo e simplesmente os coloca na memória do programa não poderia ser considerado uma transformação. O dado não sofreu nenhuma mudança.

O nível de complexidade associado a cada transformação é uma função do número de passos de processamento e o número de regras semânticas é que controla os passos de processamento.

A dimensão de controle é medida pela contagem do número de transições entre os estados. Um estado representa algum modo internamente observável de comportamento e uma transição ocorre como resultado de algum evento que força o software ou sistema a mudar seu comportamento, isto é, mudar seu estado.

Quando pontos por função 3D são computados, transições não são associadas a valores de complexidade.

Nota-se que pontos por função, "feature points" e pontos por função 3D representam a mesma coisa – "funcionalidade" ou "utilidade" fornecida pelo software. De fato, cada uma destas medidas resulta no mesmo valor se somente a dimensão de dados de uma aplicação é considerada.

Para sistemas real-time mais complexos, a contagem "feature points" é de 20 a 35% mais alta que a contagem determinada usando somente pontos por função.

Pontos por função (e suas extensões), como a medida LOC, é controversa. Os proponentes acham que FP é independente da linguagem de programação, tornando-se ideal para aplicações usando linguagens convencionais e não procedurais, e que ela é baseada em dados que são conhecidos muito cedo na evolução do projeto, fazendo a FP mais atrativa como uma estimativa mais próxima.

Oponentes acham que o método requer alguma prestidigitação em que a computação é baseada em dados subjetivos, que a contagem das informações de domínio (e outras dimensões) podem ser difíceis de coletar após terminado o projeto, e que FP não tem significado físico direto. É só um número.

### **Métricas Voltadas para Orientação a Objeto**

Muitas métricas já foram desenvolvidas para gerações passadas de tecnologia e, em muitos casos, são usadas até para desenvolvimento OO, porém não são muito coerentes, pois a diferença entre sistemas tradicionais e sistemas OO são muito grandes.

Existem várias propostas para métricas OO que levam em consideração as características básicas e interações do sistema como: número de classes, número de cases, número de métodos, médias de métodos, médias de métodos por classe, linhas de código por método, profundidade máxima da hierarquia de classes, a relação existente entre métodos públicos e privados, entre outros.

Tais métricas baseiam-se na análise detalhada do design do sistema. Como na técnica de pontos-por-função, faz sentido adicionar um peso às métricas das classes para produzir uma medida de complexidade do sistema. A maioria das medidas examina atributos em termos dos conceitos de OO, como herança, polimorfismo e encapsulamento. Para tanto, seria necessário coletar um número significativo de contagens, ou seja, seria necessário tomar valores de vários projetos e dimensioná-los selecionando as classes, os métodos e os atributos desejáveis para medir o tamanho e a complexidade de um novo software, o que nos tomaria um longo tempo.

### **Estimativa de Tempo**

Após desenvolver uma estimativa do volume de trabalho a ser feito, você pode pensar que é fácil estimar a extensão do tempo que o projeto exigirá. Afinal, se você tem um projeto estimado em dez pessoas-mês e há cinco pessoas disponíveis ele deve levar dois meses para ser concluído. Mas, e se somente duas pessoas estiverem disponíveis? O projeto leva cinco meses para ficar pronto? De um modo geral, a nossa preocupação aqui é com a relação tempo/pessoal. Muitos anos de dolorosa experiência ensinaram-nos que tal relação não é simples.

Duplicar o número de pessoas em um projeto não reduz necessariamente a duração do projeto pela metade (muito pelo contrário, se colocarmos mais pessoas num projeto em andamento isso apenas retardará ainda mais o processo, uma vez que estas pessoas deverão receber treinamento adequado e "aprender" todo o projeto desde seu início até a fase atual, e isso consome muito tempo).

A estimativa do esforço é a técnica mais comum para se levantar os custos de qualquer projeto de desenvolvimento de engenharia. Um número de pessoas-dia, pessoas-mês ou pessoas-ano é aplicado à solução de cada tarefa do projeto. Um custo em dólares é associado a cada unidade de esforço e um custo estimado será derivado. Como a técnica LOC (linhas de código) ou FP (pontos-por-função), a estimativa de esforço inicia-se com um delineamento das funções do software obtidas a partir do escopo do projeto. Uma série de tarefas de engenharia de software - análise de requisitos, projeto, codificação e teste - deve ser executada para cada função.

O planejador estima o esforço (por exemplo, pessoas-mês) que seria exigido para se concluir cada tarefa de engenharia de software para cada função de software. Taxas de mão-de-obra (isto é, custo/esforço unitário) são aplicados em cada uma das tarefas de engenharia de software. Muito provavelmente, a taxa de mão-de-obra irá variar para cada tarefa. O pessoal de nível sênior envolver-se-á fortemente na análise de requisitos e nas primeiras tarefas de realização de projeto; o pessoal de nível júnior (que é inerentemente menos dispendioso) envolver-se-á nas últimas tarefas de projeto, codificação e nos primeiros teste.

O custo e o esforço de cada função e tarefa de engenharia de software são computados como o último passo. Se a estimativa do esforço for realizada independentemente da estimativa LOC ou FP, teremos então duas estimativas para o custo e para o esforço que podem ser comparadas e reconciliadas. Se os dois conjuntos de estimativas demonstrarem razoável concordância, haverá uma boa razão para acreditar que as estimativas são confiáveis. Se, por outro lado, os resultados dessas técnicas de decomposição exibirem pouca concordância, será necessário levar a efeito a investigação e análise adicionais.

### **Estimativa de Custo**

O objetivo desta análise é calcular de maneira antecipada todo e qualquer custo que esteja associado ao sistema, tais como: construção, instalação, operação e manutenção.

O custo da construção é um tópico importante, visto que é graças a ele que sabemos o total de todas as pessoas envolvidas no desenvolvimento do sistema, tais como: burocratas, diretores, membros da comunidade usuária, consultores e programadores, membros da auditoria, do controle de qualidade ou da equipe de operações.

O custo de instalação do sistema é um projeto simples que podemos entregar em disquetes ou CD-ROMs e a instalação fica por conta do próprio usuário. Porém, em caso de sistemas grandes, o processo de instalação é mais complexo e envolve outros fatores, tais como: custo de treinamento do usuário, custo de conversão de banco de dados, custo de instalação do fornecedor, custo da aprovação legal, custo do processamento paralelo, custo da equipe de desenvolvimento durante a instala-



## **Garantia da Qualidade de Software**

Quando se fala em Garantia de Qualidade de Software (SQA) em um nível mais alto, parece que estamos nos referindo somente à avaliação do software com base em certos atributos, como funcionalidade, desempenho, adaptabilidade, etc.

No entanto, a garantia de qualidade de software vai além da qualidade específica do software, ela também inclui a qualidade do processo utilizado para desenvolver, testar e liberar o software.

O Software Quality Assurance (SQA) consiste nos meios para garantir a qualidade do software a ser lançado, monitorando os métodos e processos de engenharia de software. O SQA abrange todo o ciclo de vida de desenvolvimento de software incluindo: gerenciamento de requisitos, design de software, codificação, teste e gerenciamento de release.

## **Atributos De Qualidade De Um Software**

Um Software pode ser avaliado pela qualidade com base nas seguintes seis características:

### **Funcionalidade**

A funcionalidade do software é o conjunto de funções que o software fornece. O software deve fornecer funções apropriadas conforme os requisitos e essas funções devem ser implementadas corretamente.

O software deve ter interoperabilidade, o que significa a eficácia com que o software interage com outros componentes do sistema.

Deve estar em conformidade com as leis e diretrizes e deve tratar as transações relacionadas a dados com a máxima segurança.

### **Confiabilidade**

A confiabilidade do software é sua capacidade de executar sob condições específicas por um período de tempo definido. Implica também a capacidade do software em suportar as falhas de seus componentes.

O software é confiável com base em sua maturidade que é a frequência das falhas e capacidade de recuperação, que é a capacidade do software para ficar totalmente operacional após uma falha.

### **Usabilidade**

A usabilidade do software é sua facilidade de uso. Também se refere à facilidade com que um usuário pode entender as funções do software e quanto esforço é necessário para que os usuários entendam as suas funções.

### **Eficiência**

A eficiência do software depende das práticas de arquitetura e de codificação que foram seguidas durante o seu desenvolvimento.

### **Manutenibilidade**

A capacidade de manutenção do software depende da complexidade e legibilidade do código. Também se refere à facilidade de identificar e corrigir uma falha no software. O software é analisado com base na facilidade de identificar a principal causa de falha e a facilidade de modificação do código para remover essa falha.

A facilidade de manutenção também depende de sua testabilidade, ou seja quanto esforço é necessário para testar o sistema.

### **Portabilidade**

A portabilidade do software é sua capacidade de se adaptar às mudanças em seu ambiente. Define com que facilidade um sistema se adapta a quaisquer alterações feitas nas suas especificações. Também inclui a facilidade de instalação do software e como é fácil substituir um componente do sistema em um determinado ambiente.

Para garantir que um software tenha um bom desempenho nesses atributos de qualidade, precisamos dos seguintes componentes de garantia de qualidade de software.

### **Componentes De Garantia De Qualidade De Software**

A garantia de qualidade de software possui as seguintes classes de componentes:

Componentes pré-projeto

Os componentes do pré-projeto garantem que os recursos necessários para o projeto, o cronograma e o orçamento estejam claramente definidos. O plano de desenvolvimento e garantia de qualidade foi claramente determinado. Os componentes são os seguintes:

Plano de Desenvolvimento

Plano de qualidade

Cronograma

Recursos necessários (hardware e recursos humanos)

Avaliações de risco

Metodologia do Projeto

Componentes do ciclo de vida do projeto

Um ciclo de vida do projeto é geralmente composto por dois estágios. O primeiro é o estágio de desenvolvimento e depois vem o estágio de manutenção da operação.

No estágio de desenvolvimento, os componentes de SQA ajudam a identificar os erros de design e programação.

Os componentes SQA para o estágio de manutenção da operação incluem os componentes do ciclo de vida de desenvolvimento, juntamente com componentes de manutenção especializados, destinados a melhorar as tarefas de manutenção.

Os componentes do ciclo de vida do projeto incluem:

Revisões

Opiniões de Especialistas

Teste de Software

Manutenção de Software

Garantia de Qualidade de subcontratados



Componentes de prevenção e melhoria de erros de infra-estrutura

O principal objetivo desses componentes é a prevenção de falhas de software e minimização da taxa de erros. Esses componentes incluem:

Procedimentos e instruções de trabalho

Modelos e listas de verificação

Treinamento de equipe, retenção e certificação

Ações preventivas e corretivas

Gerenciamento de configuração

Controle de documentação

Componentes de gerenciamento de qualidade de software

Essa classe de componentes consiste em controlar as atividades de desenvolvimento e manutenção. Esses componentes estabelecem o controle gerencial dos projetos de desenvolvimento de software. O controle gerencial visa impedir que o projeto ultrapasse o orçamento e que seja entregue com atraso.

Os componentes de controle de gerenciamento incluem:

Controle de progresso do projeto

Métricas de qualidade de software

Custos de qualidade de software

Componentes de padronização, certificação e avaliação de SQA

Os componentes visam implementar padrões gerenciais e profissionais dentro da organização. Esses componentes ajudam a melhorar a coordenação entre os sistemas de qualidade da organização e estabelecem padrões para o processo do projeto.

Os componentes incluem:

Padrões de gerenciamento de qualidade

Padrões de processos de projeto

Organização para SQA - os componentes humanos

O principal objetivo desta classe de componentes é iniciar e suportar a implementação de componentes SQA, identificar quaisquer desvios dos procedimentos e métodos SQA pré-definidos e recomendar melhorias. A equipe organizacional do SQA inclui gerentes de teste, testadores, unidade SQA, comitê SQA e membros do fórum SQA.

### **Ferramentas de Garantia de Qualidade de Software**

Existem várias ferramentas de controle de qualidade que ajudam na garantia da qualidade do software. Existem diferentes ferramentas de controle de qualidade necessárias para os diferentes fins. Para uma garantia abrangente de qualidade de software, você precisará da seguinte categoria de ferramentas de controle de qualidade, também conhecida como software de controle de qualidade.

### **Infraestrutura**



## **Desenvolvimento de Aplicações WEB**

Conforme já é sabido, os serviços WEB utilizam a infra-estrutura de redes existente para disponibilização de informações e aplicações para o usuário. Para o cliente e o servidor WEB a topologia da rede é irrelevante pois isto é tratado pelo protocolo TCP/IP.

No nível de serviços, os elementos são organizados da seguinte forma: de um lado está o cliente WEB, ou browser, que solicita dados ao servidor WEB, recebe as respostas, formata a informação e a apresenta ao usuário. Do outro lado, está o servidor WEB que recebe as requisições, lê os dados (páginas HTML) do disco e as retorna para o cliente.

Esta é a forma original de funcionamento da WEB e que proporciona apenas páginas de conteúdo estático, ou seja, uma vez programadas, apresentam sempre a mesma informação.

A forma encontrada para modificar esta situação e permitir a criação de páginas dinâmicas foi a seguinte: o usuário entra com informações através do browser utilizando formulários HTML. O browser repassa as informações ao servidor WEB que executa um programa transferindo-lhe as informações vindas do cliente. O programa remoto (serve-side gateway program) trata as informações e retorna uma página HTML criada dinamicamente. Esta página é passada ao servidor que a entrega ao cliente. O padrão para comunicação entre o servidor WEB e o "Server-side gateway program" é conhecido como CGI (Common Gateway Interface).

A Interface CGI explicita como os dados devem ser passados do servidor WEB para o programa e como o este deve devolver os dados ao servidor WEB. Além disto, especifica variáveis de ambiente que devem ser criadas pelo servidor e que podem ser acessadas pelo programa. Estas variáveis contêm o endereço IP do usuário remoto, o tipo de browser que ele utiliza, dados para autenticação do usuário, dentre outros.

Os programas escritos para ser executados por servidores WEB são também conhecidos como scripts CGI. Os scripts CGI implementam a lógica do negócio e, muitas vezes, o acesso aos dados de uma aplicação WEB, pois podem acessar dados armazenados no servidor ou fazer chamadas a um servidor de banco de dados local ou remoto, permitindo o acesso às informações da empresa.

Nesta arquitetura, tem-se programas sendo executados no servidor WEB e em outras máquinas da rede, como o servidor de banco de dados. Só falta agora serem colocados programas para executar no cliente. A execução de programas no cliente torna o browser mais independente do servidor. Sem isto, qualquer operação realizada pelo cliente tem que ser validada e executada pelo servidor, pois o browser não possui nenhuma inteligência.

Os programas que executam no browser, normalmente, são scripts (VBScript ou JavaScript) que tem a capacidade de perceber os eventos causados pelo usuário e responder de forma apropriada. Eles são embutidos no código HTML e o seu código fonte pode ser visualizado pelo usuário, pois não é um código compilado. Os scripts interagem muito com todos os elementos que formam uma página HTML.

Outro tipo de programa que pode ser executado no cliente são os programas escritos na linguagem JAVA, que será descrita em maiores detalhes mais a frente.

Em JAVA, é possível escrever aplicações completas, inclusive acessando bancos de dados relacionais independentemente do servidor WEB, bem como é possível implementar os níveis de interface com o usuário e de lógica do negócio, utilizando um servidor de banco de dados para implementar o nível de acesso aos dados.

### **A Plataforma Ativa da Microsoft**

A Microsoft, como parte da sua estratégia para Internet/Intranet, está lançando no mercado a idéia da Plataforma Ativa Microsoft. Segundo a própria fabricante, pelo menos na visão do seu departamento de Marketing, trata-se de uma arquitetura aberta de software, para disponibilizar conteúdo rico e aplicações através da Internet e Intranets.

Tecnicamente, a plataforma ativa combina uma serie de tecnologias para criar sistemas de software distribuídos baseados em componentes, ou objetos. Cada componente é um pedaço de software que possui informações e implementa serviços necessários a uma ou mais aplicações, podendo residir em

uma máquina da rede e ser acessado por outros de forma independente da sua localização. Os componentes implementam serviços dos vários níveis de uma arquitetura multitier: apresentação, lógica do negócio e acesso aos dados.

O objetivo maior desta tecnologia é permitir, com o passar do tempo, o desenvolvimento de aplicações em escalas de tempo bastante reduzidas. Todo o desenvolvimento será calcado na idéia da reutilização de componentes já existentes para a criação de novas aplicações. Antes de implementar algo novo, os projetistas de software devem procurar verificar se já não existem componentes prontos para atender aquela situação específica. Estes componentes podem estar disponíveis dentro da própria organização ou podem ser adquiridos de terceiros.

Os componentes podem estar presentes nas interfaces com o usuário (browser), implementando a lógica do negócio junto ao servidor WEB ou distribuídos pela rede e realizando atividades de acesso aos dados, utilizando bases de dados em qualquer formato.

Este conjunto de tecnologias que vão permitir o desenvolvimento de aplicações distribuídas baseadas no conceito da WEB é normalmente chamado de ActiveX. Os componentes que aderem a estas especificações são conhecidos como componentes ActiveX.

O padrão para construção de componentes ActiveX é especificado pelo Component Object Model (COM). O COM especifica uma série de serviços (interfaces) que devem ser implementadas pelos componentes para que eles possam se integrar com facilidade em qualquer tipo de aplicação. Um objeto COM pode, por exemplo, ser incorporado dinamicamente a uma aplicação como o Microsoft Word ou Internet Explorer, que são conhecidos como "containers" de objetos.

### **A Plataforma Ativa Microsoft Utiliza Recursos já Existentes na WEB**

A arquitetura proposta é a seguinte: de um lado está o chamado cliente ativo, que é um browser, com a capacidade de apresentar páginas HTML, executar programas em JAVA, conter componentes ActiveX e executar scripts (VBScript ou JavaScript) que controlam todos estes elementos. As linguagens de script tem um papel extremamente importante, pois elas formam a cola que une os componentes (ActiveX e JAVA) para criar uma aplicação para o usuário. São elas que ativam os componentes ActiveX/JAVA solicitando-lhes serviços de acordo com as escolhas dos usuários.

Do outro lado, está o servidor ativo, que é um servidor WEB com a capacidade de ler e entregar páginas HTML estáticas e executar scripts que utilizam componentes ActiveX para realizar atividades.

Deve ser observada a simetria entre o cliente ativo e o servidor ativo. Ambos tem a capacidade de lidar com os mesmos elementos, sendo que o servidor não executa nenhuma atividade de apresentação de interface para o usuário. Os componentes que são utilizados pelo servidor implementam a lógica do negócio e o acesso aos dados. Já no cliente ativo, executam os componentes que implementam a interface da aplicação.

Os componentes ActiveX podem residir no mesmo servidor ou podem ser distribuídos pela rede. O padrão para criação de objetos distribuídos é conhecido como DCOM (Distributed Component Object Model).

Espera-se que esta tecnologia permita aos desenvolvedores de software criar aplicações distribuídas, utilizando um browser, que é independente de arquitetura, e reutilizando componentes pré-programados.

### **Linguagens de Script**

As linguagens de script (script languages) são assim chamadas por serem linguagens de programação simples sem muitas das restrições associadas às linguagens mais complexas. Nas linguagens de script, o tratamento dos tipos das variáveis é bastante simplificado para o programador, pois estas não possuem tipos fixos e assumem aqueles dos valores que lhes são atribuídos. Elas são interpretadas em tempo de execução ao invés de compiladas para posterior execução.

Muitas vezes, as linguagens de script são subconjuntos mais simples de linguagens existentes, como é o caso do VBScript e do JavaScript. Ambas tem o mesmo potencial de utilização na WEB apesar de usarem sintaxes diferentes.

Programas feitos em VBScript/JAVAScript podem ser inseridos em qualquer página HTML através do comando

Tratam-se de objetos programados de acordo com o padrão COM (Component Object Model) já mencionados. Este padrão define uma serie de interfaces (serviços) que os objetos devem implementar para que possam ser reutilizados em aplicações diversas. O padrão COM utiliza o formato de objetos OLE para implementar as suas interfaces.

Os objetos ActiveX expõem para o mundo externo um conjunto de atributos e operações (serviços) que ele pode realizar. Pelo fato de seguirem o padrão COM eles podem ser executados em qualquer aplicação que suporte este padrão. Hoje em dia, é possível utilizar objetos COM no Internet Explorer, Servidor WEB IIS ou no desktop do próprio WINDOWS.

Nem todas as interfaces especificadas no padrão COM são implementadas nos componentes ActiveX. Assim, estes tornam-se menores e mais rápidos, de forma que a sua transferência na rede e execução são otimizadas.

Os componentes ActiveX podem ser embutidos em páginas HTML e seus atributos e serviços podem ser acessados através das linguagens de script, como o VBScript. Os componentes ActiveX também podem ser utilizados no servidor WEB através das linguagens de script. A diferença maior entre os componentes ActiveX que rodam no browser dos que rodam no servidor é que os primeiros podem possuir interface com o usuário, enquanto que os últimos não devem possuir este recurso.

Cada objeto ActiveX, inserido em uma página HTML, recebe uma identificação e é acessado pelo VBScript através desta identificação, assim como qualquer campo de um formulário. O tratamento dos eventos que ocorrem nos componentes ActiveX também pode ser feito em VBScript na própria página HTML. O tratamento destes eventos é similar ao tratamento de eventos no Visual Basic: basta definirmos funções que tenham o mesmo nome do componente seguido do evento a ser tratado.

O objeto acima é identificado pelo nome CommandButton3. A função abaixo trata o evento do clique do mouse sobre o botão.

Quando um componente ActiveX é encontrado em uma página HTML, o browser verifica se aquele componente já encontra-se instalado na máquina do cliente e se a versão existente é compatível com a versão especificada na página. Se estas condições são satisfeitas, o browser utiliza o componente já instalado, sem a necessidade de transferências pela rede.

Caso contrário, o browser inicia a transferência do objeto. Os componentes ActiveX podem ser disponibilizados em arquivos DLL (Bibliotecas de Ligação Dinâmica) ou arquivos OCX (ActiveX Component Object). Este arquivo é instalado na máquina do usuário, e o objeto é instanciado e colocado na apresentação da página HTML.

Os componentes ActiveX podem realizar qualquer operação na máquina do usuário. Por isto, algumas medidas de segurança devem ser tomadas para evitar possíveis desastres. Os componentes ActiveX possuem uma assinatura que identifica a sua procedência. O usuário tem a capacidade de aceitar ou rejeitar componentes ActiveX de procedência duvidosa. Este controle é feito através de opções de configuração do browser.

Os componentes ActiveX podem ser implementados em qualquer linguagem de programação, seja JAVA, C++ ou Visual Basic, desde que atendam aos padrões especificados e implementem as interfaces necessárias.

Os componentes ActiveX podem ser reutilizados em inúmeras aplicações, trazendo vantagens não só no desenvolvimento de software, como também na sua manutenção. O desenvolvimento de aplicações tende a tornar-se uma atividade de reutilização do que de criação de novos componentes.

A manutenção destes sistemas pode ser feita de forma localizada através da alteração de seus componentes. A alteração de um componente que é utilizado em varias aplicações faz com que todas as aplicações sejam automaticamente atualizadas. O browser garante que serão utilizadas sempre as ultimas versões de cada componente.

Os componentes ActiveX podem ser distribuídos pela rede e utilizados como se fossem objetos locais. O padrão DCOM (Distributed Component Object Model) especifica as interfaces que os objetos devem implementar para poder funcionar de forma distribuída. A localização dos objetos na rede deverá ser transparente para as aplicações que os utilizam.

Cada componente ActiveX pode ser especializado para a realização de atividades referentes a cada um dos níveis da arquitetura cliente servidor multitier. Podemos ter componentes que implementam serviços de interface, lógica de negócios e acesso aos dados.

Os componentes de interface são objetos com representação visual para interação com o usuário. Os objetos de interface serão utilizados no browser.

Os objetos que implementam a lógica do negócio serão utilizados por diversas aplicações, pois é neles que reside a "inteligência" dos programas. Alterações nas regras do negócio da empresa deverão ser feitas nestes objetos e estarão, automaticamente, disponíveis para todas as aplicações. Os objetos de negócio poderão residir no browser, no servidor Web ou poderão estar distribuídos pela rede.

Os objetos que implementam o acesso aos dados deverão esconder os detalhes de formato do armazenamento dos dados para os objetos de negócio, que serão os seus principais usuários. Os dados poderão vir de fontes diversas, tais como bancos de dados relacionais, arquivos texto, arquivos VSAM e outros, de forma transparente. Os objetos de acesso aos dados poderão residir no servidor WEB ou poderão estar distribuídos pela rede.

JAVA é uma linguagem de programação orientada a objetos que permite a criação de aplicações que são independentes de arquitetura. Ou seja, é possível executar o mesmo programa JAVA (código binário) em máquinas com hardware e sistemas operacionais diferentes.

Uma aplicação em JAVA executa em cima de uma máquina virtual, que é chamada de Java Virtual Machine. A máquina virtual JAVA especifica os componentes de um computador, incluindo memória, registradores, pilha, etc. Os programas em JAVA, quando compilados, geram código que seria executado nesta máquina. Este código compilado é conhecido como byte code (Java Byte Codes).

Como não existe nenhum hardware que seja igual a máquina virtual JAVA, esta máquina é implementada em software e embutida nos principais browsers da atualidade. Existem também implementações da máquina virtual JAVA, independente de browsers, para diversas plataformas. Isto faz com que um programa JAVA execute hoje nas principais plataformas existentes.

Para encaixar as applets JAVA (aplicações em JAVA que rodam em um browser) no paradigma ActiveX, elas estão sendo tratadas como componentes ActiveX. As applets podem ter atributos e operações disponíveis para ser utilizadas pelas linguagens de script.

Até aqui, foram discutidos os elementos que estão presentes do lado do cliente WEB, o browser. Foram vistas as linguagens de script, representadas pelo VBScript, os componentes ActiveX e as applets JAVA. A partir da próxima seção, serão vistos os elementos que trabalham do lado do servidor para compor as aplicações WEB.

Quando um formulário HTML é preenchido por um usuário e os dados são enviados ao servidor WEB, algum programa deve ser executado para tratar aquela informação específica. O valor do atributo ACTION do tag indica qual programa deve ser utilizado.

O Common Gateway Interface especifica como os dados vindos do cliente WEB devem ser passados ao programa (script CGI) e como este programa deve retornar o resultado ao servidor WEB, para que este repasse as informações ao cliente. O script CGI pode ser escrito em qualquer linguagem de programação, desde que o código gerado possa ser executado na máquina do servidor.

A solução CGI para interatividade na WEB não é adequada para servidores muito requisitados. Cada execução de um script CGI implica na criação de um novo processo no sistema operacional. Para a criação de um novo processo várias tarefas (alocação de memória, criação de estruturas internas etc.) devem ser realizadas, o que consome bastante tempo da máquina. Além disto, a transferência de dados entre processos, no caso o servidor WEB e o script CGI, também não é muito eficiente.

Para minimizar este problema, a Microsoft, criou um servidor WEB que pode carregar módulos adicionais, sob a forma de bibliotecas de ligação dinâmica (DLL). Estas DLLs uma vez carregadas, ficam na memória do servidor e funcionam como se fossem uma parte do software servidor WEB. As novas chamadas para este modulo serão atendidas com mais rapidez e menos consumo de máquina pois os mesmos já estarão carregados e fazendo parte do processo servidor WEB. A transferência de dados também é otimizada, pois tudo acontece dentro de um mesmo espaço de endereçamento.

A criação destes módulos é feita através da implementação de uma interface padrão de comunicação com o servidor WEB. Este padrão de programação é conhecido como ISAPI.

Os módulos ISAPI podem funcionar como scripts CGI, recebendo dados do cliente, processando-os e retornando novas páginas HTML (ou outros tipos de documentos). Para o cliente WEB tudo continua funcionando como se fossem scripts CGI.

Os módulos ISAPI podem também funcionar como filtros, que tratam os dados que transitam entre o cliente e o servidor WEB. Podem ser notificados do acontecimento de determinados eventos e entrar em ação quando estes eventos acontecem.

Um filtro pode, por exemplo, solicitar ser notificado quando um cliente pedir um determinado tipo de arquivo. O filtro pode, então, tratar o arquivo interpretando scripts ou criptografando um documento antes do envio do mesmo ao cliente.

O ASP (Active Server Pages), que será visto na próxima seção, funciona como um filtro que trata todas as solicitações de páginas com a extensão de arquivo .asp. As páginas com extensão .asp são páginas HTML contendo scripts (VBScript/JAVAScript) que são interpretados e executados pelo filtro antes de serem entregues aos clientes. Além disso, este modulo disponibiliza outros componentes ActiveX para que sejam acessados pelos scripts, tais como objetos de acesso a dados.

Este mesmo esquema pode ser utilizado para a criação de outros filtros que interpretem diferentes linguagens de script como, por exemplo, PERL, que é uma linguagem muito utilizada para a criação de scripts CGI e que tem uma grande interação com o sistema operacional e grande capacidade de operar com strings.

Active Server Pages é o nome dado à tecnologia que permite a execução de scripts embutidos em páginas HTML no servidor WEB. Os scripts podem ser escritos em VBScript ou JAVAScript. Os scripts tem acesso a componentes ActiveX que residem no servidor.

A execução destes scripts fica a cargo de um filtro ISAPI que intercepta todas as requisições de arquivos com extensão .asp. As páginas são lidas pelo servidor e, antes de serem enviadas ao cliente, são tratadas pelo filtro ASP, que interpreta e executa os comandos encontrados na página. O resultado final é entregue ao cliente.

Os arquivos com extensão .asp quando solicitados, podem executar programas (server side scripts) embutidos no documento e utilizar serviços de componentes ActiveX. Estes programas geram as páginas HTML de forma dinâmica, podendo acessar bases de dados, que são repassadas ao browser.

Neste material, foi apresentada toda uma nova tecnologia de desenvolvimento de aplicações que está surgindo no mundo da informática. Os paradigmas estão sendo modificados a uma velocidade muito grande. Já encontram-se disponibilizados elementos que nos permitem criar aplicações realmente distribuídas e orientadas a objetos.

As maiores vantagens da utilização desta tecnologia são a possibilidade de reutilização de código em larga escala, utilizando os objetos, e aproveitamento do poder computacional, que está distribuído em redes de computadores de todo o mundo. Neste ambiente, todo software desenvolvido tem a capacidade de ser executado e utilizado em múltiplas plataformas de hardware e software, com total transparência para o usuário.

Em uma próxima oportunidade, será apresentada a proposta da Sun Microsystems e seus parceiros (IBM, Borland, Apple etc.) para o desenvolvimento de software distribuído baseado em componentes, o JAVA Beans.

## Interfaces Gráficas

Os produtos de software atuais são muito mais complexos do que eram há apenas alguns anos atrás.

Paralelamente, nos últimos 10 anos o computador passou a ser utilizado por usuários leigos em computação, e a experiência desses usuários com produtos como planilhas, editores de texto, e programas de apresentação os convenceu que um programa de computador pode ser fácil de usar.

Parte dessa facilidade foi proveniente da utilização de interfaces gráficas homem-computador com elementos do tipo janelas, menus, mouse e ícones, que representam um grande avanço em tornar os computadores pessoais mais amigáveis.

Embora fáceis de usar, aplicativos com interfaces gráficas são difíceis de desenvolver, envolvendo paradigmas não familiares para grande parte dos projetistas de software tais como orientação a objetos e processamento dirigido por eventos.

Estão disponíveis no mercado várias ferramentas para desenvolvimento de interfaces gráficas que oferecem uma gama de funcionalidade bastante variável, e a escolha da ferramenta para um projeto pode ter consequências não somente em sua fase de desenvolvimento, mas também na sua evolução, uma vez que algumas ferramentas permitem o desenvolvimento de código fonte portátil, isto é, que pode ser transportado para vários ambientes sem necessidade de modificações.

Por sua vez, o teste de aplicativos com interfaces gráficas também não é uma tarefa simples, e realizar tais testes manualmente é uma tarefa tediosa e pouco confiável. Encontram-se disponíveis no mercado algumas ferramentas que permitem automatizar a fase de testes, mas que apresentam preço e funcionalidade bastante díspares, não sendo fácil a escolha de uma delas.

Embora o uso do ambiente ms-windows já esteja consolidado no Brasil, a indústria nacional de software somente agora começa a produzir produtos de modo sistemático para esse ambiente.

É dentro desse contexto que o Citi, através do programa de qualidade e produtividade em software, decidiu montar um grupo com o objetivo de absorver tecnologia avançada para a produção de interfaces gráficas portáteis de alta qualidade e repassar essa tecnologia para a indústria nacional de software.

Neste documento procuramos, de forma simplificada, apresentar a experiência adquirida e as perspectivas atuais do nosso grupo, abordando o trabalho realizado nas áreas de portabilidade, testes de aplicativos e metodologia para desenvolvimento de interfaces gráficas.

Atualmente a Microsoft detém a maior parcela da base de sistemas operacionais instalados com a dupla ms-dos e ms-windows. Este fato faz com que as plataformas da Microsoft sejam o principal alvo dos desenvolvedores de software, acarretando uma grande concorrência. Embora uma pequena fração da base instalada de sistemas operacionais não seja Microsoft, o tamanho do mercado mundial de software (aproximadamente 200 bilhões de dólares/ano) torna não desprezível o oferecimento de aplicativos para outras plataformas. Uma empresa que domine uma tecnologia que permita desenvolver produtos para o ambiente Microsoft sem abandonar as outras fatias do mercado, passa a ter uma importante vantagem competitiva em suas mãos e, portanto, portabilidade torna-se uma palavra-chave nesse contexto.

A questão portabilidade encontra-se razoavelmente equacionada nas áreas de linguagens de programação, acesso a dados e comunicações através dos padrões ANSI C, SQL e TCP/IP ou OSI/ISO, entre outros.

O mesmo não se pode afirmar no caso de interfaces gráficas pois cada ambiente (ms-windows ou os/2, por exemplo) possui um conjunto de recursos básicos, o toolkit, para a implementação de interfaces gráficas, todos desenvolvidos de forma independente e não padronizada. A falta de padronização acarreta a necessidade de se reprogramar totalmente os módulos de controle da interface gráfica quando



se transporta um aplicativo para um ambiente distinto de seu ambiente alvo inicial. Essa reprogramação pode ser extremamente cara pelo fato de, em geral, a maior parte do código de um aplicativo tratar sua interface gráfica.

Quando se consideram os diversos ambientes gráficos percebe-se que a funcionalidade básica oferecida é, em grande parte, idêntica, sendo que as principais diferenças estão relacionadas com a forma pela qual elementos da interface como barras de rolagem, botões e janelas são representados graficamente.

Uma vez que a maior parte dos ambientes gráficos apresenta um conjunto comum de objetos com funcionalidades idênticas, uma das soluções para o problema de portabilidade de interfaces gráficas é a definição e implementação de uma interface de programação (api) padronizada que suporte esse conjunto comum e que o mapeie nos toolkits originais (figura 1). Dessa forma, um aplicativo desenvolvido com a interface de programação padronizada pode ser facilmente transportado para diversas plataformas, preservando sempre a aparência e funcionalidade, o look-and-feel, da plataforma na qual é executado. Várias empresas desenvolveram ferramentas baseadas em toolkits portáteis tais como zinc, zapp e xvt.

Após ter realizado um levantamento das ferramentas disponíveis no mercado, o grupo escolheu o xvt para ganhar experiência no uso de tecnologia para a produção de interfaces gráficas portáteis. Essa ferramenta fornece uma api padrão que permite a implementação de tais interfaces e seu transporte entre os principais ambientes gráficos atuais (ms-windows, ms-windows nt, presentation manager OS/2, OSF/MOTIF e macintosh).

A ferramenta apresenta características de prototipação rápida, permitindo que seus usuários simulem, na fase inicial do desenvolvimento, a execução da interface do aplicativo, possibilitando o reconhecimento de possíveis problemas de projeto [argollo 1995].

O XVT é composto por dois módulos principais: um editor de interfaces gráficas e uma biblioteca de funções multi-plataforma. O editor de interfaces gráficas permite a criação dos objetos da interface de modo interativo e gera um arcabouço da aplicação na linguagem c/c++ com o código necessário para o controle da interface [xvt 1994]. A biblioteca multi-plataforma dispõe de primitivas para a manipulação dos objetos básicos de interfaces gráficas: janelas, menus, controles (botões de rádio, campos de edição, tabelas, planilhas, etc.), e funções para operações gráficas, de impressão, manipulação do clipboard e de fontes de caracteres, etc.

Após ter utilizado o xvt para desenvolver várias aplicações e de tê-las transportado da plataforma ms-windows para a plataforma osf/motif e vice-versa podemos afirmar o seguinte:

é possível desenvolver interfaces gráficas portáteis com a tecnologia existente desde que seja utilizada uma metodologia voltada para a obtenção de portabilidade;

Normalmente as ferramentas não são orientadas para domínios específicos e, por isso, funcionalidades do tipo acesso a base de dados ou geração automática de relatórios devem ser implementadas através de pacotes específicos;

O código gerado pelas ferramentas não impacta o desempenho da aplicação final;

As ferramentas não permitem abstrair todos os detalhes de cada plataforma; por exemplo, o ambiente osf/motif direciona saídas impressas para um arquivo postscript, enquanto o ms-windows as envia para o gerente de impressão.

Embora as afirmações feitas acima tenham sido observadas a partir do uso de uma única ferramenta, o xvt, em apenas dois ambientes distintos (ms-windows e OSF/MOTIF), não existe razão técnica que

as invalide quando consideradas outras ferramentas com as mesmas características ou outros ambientes.

O teste de aplicativos com interfaces gráficas homem-computador é uma atividade difícil de realizar porque a complexidade desse componente de software, notada através de características como processamento multi-tarefa, manipulação direta de objetos com resposta ao usuário em tempo-real e processamento dirigido por eventos, é responsável por uma explosão combinatória de cenários com as possíveis interações homem-computador que devem ser testadas.

Embora existam poucas referências sobre métodos e estratégias para teste de interfaces gráficas na literatura técnica, diversas ferramentas cujo objetivo é a automação desses testes estão disponíveis no mercado.

Essas ferramentas utilizam o método "gravar e repetir" para capturar tanto as ações do usuário sobre a interface quanto os resultados dessas ações. O conjunto de ações e resultados capturados são mapeados numa linguagem de programação e gravados em arquivos denominados scripts de teste. As ferramentas permitem a repetição das ações do usuário e a comparação dos resultados obtidos através da execução automática dos scripts de teste.

Essa abordagem do problema exige que a interface gráfica a ser testada esteja operacional e executando de modo confiável para que o usuário possa gravar suas ações nos diversos cenários de interação. Essa é uma limitação importante porque a fase de testes fica relegada ao final do desenvolvimento, quando talvez já não haja tempo disponível para realizá-la de modo adequado. Apesar disso, o fato das ferramentas repetirem os scripts de teste já gravados permite que testes regressivos sejam realizados sempre que o software tenha sido modificado, o que é um passo importante para a obtenção de produtos robustos.

Com o objetivo de permitir o desenvolvimento de scripts de teste sem que a interface esteja implementada, foram criados mecanismos que permitem a especificação das ações do usuário em termos dos objetos presentes na interface gráfica. Esse novo enfoque do problema tornou possível a geração de casos de teste a partir da especificação da interface gráfica, permitindo que a fase de testes seja realizada em paralelo com a fase de implementação.

O grupo de interfaces gráficas selecionou e adquiriu a ferramenta qpartner que incorpora uma tecnologia moderna para testes automáticos de aplicativos com interface gráfica [segue 1994].

O primeiro passo para o desenvolvimento de casos de teste com essa ferramenta é a declaração dos elementos da interface. Esse passo permite que os objetos presentes na interface sejam referenciados por nomes mnemônicos e facilita o desenvolvimento de scripts de teste robustos, uma vez que mudanças de posição, tamanho e rótulo dos objetos na interface não irão impactar de modo significativo os testes já desenvolvidos.

O passo seguinte é o desenvolvimento de casos de teste na linguagem 4test que é orientada a objetos e possui uma sintaxe bastante semelhante à da linguagem c. A linguagem 4test possui comandos que referenciam os objetos da interface através de seus nomes mnemônicos anteriormente declarados. Por exemplo, o seguinte comando

```
gotoline.ok.click ();
```

especifica um clique no botão "ok" da janela "gotoline" do aplicativo sendo testado.

Com a ferramenta, os scripts de teste podem ser desenvolvidos de forma manual, sem a necessidade de o aplicativo estar operacional, ou automaticamente, com ajuda de um módulo que captura as ações do usuário, caso o aplicativo tenha uma versão executável disponível. Os scripts de teste desenvolvidos controlam a execução do aplicativo em análise, simulando a interação com o usuário. A ferramenta se

encarrega de comparar os resultados do programa testado com os resultados esperados. Tal comparação pode verificar o conteúdo de mensagens de erro apresentadas em janelas de diálogo, conteúdo de arquivos ou de áreas gráficas, etc.

O uso efetivo das ferramentas para teste não prescinde do emprego de métodos viáveis e bem definidos que permitam criar casos de teste que cubram todos os objetos da interface gráfica de uma aplicação e todos os possíveis cenários de interação homem-computador. O grupo de interfaces gráficas tem um projeto em andamento cujo objetivo é investigar técnicas e desenvolver um método para teste de interfaces gráficas [villaça 1995].

O conceito de sistema computacional modificou-se substancialmente com o surgimento das interfaces gráficas. Sua definição e seu processo de desenvolvimento tornaram-se mais abrangentes, a partir da intensa proliferação e uso dos computadores pessoais. Hoje, com os computadores constituindo-se em bens de consumo há um redirecionamento das indústrias de software para atender novos nichos de mercado, compostos por faixas mais amplas e diversificadas de usuários.

As preocupações atuais da indústria de software são diferentes daquelas de anos atrás. Basicamente, hoje há uma preocupação crescente tanto em relação à apresentação do produto quanto às necessidades dos usuários, que passam dessa maneira a ter um papel central no desenvolvimento de interfaces gráficas [norman 1986, laurel 1990, heckel 1991].

A abordagem para desenvolvimento centrada no usuário provocou uma ampliação do conceito tradicional de desenvolvimento de sistemas para além dos requisitos funcionais do aplicativo, incluindo, agora, fatores subjetivos relacionados com o conjunto de interações do usuário, essenciais para o design de interfaces com qualidade [winograd 1985]. Em linhas gerais, esta abordagem destaca a inadequação dos métodos tradicionais para desenvolvimento de sistemas com interfaces gráficas, explicitando uma questão acerca da visão errônea atribuída ao design e ao desenvolvimento de interfaces [kay 1990, norman 1990, hix 1993].

"a postura de que o processo de design da interface com o usuário não é necessário, podendo ser realizado simplesmente durante a finalização total do desenvolvimento, é ingênua. Infelizmente essa atitude tem prevalecido e tem contribuído para a imagem de que as indústrias de software desenvolvem sistemas não usáveis." [browne, 1994, p. 20]

Vários estudos mostram o impacto financeiro e a importância das interfaces gráficas. Em [browne 1994], por exemplo, é apresentado um quadro quantitativo onde cerca de 55% do código total pertence aos elementos de interface, aproximadamente 30% do custo total de desenvolvimento do aplicativo está concentrado na interface e 60% do custo na fase de manutenção (responsável por 60% do custo total do aplicativo) está relacionado com alterações na interface.

O resultado deste e de outros estudos quantitativos oferecem argumentos objetivos sobre o papel e o lugar das interfaces gráficas no processo de desenvolvimento e justificam de forma indiscutível a necessidade de utilização de novos métodos de desenvolvimento, para diminuição do custo total do sistema e consequente aumento da produtividade da indústria de software. Diante da importância financeira e da complexidade do tema, muito tem sido discutido e vários métodos de desenvolvimento para aplicativos com interfaces gráficas foram propostos [nielsen 1993, hix 1993, browne 1994, carlshamre 1994].

Nos diversos métodos para desenvolvimento identificamos certas semelhanças, que formam uma base comum. A primeira semelhança está relacionada com a utilização de considerações originárias da abordagem centrada no usuário [norman 1986].

A segunda corresponde ao consenso de que uma boa interface deve ser transparente e permitir o acesso fácil à funcionalidade da aplicação. E a terceira está relacionada com a utilização do conceito

de usabilidade como parâmetro central para desenvolvimento do aplicativo e para as avaliações em relação à qualidade da interface. Mas, como elaborar e desenvolver uma interface?

Para tratar esta questão, segundo [Nielsen 1993], é necessário diminuir o caráter subjetivo da avaliação de uma interface, que normalmente gira em torno da usabilidade do aplicativo. Com este objetivo, Nielsen associa cinco atributos concretos e mensuráveis ao conceito de usabilidade, os quais auxiliam na quantificação de questões relacionadas às condições oferecidas aos usuários na utilização da funcionalidade do aplicativo. Os atributos de usabilidade são:

- facilidade de aprendizado
- Eficiência de uso
- Facilidade de memorização
- Baixa taxa de erros
- Satisfação subjetiva

Nielsen, a partir dessa associação, concentra-se no processo de desenvolvimento e estabelece o que denominou engenharia de usabilidade, onde são propostas atividades e heurísticas necessárias para a obtenção de um produto com alto grau de usabilidade.

Basicamente, na engenharia de usabilidade é apresentado um roteiro de onze fases que, independente do tipo de interface considerada, podem ser realizadas para o aumento da usabilidade. No roteiro, apresentado na tabela 1, há uma tentativa de se definir fatores críticos para a usabilidade do aplicativo, procurando-se, através de um desenvolvimento iterativo, balancear os cinco atributos de usabilidade para obter produtos de software com um grau de usabilidade satisfatório.

Nem sempre é possível realizar todas as fases do roteiro. Afinal, a pressão do tempo e dos custos são dependentes do projeto e da possibilidade econômica das empresas. Assim, nos parece necessário efetuar uma análise e possível redução do ciclo mostrado na tabela 1 adaptando-o ao contexto brasileiro.

O uso de interfaces gráficas como meio de comunicação com programas de computador é mandatório nos produtos de software atuais; longe de um simples modismo, esse mecanismo permite que usuários de programas de computador encontrem uma facilidade muito grande para desempenharem suas tarefas.

Dois fatores fundamentais para a implementação de uma boa interface gráfica são a tecnologia e a metodologia de trabalho empregadas.

A tecnologia existente no mercado para implementação de interfaces gráficas já se encontra bastante estável, como demonstra a experiência narrada neste trabalho no uso de ferramentas que tratam dos aspectos de portabilidade e de teste de aplicativos com interfaces gráficas.

Metodologias específicas para desenvolvimento de software com interfaces gráficas foram propostas recentemente e, embora importantes, ainda são pouco empregadas pela indústria de software. A importância dada aos futuros usuários dos produtos de software durante a fase de definição da interface surge como um importante ponto comum entre todas as metodologias que estão sendo propostas.

De forma semelhante ao que acontece em outras áreas, entre o surgimento de uma tecnologia e seu emprego industrial decorre um período de tempo ocasionado pelas dificuldades que as empresas encontram para dominar e incorporar a nova tecnologia. Neste ponto, a atuação de centros difusores de tecnologia é de extrema importância pois permite que empresas possam incorporar mais rapidamente novas tecnologias em seus processos produtivos.



### Técnicas e Estratégias de Teste de Software

Uma estratégia de testes de software descreve a abordagem geral e os objetivos das atividades de teste. Ela deve contemplar os níveis ou fases de teste, os tipos de testes a serem realizados e as técnicas para sua execução. A estratégia de testes de softwares também deve descrever com clareza os critérios para a conclusão dos testes e os critérios de sucesso a serem usados.

Responsáveis: Gerente de Projeto, Engenheiros de Software e Especialistas em Testes.

Importância: A importância das estratégias é evitar tempo desperdiçado, esforço desnecessário, infiltração de erros sem serem descobertos.

Passos: Primeiramente os testes focalizam um único componente ou um pequeno número de componentes relacionados, aplicados para descobrir erros nos dados e na lógica de processamento.

Em seguida testes de alto nível são executados para descobrir erros na satisfação dos requisitos dos clientes. E por fim os erros encontrados devem ser diagnosticados e corrigidos usando o processo de depuração.

Produtos: Especificação de Teste – documenta a abordagem da equipe de software para o teste, definindo o Plano de Testes – descreve uma estratégia global e um procedimento que define passos específicos de teste e os testes que serão conduzidos.

Antes de abordarmos cada estratégias de testes é importante veja o resumo que mostra:

### Estratégias de Teste de Software para arquitetura de Software Convencionais

O que	Quem	Como
Teste de Unidade – Código	Desenvolvedores	Focaliza cada componente individualmente, garantindo que ele funcione adequadamente como uma unidade. Usa técnicas de testes que exercitam caminhos específicos na estrutura de controle de um componente, para garantir completa cobertura e máxima detecção de erros. Teste Caixa-Branca.
Teste de Integração – Projeto e Arquitetura de Software	Desenvolvedores	Os componentes devem ser montados ou integrados para formar o pacote de software completo. Cuida dos tópicos associados com os problemas duais de verificação e construção de programas. As técnicas de projeto de casos de teste que enfocam as entradas e saídas são mais prevalentes durante a integração.
Validação – Sw construído x Análise de Requisitos (alto nível)	Analista de Teste	Fornecem garantia final de que o software satisfaz a todos os requisitos funcionais, comportamentais e de desempenho.
Sistema	Analista de Teste	Verifica se todos os elementos (hardware, pessoal, banco de dados) combinam adequadamente e se a função/desempenho global do sistema é alcançada.

### Estratégias de Teste de Software para Arquitetura de Software Orientados a Objetos (OO)

Esta estratégia é idêntica a aplicada a arquiteturas convencionais, mas diferente na abordagem.

À medida que as classes são integradas em uma arquitetura OO, uma série de testes de regressão é feita para descobrir erros devidos a comunicação e colaboração entre classes (componentes) e efeitos colaterais causados pela adição de novas classes.

Finalmente o sistema é testado como um todo para garantir que erros nos requisitos sejam descobertos.

### **Teste de Unidade**

O primeiro nível de teste é o Teste Unitário, que envolve assegurar que cada funcionalidade especificada no desenho do componente tenha sido implementada corretamente neste componente.

Características:

- \_ Testes realizados em uma unidade independente do produto;
- \_ Estágio mais baixo da escala de teste;
- \_ Aplicado nos menores componentes de código criados, visando garantir que estes atendem as especificações funcionais e de arquitetura;
- \_ Geralmente realizado pelo desenvolvedor que construiu a unidade;
- \_ Utiliza técnicas de Caixa Branca.

Ex.: funções, pedaços de códigos.

Este tipo de teste está limitado à função do componente.

### **Teste de Integração**

Neste nível os testes não são focados em "o quê" os componentes fazem, mas se eles se comunicam conforme especificado no desenho do sistema.

Características:

- \_ Acontece após os Testes Unitários;
- \_ Realizados dentro de um ambiente controlado;
- \_ Geralmente realizado pelo analista de sistema para um módulo ou conjunto de programas;
- \_ Normalmente são seguidos dois métodos de interação:
  - \_ Abordagem Top-Down
  - \_ Abordagem Bottom-Up

### **Teste de Regressão**

Tem como propósito garantir que os defeitos encontrados foram corrigidos e que as correções ou inserções de novos códigos em determinados locais do software não afetaram outras partes inalteradas do produto. Trata de re-testar o teste.

É necessário ter ferramentas para execução do teste de regressão, isto porque é inviável testar novamente todo o Software.

Reduz "efeitos colaterais". Devem ser executados toda vez que uma mudança importante é feita no Software (integração de novos componentes).

Teste de regressão deve focalizar função de módulo crítico.

### **Teste Fumaça**

Em suma é considerado uma estratégia de integração constante, onde software é reconstruído (com adição de novos componentes) e submetido a teste todos os dias.

Este tipo de teste deve exercitar o sistema de ponta a ponta, sem precisar ser exaustivo, mas deve ser capaz de expor os problemas principais, no entanto deve ser rigoroso.

### **Teste de Validação**

Este tipo de teste começa no fim do teste de Integração, aqui o software está completamente montado e os erros de interface já foram descobertos e corrigidos, focaliza ações visíveis ao usuário e saídas do sistema reconhecidas pelo usuário, não existindo distinção entre o software convencional e o orientado a objetos.

Validação - o software funciona de um modo que pode ser razoavelmente esperado pelo cliente. O teste de validação é feito mediante a Especificação dos Requisitos de Software – documento que descreve todos os atributos do software visíveis aos usuários, este documento contém uma seção chamada Critério de Validação, base para a abordagem do respectivo teste.

A validação do software é conseguida por intermédio de uma série de testes que demonstram conformidade com os requisitos.

### **Documentação**

Plano de Teste – delinea as classes de teste a ser conduzidas.

Procedimento de Teste – define os casos de teste específicos.

Ambos são projetados para garantir que todos os requisitos funcionais sejam satisfeitos, bem como as características comportamentais sejam conseguidas, todos os requisitos de desempenho sejam alcançados, a documentação esteja correta, usabilidade e outros requisitos sejam satisfeitos.

Após cada caso de teste tenha sido executado, uma de duas possíveis condições se realiza:

A característica de função ou desempenho satisfaz a especificação e é aceita ou é descoberto um desvio da especificação e uma lista de deficiência é criada, estes raramente podem ser corrigidos antes da entrega programada.

É impossível para o desenvolvedor de software prever como o cliente usará realmente um programa, o que parece claro para o testador pode ser inteligível para um usuário no campo.

Quando um software é feito sob encomenda, uma série de testes de aceitação é conduzido para permitir ao cliente validar todos os requisitos, este conduzido pelo usuário final, podendo acontecer ao longo de um período de semanas ou meses, descobrindo conseqüentemente erros cumulativos que poderiam degradar o sistema ao longo do tempo.

Se o software é desenvolvido como um produto a ser usado por vários clientes, não é prático realizar testes formais de aceitação com cada um. Para isto utilizam-se os seguintes testes: alfa e beta.

Testes Alfa – conduzido na instalação do desenvolvedor com os usuários finais. Sendo o software usado em um ambiente natural com o desenvolvedor acompanhando os usuários e registrando erros e problemas de uso, conduzido em um ambiente controlado.

Teste Beta – conduzido nas instalações dos usuários finais, sem a presença do desenvolvedor. O cliente registra todos os problemas reais ou imaginários e os relata ao desenvolvedor responsável em fazer as devidas correções e publicar uma nova versão.

Podemos tratar aqui três dimensões:

Funcionalidade – quais as regras de negócio devem contemplar o Software.

Performance – envolve tempo mínimo de resposta das funções.

Qualidade - envolve atributos de confiança do Software.



Neste contexto aproveitaremos para falar do teste de funcionalidade.

### **Teste de Funcionalidade**

O teste funcional tem por meta verificar se o software executa corretamente suas funções, se a implementação das regras de negócio foi apropriada e se o sistema é capaz de sustentar sua correta execução por um período contínuo de uso.

Baseado nas técnicas de caixa-preta. Analisa as saídas e resultados. Os testes funcionais são baseados nos procedimentos determinados nos casos de teste.

Tem como objetivos: assegurar a funcionalidade do sistema, incluindo entrada de dados, processamento e resposta, verificar se os requisitos dos usuários estão implementados e se atendem os usuários, verificar se o sistema funciona corretamente após um período contínuo de utilização.

Deve ser usado em qualquer sistema devendo ter suas funcionalidades testadas, pode ser usado desde a fase de especificação de requisitos até a fase de operação do sistema.

### **Teste de Sistema**

Os testes de sistema têm foco no funcionamento do sistema como um todo, para validar a exatidão e a perfeição na execução das funções requeridas.

Deve ser executado mediante a especificação do sistema para verificar se estão sendo entregues as funcionalidades que o cliente solicitou.

Características Deste Nível:

- Acontece após todos os testes de integração,
- Realizado dentro de um ambiente controlado (servidor, equipe, máquinas clientes, Sw, browser,
- Realizado pela equipe de testes,
- Envolve testes especializados para validar todos os requisitos funcionais e não-funcionais como: Desempenho, Volume, Documentação e Robustez.

### **Teste de Recuperação - Teste de Contingência**

A meta principal do teste de contingência é verificar se, após uma falha, a sua recuperação é adequadamente executada, garantindo a continuidade dos serviços.

Força ao sw falhar de diversas maneiras e verifica se sua recuperação é executada adequadamente.

Objetivos:

- Manter o backup dos dados;
- Armazenar o backup em local seguro;
- Documentar os procedimentos de recuperação;
- Deixar claro as responsabilidades das pessoas em caso de um desastre.

Quando Usar:

- Sempre que a continuidade dos serviços for essencial para o negócio;
- Perdas devem ser estimadas (quanto vou perder por hora?);
- A quantidade de perda potencial estipula a quantidade de esforço que irá ser empregada.

### **Teste de Segurança**

A principal meta do teste de segurança é garantir que os dados ou funções de um sistema possam ser acessados apenas por atores autorizados a acessá-las.

Todas as formas de ataque de acesso indevido devem ser simuladas.

Verifica se todos os mecanismos de proteção estão embutidos no sistema ou protegerão de fato de acessos indevidos.

O escopo de teste de segurança vai desde verificar se um usuário só tem acesso as funcionalidades que ele realmente deve ter até testar a segurança do sw contra ataques de hackers.

Objetivos:

- Garantir que os dados do sistema estão protegidos adequadamente;
- Assegurar que todos os riscos que envolvem os acessos indevidos foram identificados;
- Analisar as ameaças e vulnerabilidades, tanto físicas como lógicas;
- Assegurar que os mecanismos de controle contra acessos indevidos foram corretamente implementados.

Quando Usar:

- Testes básicos de segurança devem ser aplicados a todos os softwares;
- Quando a informação que circula através do software for de importância fundamental para a organização.

### **Teste de Estresse**

A principal meta do teste de estresse é entender o comportamento do sistema durante condições-limite de execução ou fora da tolerância esperada.

Tipicamente envolve a execução do sistema com baixos recursos de hardware e software, ou a concorrência por estes recursos (retorno do BD, uso da rede).

Objetivos:

- Determinar a que condições-limite de recursos o software é capaz de ser executado;
- Verificar se o sistema é capaz de garantir tempos adequados de resposta sendo executado em condições-limite;
- Verificar se há restrições quanto ao ambiente em que o software vai operar;
- Determinar que volumes de transação, normais e acima dos normais, podem ser processados num período de tempo esperado;

Quando Usar:

- Quando não se sabe quais as condições mínimas de recursos para operacionalização do sistema, sem que haja perdas significativas.

### **Teste de Desempenho - Teste de Performance**

O teste de performance, ou de desempenho como também é conhecido, mede e avalia o tempo de resposta, o número de transações e outros requisitos sensíveis ao tempo de resposta do sistema.

Objetivos:

- Determinar o tempo de resposta das transações;
- Verificar se os critérios de desempenho estabelecidos estão sendo atendidos;



## Reengenharia Organizacional

Reengenharia organizacional, como o próprio nome sugere, nada mais é do que um modo de reorganizar e reestruturar todos os processos internos de uma empresa com o objetivo principal de alavancar e impulsionar ainda mais resultados positivos e satisfatórios.

Trata-se de um método criado nos anos 90 pelo consultor e pensador Michael Hammer que tinha como intuito acabar com o comodismo e as rotinas administrativas que impediam as organizações de crescer e se desenvolver.

A **reengenharia organizacional** verifica e questiona todo o funcionamento de um negócio a fim de identificar falhas, oportunidades de mudanças e melhorias que gerem novas soluções e maior lucratividade.

Como aplicar a reengenharia organizacional em uma empresa

Confira o passo a passo de como aplicar o **conceito de reengenharia** organizacional no ambiente corporativo:

### - Fase de organização

O primeiro passo para implantar o conceito de reengenharia organizacional é descobrir e especificar, criteriosa e detalhadamente, todos os procedimentos e processos que a empresa executa diariamente e, principalmente, definir quais deles serão aperfeiçoados e o que será necessário para a conclusão da metodologia.

### - Fase do planejamento

Na segunda fase é necessário planejar quais artifícios serão necessários para a implementação da reengenharia organizacional, ou seja, listar tudo o que será utilizado durante o procedimento: prazos, orçamentos, distribuição de tarefas e definir os profissionais necessários para cada etapa do processo.

### - Fase de implantação

Nesta fase é hora de avaliar todas as atividades, rotinas, tarefas e recursos definidos durante a fase de avaliação, as principais deficiências, carências, falhas e oportunidades de melhorias e, em seguida, reorganizar e reestruturar todas essas rotinas de forma proveitosa para a empresa ou negócio.

### - Fase de mensuração de resultados

Na última fase é feita a verificação dos resultados atingidos com a aplicação da reengenharia organizacional e a mensuração dos efeitos de cada mudança, ou seja, avaliar se as alterações foram assertivas e quais benefícios ela gerou para a empresa, seja lucratividade, agilidade nos processos, melhoria de performance, entre outras coisas.

Benefícios da reengenharia organizacional

Confira quais são as principais vantagens que a aplicação da reengenharia na administração de uma empresa oferece:

Atualização de processos e recursos corporativos;

Redução de gastos;

Aumento dos resultados e de performance;

Melhora na execução de tarefas;

Modernização da gestão administrativa;

Resultados mensuráveis;

Oportunidade de crescimento e expansão empresarial;

Eliminação de erros, comportamentos e processos sabotadores.

### **Ênfase no cliente**

O título deste texto sugere priorização e privilégios ao cliente, porém, os empresários e colaboradores que pensam em clientes de forma restrita como essa, provavelmente terão dificuldades em se manterem no mercado competitivo e exigente da atualidade. A ênfase no cliente representa a necessidade de revelar a importância do mesmo para a empresa como ser humano inteligente, produtivo e com necessidades a serem satisfeitas.

Assim, o relacionamento entre cliente e empresa está inserido na cultura da sociedade onde ambos realizam as suas práticas sociais, onde ambos resolvem as suas vidas, por isso a relação deve ser do tipo ganha-ganha.

### **Desenvolvimento**

O conteúdo deste texto dá continuidade a uma discussão antiga e fértil na área da administração, porque desde os primeiros contatos das pessoas com as atividades de compra, produção, venda e distribuição, aprendem-se que “o cliente está sempre certo”; “o cliente tem razão”; “quem manda é o cliente” e tantas outras frases que representam o entendimento de que o cliente é o “rei”. No entanto, muitas dessas afirmações permanecem na teoria ou são mal interpretadas quando transformadas em práticas.

Dessa forma, a importância deste texto para uma discussão mais amigável sobre as características das relações entre clientes e empresas, não como se o mercado fosse um campo de batalha para ambos, mas como a humanização das ações de ambos pode resultar em níveis mais elevados de satisfação mútua.

### **Reconhecer os clientes como pessoas**

Os clientes são pessoas. Cada cliente é um ser que se assemelha aos outros porque é diferente; então, é de fundamental importância o constante interesse em aprender quem e como é o cliente.

Constatamos que a “era” do cliente ou o foco no cliente não é um modismo ou “nuvens passageiras”, mas uma realidade que coloca o mercado em um processo de constantes mudanças e que as empresas, para obter as vantagens competitivas tão almejadas, devem criar todos os dias diferenciais ou benefícios que busquem satisfazer as expectativas de todos os seus clientes, sejam eles pouco ou muito exigentes. Também os concorrentes, com seus diferenciais, são aqueles que contribuem para a ampliação do nível de exigência dos clientes em seu momento de reinado.

É preciso não somente satisfazer as necessidades e desejos dos consumidores, mas também antecipá-las; dentro de um conceito amplo de mercado, podemos dizer que os estímulos de Marketing como produto, preço, promoção e ponto de venda, somados aos estímulos culturais, econômicos, sociais, geográficos, políticos e religiosos, estes considerados estímulos ambientais, interferem sobre os comportamentos de compra dos consumidores.

Dentre as características do consumidor, a sua motivação, percepção, atitude, estilo de vida e personalidade sofrem interferências dos estímulos de marketing e ambientais. O cenário é capaz de revelar o quanto é importante para as organizações, o conhecimento sobre seus clientes atuais e potenciais. Não é apenas através de uma pesquisa de mercado que é possível buscar os conhecimentos referentes aos consumidores, mas também ouvindo, observando os hábitos, costumes, gostos, preferências e reações diante de diferentes tipos de atendimento.

Alguns autores defendem a idéia de que importantes estratégias de Marketing não requerem grandes investimentos, pois presumem que clientes ou consumidores gostam da cortesia dos agentes vendedores, além de simpatia, alegria e amizade dos mesmos. Tal idéia tem fundamento quando se considera somente o dinheiro como investimento, mas ao ampliar a visão do que seja investimento descobre-se que a criação e o desenvolvimento de cultura de excelência no relacionamento com os clientes implica em investimento sim; pois, a administração de nível estratégico da empresa cria programas de longo prazo ao cuidar do desenvolvimento organizacional, nos quais a política de relacionamento interna e externa da empresa envolve a educação, dinâmicas de grupos, humanização, valorização dos colaboradores; tudo isso representa investimento em prol do marketing de relacionamento e do negócio.

As empresas encontram o fracasso, normalmente, como resultado do somatório de pequenos equívocos; raramente uma empresa fracassa por um único grande erro. Treinar e valorizar os recepcionistas, os agentes de segurança, as telefonistas, as secretárias, os auxiliares e todos aqueles que se relacionam diretamente com os clientes ou consumidores, porque esses profissionais podem, em um único contato, melhorar ou prejudicar toda a imagem de uma empresa; essa cultura é uma das chaves para atrair o cliente - o acompanhamento de todo o processo de compra, tendo o cliente como prioritário. Sendo essa cultura uma das maiores vantagens competitivas, então nenhuma venda deve ser a última; ouvir os clientes supera a mais cara pesquisa de mercado e é muito menos oneroso manter um cliente atual do que conquistar um novo entre os clientes potenciais.

Considerar as pessoas, sejam elas colaboradoras ou clientes, como o maior patrimônio da organização. Lembrar do cliente serve para melhorar o contato. As empresas devem conquistar e manter colaboradores que gostam de trabalhar com os clientes, pois reconhecem cada um deles como rei. O atendimento ao cliente deve ser sinônimo de empatia e atenção; atender ao cliente traz rentabilidade para as organizações, o cliente deve ser a melhor referência das empresas. Cliente satisfeito faz propaganda da empresa fornecedora; clientes satisfeitos geram referências de valor incalculável e contribuem não somente para a elevação do nível de vendas, mas, principalmente, como marketing do negócio, cujo valor é indiscutível.

Deve-se observar, no entanto, a existência de maus clientes ou maus consumidores; por isso, as regras de contato e relacionamento devem ser claras para que seja norteadada a ênfase aos bons clientes; saber identificar corretamente os clientes ruins, a fim de conquistá-los e criar controles desestimuladores aos hábitos e práticas dos maus clientes, no sentido de impedir opções que venham distorcer os procedimentos éticos e de conduta dentro e fora das organizações. Contudo, convém não desenvolver práticas de atendimento ao cliente baseadas no fingimento ou representação; o cliente bom ou ruim deve perceber claramente o tipo de cultura pelo qual a empresa se pauta.

Assim, a valorização dos bons clientes deve ser contínua e os maus clientes devem tomar conhecimento dessa valorização, porque somente assim possivelmente perceberão o interesse e a prática da empresa voltados para a construção de uma política que estabelece critérios em busca de uma relação boa entre empresa, fornecedores e clientes ou consumidores.

### **Ouvir para conversar melhor**

Saber conversar faz bem; na maioria das vezes essa aprendizagem produz excelentes resultados. É

conversando e ouvindo as pessoas que se podem conhecer histórias interessantes sobre os seus pensamentos em relação aos locais nos quais realizam suas compras. Restaurantes, lojas de departamentos e supermercados podem exercer um verdadeiro fascínio nos clientes, ou, ao contrário, podem causar desinteresse e repúdio. Esta dicotomia encontra-se entre os assuntos mais discutidos quando se trata da relação entre clientes e organizações de compra, venda e prestação de serviços.

É natural que as pessoas conversem, troquem idéias e questionem o local onde deixam grande parte de seus salários. Desse modo, durante os contatos com os clientes, ficou claro que estes não avaliam de forma geral o atendimento de um ponto de venda e sim de forma restrita e departamentalizada; daí, não são raros os seguintes comentários: “o açougue daquele novo hipermercado é ótimo”; “eu compro neste supermercado porque a qualidade e as opções da padaria são excelentes”; “o atendimento das moças do caixa é o melhor. Elas falam e respondem o que a gente pergunta...!”

A melhor estratégia é humanizar o atendimento ao cliente; procurar satisfazer seus anseios de satisfação de necessidades de forma clara, objetiva e simples. Desenvolver um trabalho com qualidade continuada e atenção especial, transmitindo confiança para que as pessoas percebam que na empresa tem alguém pensando no bem-estar dos clientes. Detalhe por detalhe; há palavras mágicas que fazem os clientes voltarem com maior frequência: bom dia!, boa tarde!, boa noite!; oi, tudo bem? posso ajudar? por favor! valeu! foi um grande prazer! volte sempre! muito obrigado!

Sem querer ser utópico, um ponto de venda pressupõe solução de problemas; é uma atividade basicamente de prestação de serviços desenvolvida por pessoas, por isso é preciso haver prazer no que se faz; vibração e muita paixão. Adotar como diferenciais unicamente os preços e as ofertas promocionais é facilitar a cópia por parte do concorrente da próxima esquina.

Logo, o diferencial deve estar centrado em talentos humanos, colaboradores que gostam do que fazem e utilizam suas inteligências na busca da excelência do negócio no qual trabalham. É assim que se conquistam mercados, simpatia dos clientes e garantia de vantagens competitivas para uma empresa por período de tempo maior. A valorização teórica e prática dos clientes é uma cultura que, estrategicamente construída, pode mudar a história de uma empresa.

### **O Marketing de Relacionamento**

De tempos em tempos adota-se um novo conceito em administração, muitas vezes trata-se de modismo e logo cai no esquecimento. Porém, uma nova estratégia administrativa promete figurar na pauta dos empresários brasileiros, com a expectativa de revolucionar as relações de mercado nos próximos anos: o Marketing de Relacionamento – considerado a arte de criar diferenciais para estabelecer uma sintonia fina e duradoura com os clientes.

Especialistas apostam que o marketing de relacionamento será mais uma ferramenta importante no reinado do cliente, buscando sempre a criação de diferenciais e a aproximação entre empresa e clientes.

Recompensas geralmente se constituem como ferramenta poderosa para estimular a recompra, mas é importante salientar que deve ser mantido um elo racional e emocional de equilíbrio na relação com os clientes; surpreender positivamente, oferecer benefícios intangíveis como recompensas, distribuir informações de interesse dos clientes, especialmente quando forem raras, monitorar o desejo dos clientes, perguntar o que estão achando dos serviços, não esquecendo de facilitar as vias de respostas (telefone, fax, carta resposta, e-mail etc.), e, finalmente, averiguar diuturnamente como estão funcionando os processos no interior da empresa e no ambiente onde a mesma está inserida, para a identificação e reconhecimento de eventuais falhas, equívocos ou erros e providenciar as soluções capazes de construir e reconstruir um relacionamento satisfatório tanto para os clientes quanto para a empresa.

Ao telefonar para uma empresa, o cliente não tem satisfação em ouvir a seguinte frase: “a sua ligação é muito importante para nós. Aguarde um instante, por favor. Logo iremos atendê-lo”. O tempo passa e nada. Enquanto isso, músicas nem sempre agradáveis tocam ao telefone; configura-se, neste caso, imprudência que pode ser classificada de relacionamento às avessas.

Portanto, é necessário refletir se a empresa desenvolve e constrói a cultura de que o cliente é o rei; e se isso é uma verdade, então todos os integrantes da equipe de trabalho da empresa precisam estar comprometidos com a busca da satisfação dos clientes. É fato que um cliente ou consumidor com saldo devedor, ou seja, obrigações maiores que direitos, pode sempre criar objeções aos procedimentos da empresa e tende a criticar determinadas ações, para tanto deve-se formular ações que não produzam motivos de críticas e “chacotas” originárias desses clientes.

O comprometimento baseado na seriedade, ética e profissionalismo pode contribuir para esvaziar críticas baseadas no caráter emocional e sem nenhuma fundamentação técnica. Por outro lado, o cliente ou consumidor que cumpre suas obrigações deve ser valorizado através de ações eficazes que elevem o nível de satisfação dos mesmos.

Ou seja, deve-se diferenciar sem discriminar; premiar as virtudes e as ações consideradas corretas dentro do ambiente do mercado. O respeito, a seriedade, a dedicação, o bom atendimento com base profissional, a ética e a coerência da empresa contribuirão certamente para a melhoria da satisfação dos clientes; tudo isso é possível quando as pessoas que trabalham na administração, na operação, nas vendas e nas compras integram equipes colaborativas, cujo objetivo comum é a continuidade da empresa no mercado.

### **Considerações Finais**

A valorização do cliente não é resultado de estratégias cujas fórmulas estão prontas e acompanhadas de um manual de uso. Trata-se de um conjunto de ações de pessoas, colaboradores e clientes, construídas durante as relações interpessoais, baseadas em valores humanos e universais, com a finalidade de comprar e vender bens e serviços para a satisfação de necessidades diversas da sociedade.

A compreensão do significado da frase – o cliente é o rei – é a cultura de atitudes que privilegiam a ética, a racionalidade, o comprometimento, a justiça e o interesse na manutenção de uma relação ganha-ganha entre cliente e empresa, na qual ambos estejam satisfeitos.

---

---

---

---

---

---

---

---

---

---



## Engenharia de Requisitos

Requisitos são as bases para todo projeto, definindo o que as partes interessadas de um novo sistema necessitam e também o que o sistema deve fazer para satisfazer as suas necessidades. Os requisitos guiam as atividades do projeto e normalmente são expressos em linguagem natural para que todos possam obter o entendimento.

Além dos requisitos definirem os problemas e soluções também devemos definir os riscos e prover soluções satisfatórias casos esses riscos venham a falhar. Dessa forma, os requisitos definem as bases para:

- Planejamento do Projeto
- Gerenciamento de Riscos
- Testes de Aceitação
- Controle de Mudanças

Os requisitos são tão importantes que normalmente eles possuem um grande impacto nas falhas dos projetos de software. Abaixo destacamos os três principais problemas para falhas em projetos de software:

- Requisitos: requisitos pobremente organizados, muito mal expressados, pobremente relatados para as partes interessadas, mudanças muito rápidas ou desnecessárias, expectativas não realistas.
- Gerenciamento de Problemas de Recursos: incapacidade de ter dinheiro suficiente e falta de apoio ou fracasso para impor disciplina e planejamento. Muitas delas surgem pela falta de controle de requisitos.
- Políticas: Contribui para o primeiro e segundo problema.

O mais interessante é que todos esses problemas podem ser resolvidos com pouco dinheiro.

Dessa forma, podemos afirmar que a engenharia de requisitos é um processo que engloba todas as atividades que contribuem para a produção de um documento de requisitos e sua manutenção ao longo do tempo.

Ao longo do artigo veremos mais o que é engenharia de requisitos, como se dá o processo de engenharia de requisitos e quais são as suas principais atividades.

### Processo de Engenharia de Requisitos

Portanto, a engenharia de requisitos é o processo pelo qual os requisitos de um produto de software são coletados, analisados, documentados e gerenciados ao longo de todo o ciclo de vida do software.

Entendido o que é a engenharia de requisitos, podemos partir para conhecer como funciona o processo de engenharia de requisitos. Fazendo um paralelo com o processo de desenvolvimento de software, tem-se que um processo de software envolve diversas atividades que podem ser classificadas em: Atividades de Desenvolvimento onde temos atividades que contribuem para o desenvolvimento do produto de software como levantamento e análise de requisitos, projeto e implementação; Atividades de Gerência que envolvem atividades de planejamento e acompanhamento gerencial do projeto; e Atividades de Controle da Qualidade que estão relacionadas com a avaliação da qualidade do produto.

De uma forma geral os requisitos possuem um papel fundamental para o desenvolvimento de software. Os requisitos de software são uma das principais medidas de sucesso de um software, visto

que se eles atendem aos objetivos e requisitos para os quais foi construído o software e está totalmente de acordo com as necessidades dos clientes. Requisitos são a base para estimativas, modelagem, projeto, implementação, testes e até mesmo para a manutenção. Assim, os requisitos estão presentes ao longo de todo o ciclo de vida de um software.

Já no início de um projeto temos que levantar os requisitos, entendê-los e documentá-los. Como os requisitos são extremamente importantes para o sucesso de um projeto, devemos também realizar atividades de controle da qualidade para verificar, validar e garantir a qualidade dos requisitos. Outra medida fundamental é gerenciarmos a evolução dos requisitos, visto que os negócios são dinâmicos e não temos como garantir que esses requisitos não sofrerão alterações. Dessa forma devemos manter a rastreabilidade entre os requisitos e os demais artefatos produzidos no projeto.

Portanto, podemos constatar que os requisitos envolvem atividades de desenvolvimento através do Levantamento e Análise e Documentação de Requisitos, gerência através da Gerência de Requisitos e por fim o controle da qualidade através da Verificação, Validação e Garantia da Qualidade de Requisitos. Todas essas atividades que são relacionadas a requisitos é o Processo de Engenharia de Requisitos.

No restante do artigo veremos um pouco mais o que são cada uma das dessas atividades que fazem parte do processo de engenharia de requisitos.

#### Levantamento de Requisitos

Esta é a fase inicial do processo de engenharia de requisitos. Nessa atividade levam-se em conta as necessidades dos usuários e clientes, informações de domínio, sistemas já existentes na organização, regulamentos vigentes, leis, etc.

O objetivo nessa fase é entendermos a organização como um todo, seus processos, necessidades, possibilidades de melhorias e restrições existente. Dessa forma, nos preocupamos na descoberta dos requisitos.

Essa fase é bastante complexa e também exige obtermos informações dos interessados, consultar documentos, obter conhecimentos do domínio e estudar o negócio da organização.

No levantamento de requisitos devemos atentar para quatro entendimentos que devemos possuir: Entendimento do Domínio da Aplicação onde se entende, de uma maneira geral, a área na qual o sistema será aplicado; Entendimento do Problema onde entendemos os detalhes do problema específico a ser resolvido com o auxílio do sistema a ser desenvolvido; Entendimento do Negócio onde entendemos como o sistema afetará a organização e como contribuirá para que os objetivos do negócio e os objetivos gerais da organização sejam atingidos; e por fim o Entendimento das Necessidades e das Restrições dos Interessados onde entende-se as demandas de apoio para a realização do trabalho de cada um dos interessados no sistema, entende-se os processos de trabalho a serem apoiados pelo sistema e o papel de eventuais sistemas existentes na execução e condução dos processos de trabalho.

Para o levantamento de requisitos temos diversas técnicas úteis que podem ser empregadas para ajudar o levantamento desses requisitos, são elas: entrevistas, questionários, observação do ambiente e dos indivíduos nas suas tarefas cotidianas na organização, análise de documentos existentes na organização, cenário de interação entre o usuário final e o sistema onde o usuário pode simular a sua interação com o sistema explicando para o analista o que ele está fazendo e de que informações ele precisa para realizar a tarefa, prototipagem onde uma versão preliminar do sistema, muitas vezes não operacional e descartável, é apresentada ao usuário para capturar informações específicas sobre seus requisitos de informação, observação reações, dinâmica de grupo, e diversas outras técnicas que também podem ser empregadas.

## Análise de Requisitos

Após a atividade de Levantamento de Requisitos inicia-se a atividade de Análise de Requisitos, que é onde os requisitos levantados são usados como base para a modelagem do sistema.

Os requisitos são escritos tipicamente em linguagem natural, no entanto, é útil expressarmos requisitos mais detalhados do sistema de maneira mais técnica através de diversos tipos de modelos que podem ser utilizados. Esses modelos são representações gráficas que descrevem processos de negócio, o problema a ser resolvido e o sistema a ser desenvolvido. Representações gráficas são muito mais compreensíveis do que descrições detalhadas em linguagem natural e por isso são utilizados.

Dessa forma, a análise é uma atividade de modelagem. Vale ressaltar que essa modelagem é conceitual, pois estamos preocupados com o domínio do problema e não com soluções técnicas. Portanto, os modelos de análise são elaborados a fim de obtermos uma compreensão maior do sistema a ser desenvolvido e para especificá-lo.

Na análise de requisitos buscam-se principalmente duas perspectivas, a primeira delas é a estrutural onde se busca modelar os conceitos, propriedades e relações do domínio que são consideradas relevantes para o sistema em desenvolvimento. A segunda perspectiva é a comportamental onde se busca modelar o comportamento geral do sistema, de uma de suas funcionalidades ou de uma entidade.

Os diagramas da UML proveem suporte a todos os diagramas necessários nessa fase de análise.

## Documentação de Requisitos

Os requisitos e modelos capturados nas etapas de Levantamento de Requisitos e Análise de Requisitos devem ser descritos e apresentados em documentos. A documentação é uma atividade de registro e oficialização dos resultados da engenharia de requisitos. Como resultado, um ou mais documentos devem ser produzidos.

Essa documentação escrita de uma boa forma apresenta diversos benefícios como facilidade na comunicação dos requisitos, redução no esforço de desenvolvimento, fornece uma base realista para estimativas, boa base para verificação e validação, entre outros benefícios.

A documentação produzida também possui diversos interessados que usam a documentação para diferentes propósitos. Os clientes, Usuários e Especialistas de Domínio atuam na especificação, avaliação e alteração dos requisitos. Gerentes de Cliente utilizam a documentação para planejar uma proposta para o sistema e para planejar e acompanhar o processo de desenvolvimento. Os desenvolvedores utilizam a documentação para compreender o sistema e a relação entre as suas partes. Os Testadores utilizam a documentação para projetar casos de teste.

O Documento de Requisitos deve conter uma descrição do propósito do sistema, uma breve descrição do domínio do problema e listas de requisitos funcionais, não funcionais e regras de negócio, tudo descrito em linguagem natural. Desenvolvedores, clientes, usuários e gerentes utilizam esse documento. Outro documento que pode ser produzido é o Documento de Especificação de Requisitos que deve conter os requisitos escritos a partir da perspectiva do desenvolvedor, contendo inclusive uma correspondência direta com os requisitos no Documento de Requisitos. Os modelos produzidos na fase anterior devem ficar dentro deste documento de especificação de requisitos.

## Verificação, Validação e Garantia da Qualidade de Requisitos

Essa fase deve ser iniciada o quanto antes no processo de desenvolvimento de software. Os requisitos são a base para o desenvolvimento, dessa forma é fundamental que eles sejam cuidadosamente avaliados. Portanto, os documentos produzidos durante a fase anterior devem ser submetidos à verificação e validação de requisitos.

A diferença entre verificação e validação é que verificação assegura que o software esteja sendo construído de forma correta. Por sua vez a validação assegura que o software que está sendo desenvolvido é o software correto. Portanto, a verificação assegura que os artefatos produzidos atendem aos requisitos e a validação assegura que os requisitos e o software que foi derivado desses requisitos atendem ao uso proposto.

A validação envolve a participação do usuário e do cliente, pois apenas eles possuem condições de confirmar que os requisitos atendem aos propósitos do sistema.

Nessa fase examinam-se os documentos de requisitos para garantir que todos os requisitos tenham sido declarados de modo não ambíguo, que as inconsistências, conflitos, omissões e erros tenham sido detectados e corrigidos, que os documentos estão em conformidade com os padrões estabelecidos, que os requisitos realmente satisfazem às necessidades dos clientes e usuários.

Portanto, os requisitos devem ser completos, corretos, consistentes, realistas, necessário, passível de ser priorizado, verificável e rastreável.

### Gerência de Requisitos

Mudanças nos requisitos ocorrem durante todo o processo de software, desde o levantamento de requisitos até durante a operação do sistema em produção. Isso ocorre devido à descoberta de erros, omissões, conflitos, inconsistência nos requisitos, melhor entendimento dos usuários sobre as suas necessidades, problemas técnicos, mudanças de prioridades do cliente, mudanças no negócio, concorrentes, mudanças econômicas, mudanças no ambiente de software, mudanças organizacionais, etc.

Para minimizar os problemas causados por essas mudanças é necessário gerenciar requisitos. O Processo de Gerencia de Requisitos envolve atividades que ajudam a equipe a identificar, controlar e rastrear requisitos e gerenciar mudanças de requisitos em qualquer momento ao longo do ciclo de vida do software.

Portanto, os objetivos do processo são gerenciar alterações nos requisitos acordados, gerenciar relacionamentos entre requisitos, gerenciar dependências entre requisitos e outros documentos produzidos durante o processo de software. Dessa forma, a gerência de requisitos possui as seguintes atividades: controle de mudanças, controle de versão, acompanhamento do estado dos requisitos e rastreamento de requisitos.

A definição de um processo apropriado para uma organização é muito importante e traz diversos benefícios, pois uma boa descrição de um processo fornece orientações e reduz a probabilidade de erros ou esquecimentos. O mais importante é saber que não existe um processo ideal, portanto adaptar um processo para as necessidades internas é sempre a melhor escolha ao invés de impor um processo à organização.

---

---

---

---

---

---

---

---

---

---

## Prototipação

A prototipagem de software é a atividade de criar protótipos de aplicativos de software, ou seja, versões incompletas do programa de software que está sendo desenvolvido. É uma atividade que pode ocorrer no desenvolvimento de software e é comparável à prototipagem conhecida de outras áreas, como engenharia mecânica ou manufatura.

Um protótipo normalmente simula apenas alguns aspectos e pode ser completamente diferente do produto final.

A prototipagem tem vários benefícios: o designer e o implementador de software podem obter feedback valioso dos usuários no início do projeto. O cliente e o contratante podem comparar se o software feito corresponde à especificação do software, de acordo com a qual o programa de software é construído.

Ele também permite que o engenheiro de software tenha algumas informações sobre a precisão das estimativas iniciais do projeto e se os prazos e marcos propostos podem ser cumpridos com sucesso. O grau de completude e as técnicas utilizadas na prototipagem estão em desenvolvimento e debate desde sua proposta no início da década de 1970.

O propósito de um protótipo é permitir que os usuários do software avaliem as propostas dos desenvolvedores para o projeto do produto final, experimentando-as de fato, em vez de ter que interpretar e avaliar o projeto com base em descrições. A prototipagem de software fornece uma compreensão das funções do software e possíveis ameaças ou problemas.

A prototipagem também pode ser utilizada por usuários finais para descrever e comprovar requisitos que não foram considerados, e que podem ser um fator chave na relação comercial entre desenvolvedores e seus clientes. O design de interação, em particular, faz uso intenso de prototipagem com esse objetivo.

Esse processo contrasta com o ciclo de desenvolvimento monolítico dos anos 1960 e 1970 de construir o programa inteiro primeiro e depois resolver quaisquer inconsistências entre o projeto e a implementação, o que levou a custos de software mais altos e estimativas ruins de tempo e custo.

A abordagem monolítica foi apelidada de técnica "Slaying the (software) Dragon", uma vez que assume que o designer e desenvolvedor de software é um único herói que tem que matar o dragão inteiro sozinho. A prototipagem também pode evitar a grande despesa e a dificuldade de ter que alterar um produto de software acabado.

A prática da prototipagem é um dos pontos que Frederick P. Brooks faz em seu livro de 1975 *The Mythical Man-Month* e seu artigo de aniversário de 10 anos "No Silver Bullet".

Um dos primeiros exemplos de prototipagem de software em larga escala foi a implementação do tradutor Ada/ED da NYU para a linguagem de programação Ada. Foi implementado em SETL com a intenção de produzir um modelo semântico executável para a linguagem Ada, enfatizando a clareza do design e da interface do usuário sobre a velocidade e eficiência. O sistema NYU Ada/ED foi a primeira implementação Ada validada, certificada em 11 de abril de 1983.

## Esboço

O processo de prototipagem envolve as seguintes etapas:

Identifique os requisitos básicos

Determine os requisitos básicos, incluindo as informações de entrada e saída desejadas. Detalhes, como segurança, geralmente podem ser ignorados.

Desenvolver protótipo inicial

O protótipo inicial é desenvolvido que inclui apenas interfaces de usuário. (Veja Protótipo Horizontal , abaixo)

### **Análise**

Os clientes, incluindo os usuários finais, examinam o protótipo e fornecem feedback sobre possíveis adições ou alterações.

### **Revisar E Aprimorar O Protótipo**

Usando o feedback, tanto as especificações quanto o protótipo podem ser melhorados. A negociação sobre o que está dentro do escopo do contrato/produto pode ser necessária. Se forem introduzidas alterações, pode ser necessário repetir os passos #3 e #4.

### **Protótipo Horizontal**

Um termo comum para um protótipo de interface do usuário é o protótipo horizontal. Ele fornece uma visão ampla de um sistema ou subsistema inteiro, concentrando-se na interação do usuário mais do que na funcionalidade do sistema de baixo nível, como acesso ao banco de dados. Os protótipos horizontais são úteis para:

Confirmação dos requisitos da interface do usuário e escopo do sistema,

Versão de demonstração do sistema para obter buy-in do negócio,

Desenvolva estimativas preliminares de tempo, custo e esforço de desenvolvimento.

Protótipo vertical

Um protótipo vertical é uma elaboração completa aprimorada de um único subsistema ou função. É útil para obter requisitos detalhados para uma determinada função, com os seguintes benefícios:

Design de banco de dados de refinamento ,

Obtenha informações sobre volumes de dados e necessidades de interface do sistema, para dimensionamento de rede e engenharia de desempenho,

Esclareça requisitos complexos analisando a funcionalidade real do sistema.

Tipos

A prototipagem de software tem muitas variantes. No entanto, todos os métodos são de alguma forma baseados em duas formas principais de prototipagem: prototipagem descartável e prototipagem evolutiva.

### **Prototipagem Descartável**

Também chamado de prototipagem fechada. A prototipagem descartável ou rápida refere-se à criação de um modelo que eventualmente será descartado em vez de se tornar parte do software final entregue. Depois que a coleta preliminar de requisitos é realizada, um modelo de trabalho simples do sistema é construído para mostrar visualmente aos usuários como seus requisitos podem parecer quando implementados em um sistema finalizado. É também uma forma de prototipagem rápida.

A prototipagem rápida envolve a criação de um modelo de trabalho de várias partes do sistema em um estágio muito inicial, após uma investigação relativamente curta.

O método utilizado na sua construção costuma ser bastante informal, sendo o fator mais importante a rapidez com que o modelo é fornecido. O modelo torna-se então o ponto de partida a partir do qual os usuários podem reexaminar suas expectativas e esclarecer seus requisitos. Quando esse objetivo é alcançado, o modelo do protótipo é 'jogado fora' e o sistema é formalmente desenvolvido com base nos requisitos identificados.

A razão mais óbvia para usar prototipagem descartável é que ela pode ser feita rapidamente. Se os usuários puderem obter feedback rápido sobre seus requisitos, eles poderão refiná-los no início do desenvolvimento do software. Fazer alterações no início do ciclo de vida do desenvolvimento é extremamente econômico, pois não há nada a ser refeito nesse ponto.

Se um projeto for alterado após uma quantidade considerável de trabalho ter sido feita, pequenas alterações podem exigir grandes esforços para serem implementadas, pois os sistemas de software têm muitas dependências. A velocidade é crucial na implementação de um protótipo descartável, pois com um orçamento limitado de tempo e dinheiro, pouco pode ser gasto em um protótipo que será descartado.

Outro ponto forte da prototipagem descartável é sua capacidade de construir interfaces que os usuários podem testar. A interface do usuário é o que o usuário vê como o sistema e, ao vê-la na frente dele, é muito mais fácil entender como o sistema funcionará.

Afirma-se que a prototipagem rápida revolucionária é uma maneira mais eficaz de lidar com problemas relacionados aos requisitos do usuário e, portanto, um aprimoramento maior da produtividade geral do software.

Os requisitos podem ser identificados, simulados e testados de forma muito mais rápida e barata quando questões de capacidade de evolução, manutenibilidade e estrutura de software são ignoradas. Isso, por sua vez, leva à especificação precisa dos requisitos e à subsequente construção de um sistema válido e utilizável do ponto de vista do usuário, por meio de modelos convencionais de desenvolvimento de software.

Os protótipos podem ser classificados de acordo com a fidelidade com que se assemelham ao produto real em termos de aparência, interação e tempo. Um método de criar um protótipo descartável de baixa fidelidade é a prototipagem em papel. O protótipo é implementado usando papel e lápis e, portanto, imita a função do produto real, mas não se parece com isso.

Outro método para construir facilmente protótipos descartáveis de alta fidelidade é usar um GUI Builder e criar um click dummy, um protótipo que se parece com o sistema de metas, mas não fornece nenhuma funcionalidade.

O uso de storyboards, animatics ou desenhos não é exatamente o mesmo que prototipagem descartável, mas certamente se enquadra na mesma família. Estas são implementações não funcionais, mas mostram como o sistema ficará.

Resumo: Nesta abordagem o protótipo é construído com a ideia de que será descartado e o sistema final será construído do zero. Os passos desta abordagem são:

Escreva os requisitos preliminares

Desenhe o protótipo

O usuário experimenta/usa o protótipo, especifica novos requisitos

Repita se necessário

Escreva os requisitos finais

### **Prototipagem Evolutiva**

A prototipagem evolutiva (também conhecida como prototipagem de prototipagem) é bem diferente da prototipagem descartável. O principal objetivo ao usar a prototipagem evolutiva é construir um protótipo muito robusto de forma estruturada e refiná-lo constantemente. A razão para esta abordagem é que o protótipo evolucionário, quando construído, forma o coração do novo sistema, e as melhorias e outros requisitos serão então construídos.

Ao desenvolver um sistema usando prototipagem evolucionária, o sistema é continuamente refinado e reconstruído.

"... a prototipagem evolutiva reconhece que não entendemos todos os requisitos e constrói apenas aqueles que são bem compreendidos."

Essa técnica permite que a equipe de desenvolvimento adicione recursos ou faça alterações que não puderam ser concebidas durante a fase de requisitos e design.

Para que um sistema seja útil, ele deve evoluir por meio do uso em seu ambiente operacional pretendido. Um produto nunca está "pronto"; está sempre amadurecendo à medida que o ambiente de uso muda... muitas vezes tentamos definir um sistema usando nosso quadro de referência mais familiar - onde estamos agora.

Fazemos suposições sobre a forma como os negócios serão conduzidos e a base tecnológica na qual os negócios serão implementados. Um plano é elaborado para desenvolver a capacidade e, mais cedo ou mais tarde, algo semelhante ao sistema imaginado é entregue.

Os protótipos evolutivos têm uma vantagem sobre os protótipos descartáveis, pois são sistemas funcionais. Embora eles possam não ter todos os recursos planejados pelos usuários, eles podem ser usados temporariamente até que o sistema final seja entregue.

"Não é incomum dentro de um ambiente de prototipagem para o usuário colocar um protótipo inicial para uso prático enquanto espera por uma versão mais desenvolvida... O usuário pode decidir que um sistema 'defeituoso' é melhor do que nenhum sistema."

Na prototipagem evolucionária, os desenvolvedores podem se concentrar em desenvolver partes do sistema que eles entendem, em vez de trabalhar no desenvolvimento de um sistema inteiro.

Para minimizar o risco, o desenvolvedor não implementa recursos mal compreendidos. O sistema parcial é enviado para as instalações do cliente. À medida que os usuários trabalham com o sistema, eles detectam oportunidades para novos recursos e solicitam esses recursos aos desenvolvedores.

Os desenvolvedores então aceitam essas solicitações de aprimoramento junto com as suas próprias e usam práticas sólidas de gerenciamento de configuração para alterar a especificação de requisitos de software, atualizar o design, recodificar e testar novamente.

### **Prototipagem Incremental**

O produto final é construído como protótipos separados. No final, os protótipos separados são mesclados em um design geral. Com a ajuda da prototipagem incremental, o intervalo de tempo entre o usuário e o desenvolvedor de software é reduzido.

### **Prototipagem Extrema**

A prototipagem extrema como processo de desenvolvimento é usada especialmente para o desenvolvimento de aplicativos da web.



Basicamente, ele divide o desenvolvimento web em três fases, cada uma baseada na anterior. A primeira fase é um protótipo estático que consiste principalmente em páginas HTML. Na segunda fase, as telas são programadas e totalmente funcionais usando uma camada de serviços simulada. Na terceira fase, os serviços são implementados.

"O processo é chamado de Extreme Prototyping para chamar a atenção para a segunda fase do processo, em que uma interface de usuário totalmente funcional é desenvolvida com muita pouca consideração pelos serviços além do contrato."

### **Vantagens**

Há muitas vantagens em usar a prototipagem no desenvolvimento de software – algumas tangíveis, outras abstratas.

**Tempo e custos reduzidos:** A prototipagem pode melhorar a qualidade dos requisitos e especificações fornecidos aos desenvolvedores. Como as mudanças custam exponencialmente mais para serem implementadas à medida que são detectadas posteriormente no desenvolvimento, a determinação antecipada do que o usuário realmente deseja pode resultar em um software mais rápido e mais barato.

**Envolvimento do usuário aprimorado e aumentado:** A prototipagem requer o envolvimento do usuário e permite que eles vejam e interajam com um protótipo, permitindo que forneçam feedback e especificações melhores e mais completos. A presença do protótipo sendo examinado pelo usuário evita muitos mal-entendidos e falhas de comunicação que ocorrem quando um lado acredita que o outro entendeu o que foi dito. Como os usuários conhecem o domínio do problema melhor do que qualquer pessoa da equipe de desenvolvimento, o aumento da interação pode resultar em um produto final com maior qualidade tangível e intangível. É mais provável que o produto final satisfaça o desejo do usuário por aparência, toque e desempenho.

### **Desvantagens**

Usar, ou talvez usar mal, a prototipagem também pode ter desvantagens.

**Análise insuficiente:** O foco em um protótipo limitado pode distrair os desenvolvedores de analisar adequadamente o projeto completo. Isso pode levar ao esquecimento de soluções melhores, à preparação de especificações incompletas ou à conversão de protótipos limitados em projetos finais mal projetados e difíceis de manter .

Além disso, como um protótipo é limitado em funcionalidade, ele pode não ser bem dimensionado se o protótipo for usado como base de uma entrega final, o que pode não ser percebido se os desenvolvedores estiverem muito focados em construir um protótipo como modelo.

**Confusão do usuário de protótipo e sistema finalizado:** Os usuários podem começar a pensar que um protótipo, destinado a ser jogado fora, é na verdade um sistema final que apenas precisa ser finalizado ou polido. (Eles, por exemplo, muitas vezes desconhecem o esforço necessário para adicionar verificação de erros e recursos de segurança que um protótipo pode não ter.)

Isso pode levá-los a esperar que o protótipo modele com precisão o desempenho do sistema final quando isso não for possível. a intenção dos desenvolvedores. Os usuários também podem se apegar a recursos que foram incluídos em um protótipo para consideração e depois removidos da especificação para um sistema final. Se os usuários puderem exigir que todos os recursos propostos sejam incluídos no sistema final, isso pode levar a conflitos.

**Incompreensão do desenvolvedor sobre os objetivos do usuário:** Os desenvolvedores podem presumir que os usuários compartilham seus objetivos (por exemplo, entregar a funcionalidade principal no prazo e dentro do orçamento), sem entender questões comerciais mais amplas.



## **Gestão de Segurança da Informação**

Com a utilização dos computadores em diversas organizações, as informações começaram a se concentrar em um único lugar e o grande volume dessas informações passou a ser um problema para a segurança. Os riscos aumentaram com o uso dos microcomputadores, a utilização de redes locais e remotas, a abertura comercial da Internet e a disseminação da informática para diversos setores da sociedade.

As pequenas e médias empresas também são atingidas por estes problemas, porém dispõem de menos recursos para investir na gestão da segurança da informação.

O problema de pesquisa tratado neste trabalho é: "que fatores são capazes de influenciar a adoção da gestão da segurança da informação por pequenas e médias empresas?"

O objetivo geral foi identificar os fatores que influenciam pequenas e médias empresas a adotarem medidas de gestão da segurança da informação e avaliar o grau de importância deles. Outro objetivo foi descrever, por meio dos controles contidos na norma de segurança da informação ISO IEC 27002:2005, se as empresas pesquisadas possuem requisitos mínimos e satisfatórios de gestão da segurança da informação. Para tanto, os controles descritos na norma foram classificados em três camadas: física, lógica e humana. A empresa considerada "satisfatória" deve possuir controles efetivos nas três camadas.

Este trabalho estudou pequenas e médias empresas (PMEs) industriais presentes na região do Grande ABC, composta pelas cidades de Santo André, São Bernardo do Campo, São Caetano do Sul, Diadema, Mauá, Ribeirão Pires e Rio Grande da Serra. A categorização usada para pequenas e médias empresas foi o número de empregados, sendo: pequena empresa - de 10 a 99 empregados; média empresa - entre 100 e 499 empregados.

## **Segurança da Informação**

Segurança da informação, conforme Beal (2005), é o processo de proteção da informação das ameaças a sua integridade, disponibilidade e confidencialidade. Sêmola (2003) define segurança da informação como "uma área do conhecimento dedicada à proteção de ativos da informação contra acessos não autorizados, alterações indevidas ou sua indisponibilidade." A ISO/IEC 17799:2005, em sua seção introdutória, define segurança da informação como "a proteção da informação de vários tipos de ameaças para garantir a continuidade do negócio, minimizar o risco ao negócio, maximizar o retorno sobre os investimentos e as oportunidades de negócio". Assim, podemos definir segurança da informação como a área do conhecimento que visa à proteção da informação das ameaças a sua integridade, disponibilidade e confidencialidade a fim de garantir a continuidade do negócio e minimizar os riscos.

a integridade da informação tem como objetivo garantir a exatidão da informação, assegurando que pessoas não autorizadas possam modificá-la, adicioná-la ou removê-la, seja de forma intencional ou acidental;

A disponibilidade garante que os autorizados a acessarem a informação possam fazê-lo sempre que necessário;

A confidencialidade da informação é a garantia de que somente pessoas autorizadas terão acesso a ela, protegendo-a de acordo com o grau de sigilo do seu conteúdo;

Sêmola (2003) acrescenta a estes três objetivos os de:

Legalidade - garantia de que a informação foi produzida em conformidade com a lei;

Autenticidade - garantia de que num processo de comunicação os remetentes sejam exatamente o que dizem ser e que a mensagem ou informação não foi alterada após o seu envio ou validação.

A fim de garantir um nível de proteção adequado para seus ativos de informação, as organizações e seus principais gestores precisam ter uma visão clara das informações que estão tentando salvar, de que ameaças e por que razão, antes de poder passar a seleção de soluções específicas de segurança (BEAL, 2005). Grande parte dos dados importantes ao negócio da empresa está armaze-

nada em computadores, por isso as organizações dependem da confiabilidade de seus sistemas baseados em TI; se a confiança nesses dados for destruída, o impacto pode ser comparável à própria destruição do sistema.

Dessa forma, as organizações precisam adotar controles de segurança - medidas de proteção que abranjam uma grande diversidade de iniciativas - que sejam capazes de proteger adequadamente dados, informações e conhecimentos, escolhidos, levando-se em conta os riscos reais a que estão sujeitos esses ativos. (BEAL, 2005).

À medida que as empresas se tornam mais dependentes da informática, mais vulneráveis ficam a crimes e fraudes cometidas com o uso de recursos computacionais. Na maioria dos casos ocorridos, nada é publicado, por necessidade de preservação da imagem. (CARUSO e STEFFEN, 1999).

Pela alta capacidade de que dados, informação e conhecimento têm de adicionar valor a processos, produtos e serviços, estes constituem recursos cada vez mais críticos para o alcance da missão e dos objetivos organizacionais (CARUSO e STEFFEN, 1999).

Conseqüentemente, as informações críticas para o negócio precisam ser protegidas contra as ameaças que podem levar à sua destruição, indisponibilidade temporária, adulteração ou divulgação não autorizada. (BEAL, 2005, p. XI). Fontes (2006, p. 38) assevera "a informação é um recurso que tem valor para a organização e deve ser bem gerenciada e utilizada [...] é necessário garantir que ela esteja sendo disponibilizada apenas para as pessoas que precisam dela para o desempenho de suas atividades profissionais".

Segundo Moraes, Terence e Escrivão Filho (2004), nenhuma empresa pode escapar dos efeitos da revolução causada pela informação. Dessa forma, deve-se ter consciência de que a informação é um requisito tão importante quanto os recursos humanos, pois dela depende o sucesso ou fracasso das tomadas de decisões diárias.

Segurança - mais que estrutura hierárquica, homens e equipamentos - envolve uma postura gerencial, o que ultrapassa a tradicional abordagem da maioria das empresas. É preciso cercar o ambiente de informações com medidas que garantam sua segurança efetiva, a um custo aceitável, visto ser impossível obter-se segurança absoluta, já que a partir de um determinado ponto, os custos se tornam inaceitáveis. (CARUSO e STEFFEN, 1999).

Fontes (2006) alerta para o constante crescimento de incidentes de segurança da informação, principalmente no Brasil. De forma crescente, as organizações estão potencialmente mais expostas a novas formas de ataques, independentemente do porte ou do tipo de negócio.

Para Beal (2005), devido à alta complexidade e ao alto custo de manter os ativos da informação salvos de ameaças à sua confidencialidade, integridade e disponibilidade, é importante a empresa adotar um enfoque de gestão baseado nos riscos específicos para o negócio. Sêmola (2003) define risco como: "a probabilidade de que agentes, que são ameaças, explorem vulnerabilidades, expondo os ativos a perdas de confidencialidade, integridade e disponibilidade e causando impacto nos negócios".

Os impactos são limitados por medidas de segurança, que ajudam a diminuir o risco. Assim, a gestão do risco é o conjunto de processos que permite às organizações identificarem e implementarem as medidas de proteção necessárias para diminuir os riscos a que estão sujeitos os seus ativos de informação, e equilibrá-los com os custos operacionais e financeiros envolvidos. (BEAL, 2005).

### **Camadas de Segurança da Informação**

A todo instante os negócios, seus processos e ativos físicos, tecnológicos e humanos são alvo de investidas de ameaças de toda ordem, que buscam identificar um ponto fraco compatível, uma vulnerabilidade capaz de potencializar sua ação. Quando essa possibilidade aparece, a quebra de segurança é consumada. (SÊMOLA, 2001, p. 18).

Para Schneier (2001), "as ameaças do mundo digital espelham as ameaças no mundo físico. Se o desfalque é uma ameaça, então o desfalque digital também é uma ameaça. Se os bancos físicos são roubados, então os bancos digitais serão roubados." O crime no ciberespaço inclui tudo o que se pode esperar do mundo físico: roubo, extorsão, vandalismo, voyeurismo, exploração, jogos de trapaças, fraude etc.

Para Sêmola (2003), a gestão da segurança da informação pode ser classificada em três aspectos: tecnológicos, físicos e humanos. As organizações preocupam-se principalmente com os aspectos tecnológicos (redes, computadores, vírus, hackers, Internet) e se esquecem dos outros - físicos e humanos - tão importantes e relevantes para a segurança do negócio quanto os aspectos tecnológicos. Neste trabalho, optou-se pela classificação apresentada por Adachi (2004) que estudou a gestão da segurança em Internet Banking dividido-a em três camadas: física, lógica e humana.

### **Camada Física**

É o ambiente onde está instalado fisicamente o hardware - computadores, servidores, meio de comunicação - podendo ser o escritório da empresa, a fábrica ou até a residência do usuário no caso de acesso remoto ou uso de computadores portáteis. Para Adachi (2004), "a camada física representa o ambiente em que se encontram os computadores e seus periféricos, bem como a rede de telecomunicação com seus modems, cabos e a memória física, armazenada em disquetes, fitas ou CDs".

As pequenas e a médias empresas têm seus dados armazenados, geralmente, em servidores de rede ou em estações compartilhadas, e o acesso físico a estes equipamentos nem sempre é restrito. Na maioria das vezes, esse mesmo servidor ou estação possui acesso liberado e ilimitado à Internet, o que aumenta o risco de um incidente de segurança. Na média empresa, o cenário é menos problemático, porém não o ideal, principalmente, devido à conscientização dos funcionários sobre segurança da informação.

O controle de acesso aos recursos de TI, equipamentos para fornecimento ininterrupto de energia e firewalls são algumas das formas de se gerir a segurança desta camada.

### **Camada Lógica**

A camada lógica é caracterizada pelo uso de softwares - programas de computador - responsáveis pela funcionalidade do hardware, pela realização de transações em base de dados organizacionais, criptografia de senhas e mensagens etc. Segundo Adachi (2004), é nessa camada que estão as "regras, normas, protocolo de comunicação e onde, efetivamente, ocorrem as transações e consultas".

A segurança, em nível lógico, refere-se ao acesso que indivíduos têm às aplicações residentes em ambientes informatizados, não importando o tipo de aplicação ou o tamanho do computador. As ferramentas de controle são, em sua maior parte, "invisíveis" aos olhos de pessoas externas aos ambientes de informática; estas só os reconhecem quando têm o seu acesso barrado pelo controle de acesso. (CARUSO e STEFFEN, 1999).

Manter o software de sistema operacional atualizado com a mais recente correção de segurança disponibilizada pelo fabricante é uma forma de minimizar os riscos de segurança nesta camada.

### **Camada Humana**

A camada humana é formada por todos os recursos humanos presentes na organização, principalmente os que possuem acesso aos recursos de TI, seja para manutenção ou uso. São aspectos importantes desta camada: a percepção do risco pelas pessoas: como elas lidam com os incidentes de segurança que ocorrem; são usuários instruídos ou ignorantes no uso da TI; o perigo dos intrusos maliciosos ou ingênuos; e a engenharia social (ADACHI, 2004).

Das três camadas, esta é a mais difícil de se avaliar os riscos e gerenciar a segurança, pois envolve o fator humano, com características psicológicas, sócio-culturais e emocionais, que variam de forma individual (SCHNEIER, 2001).

A gestão da segurança da informação envolve mais do que gerenciar os recursos de tecnologia - hardware e software - envolve pessoas e processos, porém algumas empresas negligenciam este fator. A política de segurança e a conscientização dos usuários são algumas das formas de se controlar a segurança desta camada.

### **Norma de Segurança**

"Normas e padrões têm por objetivo definir regras, princípios e critérios, registrar as melhores práticas e prover uniformidade e qualidade a processos, produtos ou serviços, tendo em vista sua eficiência e

eficácia." (BEAL, 2005, p. 36). Concomitantemente, Sêmola (2003) diz que "uma norma tem o propósito de definir regras, padrões e instrumentos de controle que dêem uniformidade a um processo, produto ou serviço".

Devido ao interesse internacional em uma norma de segurança da informação, em dezembro de 2000, foi publicada a norma internacional ISO 17799:2000. Em 2001, a Associação Brasileira de Normas Técnicas (ABNT) publicou a versão brasileira que ficou com a denominação de NBR/ISO 17799 - Código de Prática para a Gestão da Segurança da Informação (OLIVA e OLIVEIRA, 2003). Em setembro de 2005, a norma foi revisada e publicada como NBR ISO/IEC 17799:2005. (ISO 17799, 2005).

Segundo Holanda (2006), o comitê que trata da segurança da informação na ISO aprovou a criação de uma família de normas sobre gestão da segurança da informação, batizada pela série 27000, onde a então ISO IEC 17799:2005 foi rebatizada por ISO IEC 27002:2005.

A norma define 127 controles que compõem o escopo do Sistema de Gestão de Segurança da Informação (Information Security Management System - ISMS), agrupados em 11 seções de controles: Política de Segurança da Informação; Organização da Segurança da Informação; Gestão de Ativos; Segurança em Recursos Humanos; Segurança Física e do Ambiente; Gestão das Operações e Comunicações; Controle de Acesso; Aquisição, Desenvolvimento e Manutenção dos Sistemas de Informação; Gestão de Incidentes da Segurança da Informação; Gestão da Continuidade do Negócio e Conformidade.

A adequação de qualquer empresa à norma ISO IEC 27002:2005 garante conformidade com as melhores práticas em gestão da segurança da informação. "As normas são criadas para estabelecerem diretrizes e princípios para melhorar a gestão de segurança nas empresas e organizações." (HOLLANDA, 2006).

Muitas seções da norma ISO IEC 27002:2005 possuem características das três camadas de segurança da informação (física, lógica e humana). Houve um esforço neste trabalho no sentido de classificar a seção pela camada que apresenta a maioria dos controles de uma delas.

### **Fatores Influenciadores para Adoção de TI ou Segurança da Informação em PMEs**

Pouca literatura foi encontrada sobre a adoção da gestão da segurança da informação em organizações de qualquer porte. Porém, devido ao fato da maioria das empresas entenderem segurança da informação como simplesmente segurança de rede ou segurança em TI, considerou-se neste trabalho que os motivos que levam à adoção de TI estão associados ou são equivalentes aos motivos que levam à adoção da gestão da segurança da informação. Seguem os autores pesquisados que tratam da adoção de TI e suas considerações:

Thong (apud Prates e Ospina, 2004) salienta que as pequenas empresas não conhecem a importância de fatores-chave em TI, além das PMEs dispuserem de recursos reduzidos, podem estar gastando recursos e energia em fatores de pouca importância para o sucesso da implementação da TI. O autor, em pesquisa realizada com 114 pequenas empresas de Singapura, concluiu que as pequenas empresas com sucesso em TI tendiam a ter alta participação de especialistas externos.

Palvia e Palvia (1999) conduziram uma pesquisa em uma amostra de 1460 pequenas empresas para verificar os padrões de satisfação com TI, onde o proprietário era também gerente, principal usuário, além de desempenhar as principais atividades de TI. Os autores concluíram que as características do proprietário têm impacto maior na satisfação em TI do que qualquer outro fator; para tanto foram considerados gênero, idade do proprietário, raça e habilidade em computação.

Outra pesquisa realizada com 25 pequenas empresas da macro-região de Ribeirão Preto - SP, Prates e Ospina (2004), identificaram que os principais motivos que levaram as empresas a implantarem TI foram: melhoria dos controles organizacionais, aumento de participação no mercado, aumento de produtividade e redução de custos. Em relação às dificuldades encontradas, a resistência pelos funcionários foi a mais expressiva, seguida pela cultura tradicional e ausência de pessoal qualificado.

Cragg e King (1993) pesquisaram os fatores motivadores e inibidores para utilização de computadores em pequenas empresas. Como fatores motivadores encontraram o que nomearam como relative advantage que se referem às economias de tempo e esforço; benefícios econômicos e diminuição de

muitas tarefas repetidas. O entusiasmo de alguns proprietários com a tecnologia e a forte influência de consultores de TI também foram fatores considerados como motivadores da adoção.

Os fatores que desencorajaram o crescimento de TI foram agrupados em: educacionais, tempo administrativo, econômicos e técnicos. Os fatores educacionais são relativos à falta de conhecimento sobre os sistemas utilizados, bem como falta de pessoas com conhecimentos específicos de análise de sistemas, design e desenvolvimento.

O fator tempo administrativo refere-se ao fato que muitos sistemas acabam consumindo considerável quantia de tempo dos gerentes no processo de implantação. Os fatores econômicos referem-se à situação econômica da empresa no momento e à análise informal de custo-benefício dos sistemas. Com pouco conhecimento técnico interno, pequenas empresas são muito confiantes no conselho e apoio que obtêm de seus fornecedores de TI, o que as limita, muitas vezes, ao uso de pacotes de aplicativos, à aceitação de limitações no software e a sua adaptação aos requerimentos do sistema.

Lunardi e Dolci (2006) realizaram uma pesquisa com 123 micros e pequenas empresas do Rio Grande do Sul e concluíram que os principais motivos que têm levado-as a adotarem TI estão relacionadas às pressões externas (os concorrentes diretos têm adotado ou por influência de clientes, fornecedores ou do próprio governo) que a empresa enfrenta e à existência de um ambiente organizacional favorável (funcionários em condições de utilizá-la e com uma estrutura organizacional adequada).

Relacionado à adoção da gestão da segurança da informação, Gupta e Hammond (2004) realizaram uma pesquisa com 138 pequenas e médias empresas nos Estados Unidos que apontou que somente 19% dos pesquisados tiveram um incidente de segurança nos últimos 12 meses, o que pode explicar a baixa porcentagem de pequenas empresas que desenvolve uma política de segurança e adquire proteção básica e software de backup.

Uma outra pesquisa realizada por Gabbay (2003) no Rio Grande do Norte, estudou os fatores que influenciam os Executivos e Gerentes de TI nas suas percepções em relação às diretrizes de Segurança da Informação na norma NBR ISO/IEC 17799 - dimensão controle de acesso. Em sua conclusão, evidenciou a associação entre as variáveis, "tamanho do parque de informática" e a "frequência dos ataques sofridos", com a variável "Nível de concordância em relação à norma NBR ISO/IEC 17799 - dimensão controle de acesso".

Esta pesquisa utilizou o método exploratório-descritivo e teve como delineamento o levantamento (survey). Para realização do estudo, foi selecionado o setor de fabricação de produtos de metal, exclusive máquinas e equipamentos, localizado na região do ABC paulista, que é o mais expressivo do cadastro da CIESP - com 256 empresas cadastradas, sendo 225 classificadas como empresas de pequeno porte e 31 empresas classificadas como médio porte.

Os sujeitos da pesquisa foram os gestores (gerentes ou proprietários) que possuam algum envolvimento no processo de aquisição ou em investimentos em gestão da segurança da informação ou em TI.

Para fornecer subsídios para criação do questionário, foram realizadas entrevistas semi-estruturadas com sete gestores de quatro organizações diferentes. As entrevistas foram realizadas no mês de setembro de 2006. Foram gravadas e tiveram duração aproximada de quarenta minutos. Em três empresas, as entrevistas foram realizadas com dois gestores simultaneamente, somente em uma das empresas, a entrevista foi individual.

Por serem semi-estruturadas, as entrevistas permitiram o acompanhamento da resposta e, quando necessário, foram efetuadas perguntas relacionadas, que não estavam incluídas no roteiro original. Isso ajudou, conforme recomenda Hair, Jr. et al. (2005), na descoberta de informações adicionais.

Procurou-se nas entrevistas conhecer primeiramente o perfil do gestor entrevistado, questionando-o sobre incidentes pessoais de segurança ocorridos anteriormente e como ele se mantém informado sobre assuntos ligados à TI e à segurança da informação. Buscou-se levantar também o perfil da empresa e saber o conhecimento do gestor sobre incidentes ocorridos com sua empresa.

O valor da informação para a empresa e o risco inerente à ela também foram objetos de questionamento, buscando-se entender como as empresas têm lidado com este tema.

Por fim, a entrevista questionou-os sobre as ferramentas e técnicas de defesa implantadas na empresa e os motivos que contribuíram ou contribuiriam para elevar os investimentos em gestão da segurança da informação.

A pesquisa foi realizada entre os meses de fevereiro e março de 2007. Foram contatadas por telefone as 256 empresas da população, sendo que destas 43 responderam ao questionário.

Entre os respondentes 84% ocupam cargos gerenciais, conforme exibido no gráfico 1, e 98% dos pesquisados possuem envolvimento sobre a decisão de compra de ferramentas e técnicas de gestão da segurança da informação ou TI.

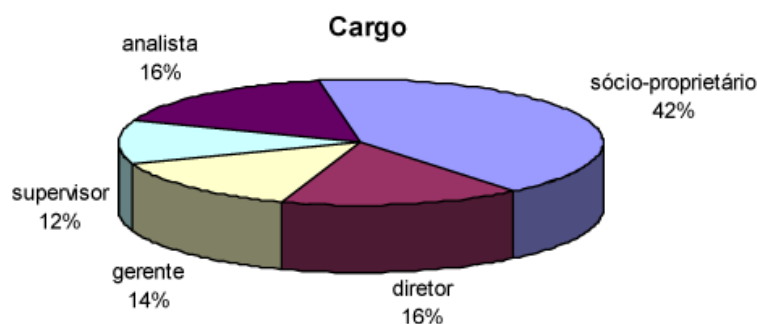


Gráfico 1: distribuição de cargos nas empresas pesquisadas

Quanto às características das empresas respondentes em relação ao porte e ao número de empregados, a amostra coletada está representada da seguinte forma, conforme o gráfico 2: 5% são microempresas (até 10 empregados), 81% são pequenas empresas (entre 10 e 99 funcionários) e 14% são médias empresas (de 100 a 499 funcionários).

A delimitação da pesquisa inclui somente pequenas e médias empresas, assim as 5% consideradas microempresas foram excluídas da amostra para as análises das ferramentas/técnicas e fatores de adoção.

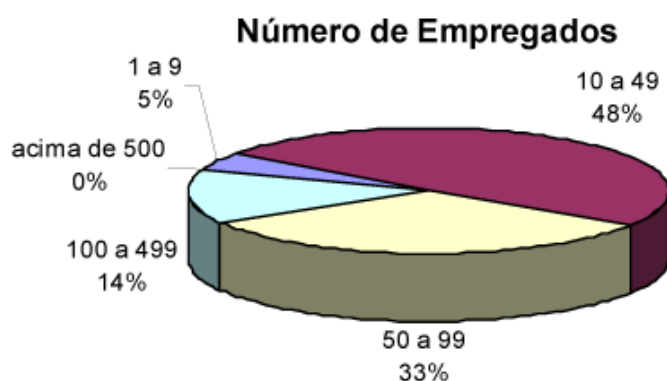
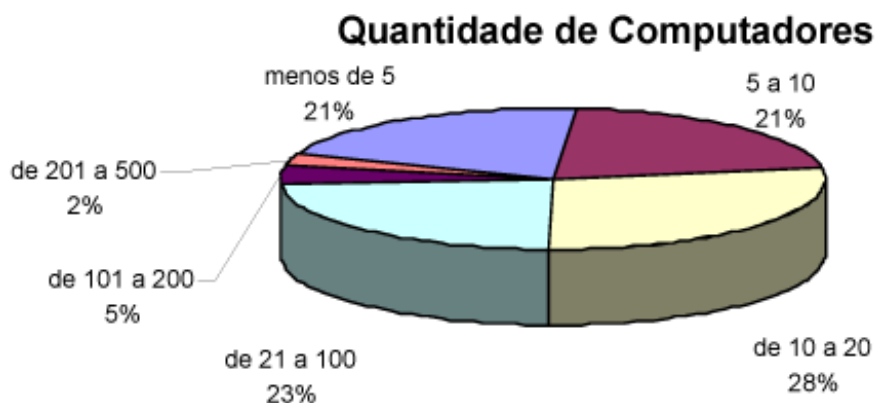


Gráfico 2: número de empregados das empresas pesquisadas

O gráfico 3 exibe a distribuição da quantidade de computadores nas empresas pesquisadas.





**Gráfico 3: quantidade de computadores**

A responsabilidade da área de TI na maioria das empresas da amostra é de um departamento interno ou funcionário (56%), enquanto os outros 44% são de empresas terceiras, sendo 23% com contrato e 21% contatadas por chamados eventuais.

Quando perguntados sobre o nível de informatização de suas operações, a maioria das empresas o considerou entre médio (65%) e alto (28%), o que pode sugerir a necessidade de uma gestão de segurança da informação mais eficaz nestas empresas devido a maior concentração de informações em computadores.

Para ajudar as empresas pesquisadas a responder sobre o nível de informatização de suas operações, foram consideradas as seguintes proposições no questionário:

Baixo: uso constante de edição de documentos, e-mails, acesso à Internet;

Médio: as considerações do nível baixo, mais uso intensivo de planilhas eletrônicas e Internet Banking;

Alto: as considerações do nível médio, mais uso de sistema integrado, acesso remoto a funcionários/fornecedores, comércio eletrônico.

### **Ferramentas e Técnicas de Gestão da Segurança da Informação**

#### **Camada Física**

Nove questões foram formuladas para representar a camada física com base nas seções: Gestão das operações e comunicações, Segurança física e do ambiente, Controle de acesso e, Gestão de incidentes de segurança da informação.

Se observarmos os gráficos de adesão das três camadas: física, lógica e humana pelas empresas pesquisadas, percebe-se que a camada humana, devido ao número de ferramentas/técnicas, é a que apresenta maior carência de cuidados por parte dos administradores. Esta constatação confirma as alegações de Schneier (2001) e as preocupações de Fontes (2006). O interessante é que muitas das ferramentas/técnicas listadas neste trabalho na camada humana não são de difícil implementação, requerem, na maioria dos casos, baixo investimento em ferramentas computacionais, tempo e dedicação da gerência.

O que confirma as considerações de Sêmola (2003) quando diz que as empresas se preocupam mais com os aspectos tecnológicos da segurança da informação do que com os aspectos físicos e humanos.

#### **Gestão da Segurança da Informação nas Três Camadas**

A fim de avaliar o nível de gestão da segurança da informação implementadas nas pequenas e médias empresas pesquisadas e, conseqüentemente, sua adequação a alguns itens da norma ISO IEC 27002:2005 foi desenvolvida a seguinte metodologia:

Verificar se a empresa possui pelo menos uma ferramenta/técnica instalada em cada uma das camadas de segurança: física, lógica e humana;

Verificar se a porcentagem das ferramentas/técnicas que a empresa possui instalada é maior ou igual a 50%, independentemente da camada de segurança;

Caso a empresa atenda às condições determinadas no item a e b, sua gestão da segurança da informação é classificada como satisfatória, caso contrário é classificada como insatisfatória.

A maioria das empresas pesquisadas (59%) se enquadraram no nível satisfatório, o que indica que existe uma preocupação da maioria das empresas com as três camadas da segurança.

### **Gestão da Segurança da Informação nas Três Camadas**

A fim de avaliar o nível de gestão da segurança da informação implementadas nas pequenas e médias empresas pesquisadas e, conseqüentemente, sua adequação a alguns itens da norma ISO IEC 27002:2005 foi desenvolvida a seguinte metodologia:

Verificar se a empresa possui pelo menos uma ferramenta/técnica instalada em cada uma das camadas de segurança: física, lógica e humana;

Verificar se a porcentagem das ferramentas/técnicas que a empresa possui instalada é maior ou igual a 50%, independentemente da camada de segurança;

Caso a empresa atenda às condições determinadas no item a e b, sua gestão da segurança da informação é classificada como satisfatória, caso contrário é classificada como insatisfatória.

A maioria das empresas pesquisadas (59%) se enquadraram no nível satisfatório, o que indica que existe uma preocupação da maioria das empresas com as três camadas da segurança.

Das empresas pesquisadas, 80% possuem pelo menos um controle em cada uma das camadas de segurança (física, lógica e humana), o que indica que as PMEs se mostram preocupadas com a gestão da segurança da informação. Quando utilizada a classificação presente neste estudo sobre a gestão da segurança da informação, 59% das empresas pesquisadas podem ser consideradas satisfatórias com os controles implantados. A ferramenta mais utilizada foi o antivírus, presente em 100% das empresas pesquisadas, seguida por sistema de backup (97,6%) e firewall (82,9%). Todos estes controles relativos à camada física.

A camada humana é a que carece de maior atenção por parte das empresas, pois foi a que apresentou o menor índice de controles implantados. Os dados confirmam que as empresas investem principalmente em controles tecnológicos para diminuir o risco de incidentes de segurança da informação, porém esquecem que o fator humano é um dos grandes responsáveis por falhas na segurança.

Em relação às seções da norma ISO IEC 27002:2005, foi verificada uma baixa adequação das pequenas e médias empresas, o que pode demonstrar que a norma requer muitos controles que a maioria não está preocupada em implantar ou não possui tempo ou dinheiro para isso. Contando que a norma sugere 127 controles e neste trabalho foram selecionados somente 20, esperava-se uma grande adequação aos controles.

Evitar perdas financeiras foi o fator motivador para adoção de gestão da segurança da informação que apresentou maior média e o único que apresentou diferença significativa das médias comparando com os demais fatores.

O fator reforça a preocupação das empresas com o lado financeiro, visto ser mais fácil de mensurar do que perda de produtividade ou imagem, por exemplo. Os demais fatores motivadores, conforme comprovaram os testes estatísticos, podem ser considerados com pesos iguais.

Não foi possível indicar o principal fator inibidor na adoção da gestão da segurança da informação, pois os testes estatísticos revelaram que todos os fatores possuíam o mesmo nível de significância. Porém, nas entrevistas realizadas com os gestores a falta de conhecimento apareceu como um possível fator inibidor e, após a realização das pesquisas quantitativas, apresentou a maior média matemática.

O presente estudo mostrou que as pequenas e médias empresas, apesar de considerarem a perda financeira como o principal fator para adoção da gestão da segurança da informação, são carentes de informações sobre a correta gestão da segurança da informação.

Para estudos futuros, recomenda-se aplicar a pesquisa em outros setores da economia como empresas de serviços ou comércio, a fim de verificar a amplitude das análises. Uma amostra maior de empresas também poderia relevar mais informações e possibilitar a indicação de um fator inibidor. Recomendam-se também estudos para verificar a causa da falta de conhecimento dos gestores em gestão da segurança da informação e TI.

O avanço acelerado das Tecnologias da Informação e da Comunicação nos últimos anos, em especial a Internet e a mobilidade constituíram a Sociedade da Informação e do Conhecimento.

A nova era tecnológica tem trazido importantes ganhos para a humanidade, proporcionando crescimento e produtividade; mas, em contrapartida, tem colocado as organizações diante de riscos inerentes ao acesso ou ao ataque às informações armazenadas nos sistemas computacionais corporativos.

A informação, ativo cada vez mais valorizado, impacta diretamente na continuidade dos negócios e na sua credibilidade. Por conta disso, as empresas têm buscado soluções para mitigar esses riscos, estabelecendo um conjunto de boas práticas por meio de políticas de segurança gerenciadas em diferentes instâncias com funções e responsabilidades bem definidas. Tudo isso para assegurar o nível de segurança adequado ao negócio.

Este é o conceito de Gestão da Segurança da Informação que abrange a criação de processos voltados ao monitoramento contínuo da integridade das informações, à prevenção de ataques e ao furto dos dados, assegurando em casos emergenciais o pronto restabelecimento dos sistemas e o acesso seguro às informações das companhias.

No nosso país, o Comitê Brasileiro sobre as Normas de Gestão de Segurança da Informação (série 27000) é responsável por normatizar essa questão. O grupo é formado por especialistas que colaboram com a ISO (International Organization for Standardization) para o desenvolvimento de padrões internacionais nesta esfera.

As normas de Gestão da Segurança da Informação se fundamentam em 10 premissas básicas aplicadas em qualquer tipo de organização, sendo elas:

- Política de Segurança da Informação
- Segurança Organizacional
- Classificação e controle dos ativos de informação
- Segurança em pessoas
- Segurança Física e Ambiental
- Gerenciamento das operações e comunicações
- Controle de Acesso
- Desenvolvimento de Sistemas e Manutenção
- Gestão da continuidade do negócio e a Conformidade.

Essas premissas abrangem o conjunto de melhores práticas a serem seguidas pelas companhias tais como: a estruturação do plano diretor de segurança e de contingência; a definição da política de segurança da informação; a análise de riscos, vulnerabilidades e testes de invasão; a implementação de controles de segurança; autenticação e autorização.

Toda essa orientação está prevista no Sistema de Gestão da Segurança da Informação (SGSI), um conjunto de processos e procedimentos, baseado em normas ISO, implementado para prover segurança no uso dos ativos tecnológicos de uma empresa. Tal sistema deve ser seguido por todos aqueles que se relacionam direta ou indiretamente com a infraestrutura de TI da organização.

A implantação do SGSI envolve primeiramente a análise de riscos na infraestrutura de TI para identificar os pontos vulneráveis e as falhas nos sistemas que deverão ser corrigidos. Em seguida, são definidos processos para detectar e responder aos incidentes de segurança e procedimentos para auditoria.

Este sistema garante segurança e integridade às informações das organizações. “Os bancos quando implementaram as primeiras soluções de internet banking ainda não possuíam soluções avançadas de Segurança da Informação alinhadas a esse conceito e tiveram grandes prejuízos com os roubos virtuais”, comenta José Antonio Antonioni, Diretor de Qualidade e Competitividade da SOFTEX, organização não-governamental que promove atividades de inovação e desenvolvimento por meio da educação, cultura e treinamento apropriados, de natureza técnica e mercadológica em Tecnologia de Software.

Com foco nas ameaças iminentes e na evolução constante da tecnologia, o Comitê Brasileiro trabalha neste momento no desenvolvimento de novas normativas relacionadas aos programas de auditoria do SGSI.

Essas diretrizes irão orientar a condução de auditorias internas e externas de acordo com a ISO/IEC 27001:2005 (Sistemas de Gestão de Segurança da Informação – Requisitos) formando auditores líderes no SGSI com capacidade analítica para identificar eventuais riscos e/ou oportunidades de melhoria no processo de Segurança da Informação.

A expectativa do Comitê é que as organizações realizem, o mais breve possível, a reestruturação da área de auditoria atualizando o conhecimento dos auditores e aplicando essas novas metodologias.

#### SGSI

É um sistema não necessariamente informatizado, embasado nas normas NBR ISO/IEC 27001:2006 e NBR ISO/IEC 27002:2006\*.

O SGSI torna-se pré-requisito à ser implementado em ambientes corporativos, educacionais, industriais, governamentais e qualquer outro que tenha por objetivo resguardar ambientes que criam, manipulam ou destroem informações relevantes.

O sistema será informatizado, caso seja necessário, conforme a peculiaridade de cada negócio (ambiente). Pois a Segurança da Informação (SI) é norteadada por boas práticas, mudança de hábitos e cultura conforme o tópico “Impactos Culturais” (parte 2 do artigo) e não apenas definidas por bons softwares e ferramentas de apoio.

#### **Tecnologias e Ferramentas de Apoio Como Segurança**

Muitos gerentes, analistas e consultores de TI dentre outros, equivocadamente associam Segurança da Informação (SI) e sua totalidade a: Antivírus, Antispyware, Antiphishing, Firewall, criptografia, soluções de DLP (Data Loss Prevention), ferramentas de monitoramento, ambientes de desenvolvimento segregados, tolerância a falhas, salas cofre e muitos outros. Entretanto estes são simplesmente alguns dos requisitos de segurança que aliados a procedimentos, políticas e processos compõem a prática de SI. Notem que ainda não citei o Compliance pois nos dará conteúdo para mais um artigo ao qual falaremos em um futuro breve. Desta forma softwares e soluções informatizadas utilizados isoladamente são apenas paliativos que maquiagem o ambiente e fornecem a falsa sensação de segurança.

#### Espinha Dorsal do Sistema

Conforme citado no primeiro tópico, as referências utilizadas são as normas, através delas e de um mapeamento prévio melhor explanado no tópico “O que Proteger?”, a PSI (Política de Segurança da Informação) é desenvolvida.

A PSI contempla as boas práticas de SI adaptadas ao cenário de cada empresa, se tornando base para criação de procedimentos e processos, juntamente com as demais políticas do âmbito de negócio e tecnologia. Após criada a PSI e antes de sua divulgação, a diretoria possui a incumbência de aprová-la e principalmente apoiá-la, pois, será fonte de consulta para tudo que envolver segurança seja ela física ou lógica.

O que proteger?



## Governança de TI

O termo "Governança de TI" tem suas origens no conceito de "Governança Corporativa", mas não devemos confundir esses conceitos. Além disso, a "Governança de TI" relaciona-se com a "Gestão/gerenciamento de TI", porém não são a mesma coisa. Para melhor entendermos as semelhanças, as diferenças e o inter-relacionamento do significado desses termos, seguem algumas definições.

### Governança Corporativa

Segundo o Instituto Brasileiro de Governança Corporativa (IBGC):

"Governança Corporativa é o sistema pelo qual as organizações são dirigidas, monitoradas e incentivadas, envolvendo os relacionamentos entre proprietários, Conselho de Administração, Diretoria e órgãos de controle. As boas práticas de Governança Corporativa convertem princípios em recomendações objetivas, alinhando interesses com a finalidade de preservar e otimizar o valor da organização, facilitando seu acesso a recursos e contribuindo para sua longevidade."

### Sistema de Governança Corporativa



Para o Ministro Aroldo Cedraz, "Governança de TI é o conjunto estruturado de políticas, normas, métodos e procedimentos destinados a permitir à alta administração e aos executivos o planejamento, a direção e o controle da utilização atual e futura de tecnologia da informação, de modo a assegurar, a um nível aceitável de risco, eficiente utilização de recursos, apoio aos processos da organização e alinhamento estratégico com objetivos desta última. Seu objetivo, pois, é garantir que o uso da TI agregue valor ao negócio da organização." (Voto do Ministro Relator – Acórdão 2.308/2010 – Plenário)

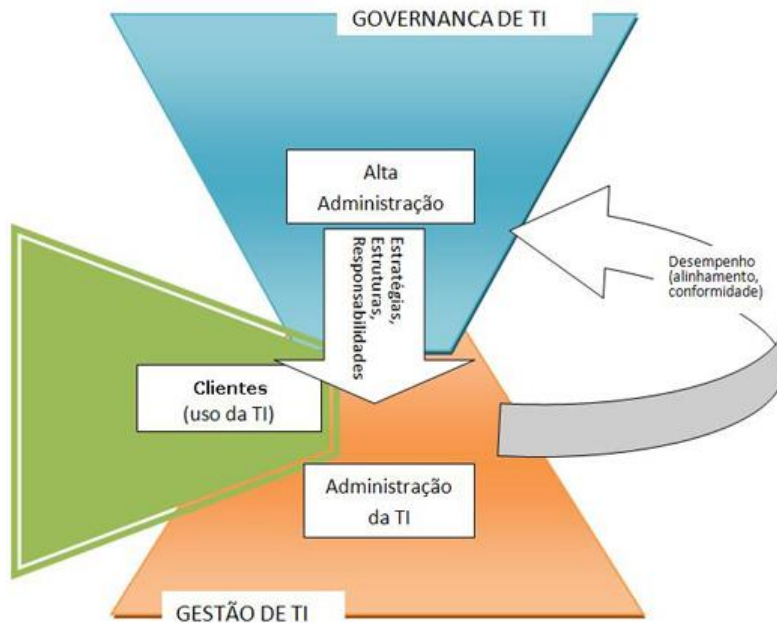
Para o Information Technology Governance Institute (ITGI), "governança de TI é de responsabilidade dos executivos e da alta direção, consistindo em aspectos de liderança, estrutura organizacional e processos que garantam que a área de TI da organização suporte e aprimore os objetivos e as estratégias da organização."

Observa-se, portanto, que a "Governança Corporativa" tem foco no direcionamento e monitoramento da gestão da instituição, e busca permitir a intervenção dos responsáveis finais sempre que houver desvio em relação ao esperado. Em última instância, esses responsáveis são os detentores da propriedade: sócios e acionistas, no caso das organizações privadas, e a sociedade, no caso das organizações públicas federais.

Já a "Governança de TI" tem foco no direcionamento e monitoramento das práticas de gestão e uso da TI de uma organização, tendo como indutor e principal beneficiário a alta administração da instituição.

Um exemplo prático de mecanismo de governança de TI é o estabelecimento de um processo transparente de tomada de decisão sobre a priorização de grandes demandas de TI. Tal processo é necessário para garantir que as ações de TI estejam alinhadas com os objetivos institucionais e para

garantir que as demandas que tenham maior impacto nesses objetivos tenham atendimento prioritário. Esta é uma decisão que não cabe às unidades de TI (embora devam sempre opinar). Portanto, o estabelecimento desse processo, os participantes e suas competências é uma iniciativa de governança de TI a ser liderada pela alta administração.



Conjunto de processos realizados pelas unidades provedoras de TI, visando ao planejamento e à realização das atividades necessárias ao provimento ou entrega de soluções e serviços de TI.

Relação entre os conceitos

Do que vimos, pode-se pensar, erroneamente, que a Governança Corporativa não tem relação com a Governança de TI e que esta não tem relação com a Gestão/Gerenciamento de TI.

Entretanto, o que de fato ocorre é uma dependência entre elas, tal como mostrado na figura abaixo. O gerenciamento de serviços de TI é, de fato, um enabler (facilitador) da governança de TI e esta é um facilitador da governança corporativa. Em resumo, uma organização que faça uso intenso de TI (situação comum, atualmente), não pode afirmar ter uma boa governança corporativa sem ter boa governança de TI. Igualmente, a instituição não pode afirmar ter uma boa governança de TI sem ter um bom gerenciamento de (serviços) de TI. A figura abaixo ilustra essa interdependência.



### Áreas De Foco Da Governança De TI

Na prática, a Governança de TI se traduz em um conjunto de políticas, processos, papéis e responsabilidades associados a estruturas e pessoas da organização, de modo a se estabelecer claramente o processo de tomada de decisões e as diretrizes para o gerenciamento e uso da TI, tudo isso de forma alinhada com a visão, missão e metas estratégicas da organização.

A figura abaixo resume as principais dimensões focalizadas pela Governança de TI.

Áreas de foco da Governança de TI Na prática, a Governança de TI se traduz em um conjunto de políticas, processos, papéis e responsabilidades associados a estruturas e pessoas da organização, de modo a se estabelecer claramente o processo de tomada de decisões e as diretrizes para o gerenciamento e uso da TI, tudo isso de forma alinhada com a visão, missão e metas estratégicas da organização. A figura abaixo resume as principais dimensões focalizadas pela Governança de TI.



**Alinhamento Estratégico** - assegura o alinhamento dos planos da TI com os de negócio e alinha a operação e as entregas da TI com as operações da organização.

**Entrega de Valor** – assegura que os benefícios previstos pela TI estão realmente sendo gerados, dentre eles a otimização de custos e outros valores intrínsecos que a TI pode proporcionar.

**Gestão de Riscos** – Permite que a organização reconheça todos os riscos (e oportunidades) derivados da TI para o negócio e que decida e tenha planos para mitigá-los na medida que julgue necessário.

**Gestão de Recursos** – assegura a gestão dos recursos mais importantes para TI: recursos humanos e recursos tecnológicos (informações, infraestrutura, aplicações). Promove a valorização do conhecimento e da infraestrutura.

**Mensuração de Desempenho** – acompanha e monitora a implementação da estratégia, consumação de projetos, uso dos recursos e entrega dos serviços quanto à sua contribuição para as estratégias e objetivos do negócio, utilizando-se não apenas de critérios financeiros. Resumindo tudo em uma figura:



**Importância De Se Aplicar Governança De TI No TCU**

O TCU é uma instituição que depende de informação para a realização de seus trabalhos, e cada vez mais da Tecnologia da Informação (TI) para adequadamente tratar, analisar, fazer uso, disseminar e proteger essas informações. Além disso, é cada vez maior a automação de processos de trabalho do Tribunal, como meio de se assegurar o alcance e a manutenção de padrões de desempenho e qualidade compatíveis com as necessidades da sociedade brasileira.



Entretanto, não é suficiente simplesmente reconhecer a importância da TI e aplicar recursos e esforços em tais iniciativas. No mercado, são incontáveis os exemplos de projetos de TI que fracassaram pela falta ou imaturidade dos mecanismos de governança de TI.

Organizações bem-sucedidas reconhecem os benefícios da tecnologia da informação e a utilizam para adicionar valor ao negócio, valendo-se de mecanismos de governança de TI adequados às estratégias institucionais.

### **Compreendendo Os Principais Conceitos Do COBIT 5**

O COBIT 5 é a mais recente versão do framework de boas práticas de governança e gerenciamento empresarial de TI, que incorpora muitos conceitos e teorias amplamente aceitos.

Nesta série de artigos, vamos explorar os princípios fundamentais do framework, servindo como uma referência para a aplicação do COBIT 5 em sua organização. Acompanhe!

### **A Governança Empresarial De TI E O COBIT 5**

Informação (no sentido mais abrangente da palavra) e tecnologias relacionadas estão se tornando fatores cruciais na sustentabilidade, crescimento e gerenciamento do valor e risco na maioria das empresas. Como resultado, a TI deixou de atuar simplesmente no papel de suporte e passou a assumir uma posição central dentro das empresas.

O papel realçado da TI na criação de valor e gerenciamento de risco empresarial veio acompanhado por uma crescente ênfase na Governança e Gerenciamento Empresarial de TI. Os stakeholders anseiam por assegurar que a TI cumpra as metas empresariais.

A Governança e Gerenciamento Empresarial de TI é parte integral de todo o meio corporativo. Ela endereça a definição e implementação de processos, estruturas e mecanismos relacionais dentro da empresa, que permitem ao pessoal de negócio e da TI executar suas responsabilidades para suportar a criação e sustentabilidade do valor ao negócio.

A Governança e Gerenciamento Empresarial de TI é complexa e multifacetada. Membros do comitê de governança e a alta direção tipicamente precisam de assistência para implementá-la. Através dos anos, frameworks de boas práticas foram desenvolvidos e promovidos para auxiliar neste processo.

Lançado em 2012, o COBIT 5 foi construído e integrado com base em 20 anos de desenvolvimento neste campo de atuação. Desde os seus primórdios, centrado na comunidade de auditoria de TI, o COBIT se tornou um framework de Governança e Gerenciamento de TI mais abrangente, compreensivo e aceito.

O COBIT 5 foi adicionalmente complementado com os frameworks Val IT e Risk IT. Antes do COBIT 5, o Val IT endereçava processos de negócio e responsabilidades na criação de valor empresarial e o Risk IT fornecia uma visão de negócio holística sobre o gerenciamento de riscos. Agora, ambos estão incorporados ao COBIT 5.

O framework COBIT 5 é construído em torno de cinco princípios fundamentais:

Satisfazer necessidades das partes interessadas;

Cobrir a organização de ponta a ponta;

Aplicar um framework integrado e único;

Possibilitar uma visão holística;

Separar Governança do Gerenciamento.

### **Entendendo Os Principais Conceitos Do COBIT**

As boas práticas em destaque na ITIL são tão importantes que viraram um padrão de fato no mundo da TI. Seus fundamentos são usados nos níveis táticos e operacionais, possibilitando que o setor da

Tecnologia da Informação estrutura o ciclo vital de seus serviços de modo geral, a fim de buscar excelência operacional.

No caso do framework do COBIT, tem foco na área estratégica e — por se tratar de um framework voltado para o controle — permite que a TI consiga seu desempenho mensurado e seus riscos devidamente apontados e consertados.

Estudando o framework do COBIT com ampla profundidade, torna-se possível saber que ele aponta os objetivos de controle. Porém, percebe-se que não existe detalhamento de como os processos podem ser definidos.

O COBIT 5 não é um padrão, ele não é um regramento como a ISO 9.001, ISO 20.000 ou ISO 17.799. Ressalta-se que ele não serve também como meio para ampliar as vantagens da TI. Em vez disso, o COBIT auxilia o direcionamento ou a priorização dos recursos e esforços da TI para atender aos requisitos do negócio.

O uso do COBIT não tem como objetivo ter o controle de todos os processos. Sua meta é apenas identificar e compreender quais são os processos da TI que estão gerando um impacto, ou ocasionando possíveis riscos para o negócio, de forma a dar prioridade à gestão desses processos.

O framework de controle do COBIT argumenta que não é possível administrar aquilo que não se mede. Posto isso, ele apresenta uma sequência de metas de controle e seus respectivos indicadores de desempenho.

### **Missão**

O objetivo deve ser a pesquisa, o desenvolvimento, a publicação e a promoção de um conjunto de metas de controle para tecnologia que seja apoiado e admitido em geral para o uso do dia a dia de administradores de negócio e auditores.

### **Comparação**

O framework do COBIT foi produzido tendo como fundamentais aspectos a base do negócio, a orientação a métodos, além de ser apoiado em controles e conduzido por métricas.

Optar pelo COBIT auxilia um empreendimento a colocar em prática boas maneiras de governança de TI, uma vez que ele fornece um guia de melhores métodos e direcionamento.

Sua sustentação classifica os processos em 4 comandos, e apresenta ações em uma estrutura gerenciável e lógica.

Vários modelos, referências e guias de boas práticas podem ser escolhidos para moldar um modelo de governança de TI para as empresas. Cabe aos executivos constatar qual é o modelo ideal para cumprir com as necessidades de negócio de suas empresas, mas é notório que a regulamentação externa (SOX/Basiléia II) direciona fortemente a opção do COBIT em seus métodos de governança de TI.

Uma questão bastante significativa é o que o COBIT — devido ao fato de ser um framework de controle de nível altíssimo — indica o que precisa ser controlado, mas não diz a maneira como isso deve ser feito. Ele se encaixa muito bem com os melhores métodos para gestão de serviços de TI descritos na IT(Infrastructure Library -ITIL), que possui foco mais específico e operacional no tocante aos processos internos de TI.

Os frameworks do COBIT e do ITIL se acrescentam e cobrem uma boa área dos aspectos da organização da TI, de modo que, quando as práticas acertadas em cada modelo são adotadas pelas empresas de TI, em seus processos íntimos, o perigo operacional de TI é diminuído de forma significativa. Tudo isso tem o objetivo de fazer com que a TI se torne um aliado estratégico para que os empreendimentos consigam alcançar seus objetivos de negócio.

A Tecnologia da Informação é moderadamente nova se comparada à Medicina, Engenharia, Arquitetura ou Advocacia. Contudo, a reunião de boas práticas vem passando por um ciclo de melhoria contínua cujos resultados positivos são comprovados pelo mercado há pelo menos 10 anos.

Como dito previamente, o framework de controle do COBIT parte da ideia inicial de que não é possível administrar aquilo que não se mede. Diante disso, ele propõe uma série de metas de gestão e seus respectivos indicadores de desempenho.

O modelo também leva em conta que a TI precisa entregar a informação que o empreendimento precisa para atingir os seus objetivos de negócio. Além disso, o COBIT é harmonizável com outros padrões de mercado, visto que ele se posiciona em um nível geral, abrangendo diversos processos de TI, definindo as metas de cada um dos processos e como elas devem ser utilizadas.

O COBIT não mira em como cada processo deve ser executado, sendo exatamente esse o motivo que o torna adaptável a outros modelos existentes.

#### Características

O framework do COBIT foi criado tendo como qualidades fundamentais:

A orientação a processos;

A meta no negócio;

Ser direcionado por métricas;

Ser baseado em controles.

#### Objetivos

Abaixo podemos ver o que o COBIT possui como metas:

Colocar as melhores práticas a partir de uma matriz de atividades, domínios e processos organizados de modo lógico e gerenciável;

Ser um padrão aprovado nas melhores práticas de governança de TI;

Ajudar na associação entre as necessidades de controle, aspectos tecnológicos e riscos do negócio.

Optar pelo COBIT como forma de governança é vantajoso por:

Auxiliar a compreender os requisitos regulatórios;

Ser conciliável tanto com o COSO quanto ao controle do ambiente de TI;

Ser focado nos requisitos de negócio;

Determinar uma linguagem comum entre TI e o negócio;

Ser aceito internacionalmente como framework de modelo para Governança de TI;

Ser suportado por treinamentos e ferramentas;

Mapear os melhores padrões e frameworks de mercado, como o ISO 20.000, ISO 27.001 e ITIL.

Ser orientado a processos;

Estar em crescimento contínuo.

#### **O Cubo do COBIT**

Como todas as peças do COBIT estão organizados em conjunto, a figura do cubo é usada para abreviar que os recursos de TI são administrados pelos processos de TI. Eles visam conseguir metas que correspondam aos requisitos do negócio. Esse é o princípio vital do framework do COBIT.

No tocante à similaridade com o ITIL, o COBIT resguarda grande parte dos processos ITIL, tanto na opção 2 quanto na opção 3. No entanto, o ITIL possui os processos demonstrados com maior nível de detalhamento.

De modo geral, enquanto o ITIL está mais focado em “como”, o COBIT foca no “o que”. Dessa forma, podemos dizer que o COBIT é um framework de controle que estabelece o que tem que ser realizado, mas não diz como proceder.

### **ISO/IEC 38500**

Esta Norma oferece princípios para orientar os dirigentes das organizações (incluindo proprietários, diretores, parceiros, executivos ou similares) sobre o uso eficaz, eficiente e aceitável da Tecnologia de Informação (TI) dentro de suas organizações.

Aplica-se a organizações de todos os tamanhos (pequenas a grandes), independentemente da extensão de seus usos de TI.

A norma ISO/IEC 38500:2008 define seis princípios para a boa governança de TI:

**Responsabilidade:** Os indivíduos e grupos na organização devem compreender e aceitar as suas responsabilidades no fornecimento de TI. Os indivíduos responsáveis por ações devem ter a autoridade para as desempenharem.

**Estratégia:** A estratégia de negócio da organização tem em conta as capacidades de TI atuais e futuras; os planos estratégicos para as TI satisfazem as necessidades atuais e contínuas da estratégia de negócio da organização.

**Aquisições:** As aquisições de TI são feitas por razões válidas, com base e análise apropriada com decisões claras e transparentes. Há um equilíbrio adequado entre os benefícios, oportunidades, custos e riscos, tanto no curto como no longo prazo.

**Desempenho:** A TI é adequada à finalidade de suporte da organização, à disponibilização de serviços e quanto aos níveis e qualidade dos serviços necessários para responder aos requisitos atuais e futuros do negócio.

**Conformidade:** A TI encontra-se em conformidade com a legislação e regulamentos aplicáveis. As políticas e as práticas estão claramente definidas, encontram-se implementadas e são aplicadas.

**Comportamento Humano:** As políticas, práticas e decisões de TI revelam respeito pelo Comportamento Humano, incluindo as necessidades atuais e a evolução das necessidades de todas as “pessoas no processo”.

A norma preconiza que os dirigentes governem a TI através de três tarefas principais:

Avaliar o uso atual e futuro da TI.

Orientar a preparação e a implementação de planos e políticas para assegurar que o uso da TI atenda aos objetivos do negócio.

Monitorar o cumprimento das políticas e o desempenho em relação aos planos.

Se bem utilizada, a norma assegura que os dirigentes poderão avaliar os riscos da TI para o negócio e aproveitar as oportunidades advindas com o uso da TI.

### **Cobit**

O que é COBIT?

O COBIT fornece um modelo abrangente que auxilia as organizações a atingirem seus objetivos de governança e gestão de TI.

Em termos simples, O COBIT 5 ajuda as organizações a criar valor por meio da TI mantendo o equilíbrio entre a realização de benefícios, a otimização dos níveis de risco e de utilização dos recursos. O COBIT permite que a TI seja governada e gerida de forma holística para toda a organização, abrangendo o negócio de ponta a ponta bem como todas as áreas responsáveis pelas funções de TI, levando em consideração os interesses internos e externos relacionados com TI.

O COBIT é genérico e útil para organizações de todos os portes, sejam comerciais, sem fins lucrativos ou públicas.

O COBIT baseia-se em cinco princípios básicos (demonstrados na figura) para governança e gestão de TI da organização:



### Por que usar o COBIT 5?

Novas exigências dos usuários, os regulamentos específicos do setor e cenários de risco surgem a cada dia. Maximizar o valor da propriedade intelectual, gestão do risco e segurança e assegurar a conformidade por meio de governança de TI nunca foi tão importante.

Nenhuma outra estrutura focada em TI corporativa oferece a amplitude ou benefícios do COBIT. Ele ajuda empresas de todos os tamanhos a:

Manter a informação de alta qualidade para apoiar decisões de negócios;

Atingir os objetivos estratégicos através da utilização eficaz e inovadora de TI;

Atingir a excelência operacional através da aplicação confiável, eficiente da tecnologia;

Manter riscos a um nível aceitável;

Otimizar o custo de serviços de TI e tecnologia;

Manter o cumprimento de leis, regulamentos, acordos contratuais e políticas.

O framework COBIT para a governança e gestão de TI empresarial é uma otimização de negócios que utiliza práticas comprovadas, liderança com pensamento global e ferramentas inovadoras para inspirar a inovação em TI e negócios de sucesso.

### Certificações Relacionadas

A ISACA oferece três níveis de certificação profissional, relacionados ao conhecimento e a proficiência na utilização do CobiT:

COBIT 5 Foundation: atesta que os profissionais certificados compreendem os problemas de governança e gestão da TI das empresas e sabem como utilizar o CobiT para enfrentar e solucionar esses desafios.



## O que é a Tecnologia Java e porque preciso dela?

Java é uma linguagem de programação e plataforma computacional lançada pela primeira vez pela Sun Microsystems em 1995. Existem muitas aplicações e sites que não funcionarão, a menos que você tenha o Java instalado, e mais desses são criados todos os dias. O Java é rápido, seguro e confiável. De laptops a datacenters, consoles de games a supercomputadores científicos, telefones celulares à Internet, o Java está em todos os lugares!

O download do Java é gratuito?

Sim, o download do Java é gratuito. Obtenha a última versão no site [java.com](http://java.com).

Se você estiver criando um dispositivo de consumo ou incorporado e quiser incluir o Java, entre em contato com a Oracle para obter mais informações sobre como incluir o Java no seu dispositivo.

Por que devo fazer upgrade para a versão mais recente do Java?

A versão mais recente do Java contém aprimoramentos importantes para melhorar o desempenho, a estabilidade e a segurança das aplicações Java executadas na sua máquina. Ao instalar essa atualização gratuita, você garantirá que as suas aplicações Java continuem sendo executadas de forma segura e eficiente.

## Mais Informações Técnicas

O que eu receberei quando fizer o download do software Java?

O Java Runtime Environment (JRE) é o que você recebe ao fazer o download do software Java. O JRE é composto pela Java Virtual Machine (JVM), pelas classes de núcleo da plataforma Java e bibliotecas da plataforma Java para suporte. O JRE é a parte de runtime do software do software Java, que é o necessário para executá-lo no seu Web browser.

O que é o software Java Plug-in?

O software Java Plug-in é um componente do Java Runtime Environment (JRE). O JRE permite applets gravados na linguagem de programa Java para serem executados dentro de vários browsers. O software Java Plug-in não é um programa stand-alone e não pode ser instalado separadamente.

Eu ouvi os termos Java Virtual Machine e JVM. Eles são o software Java?

A Java Virtual Machine é apenas um aspecto do software Java envolvido na interação Web. A Java Virtual Machine é criada diretamente no seu download de software Java e ajuda a executar aplicações Java.

## Obtenha Informações sobre a Tecnologia Java

O Java é a base para praticamente todos os tipos de aplicações em rede e é o padrão global para o desenvolvimento e distribuição de aplicações móveis e incorporadas, jogos, conteúdo baseado na Web e softwares corporativos. Com mais de 9 milhões de desenvolvedores em todo o mundo, de forma eficiente, o Java permite que você desenvolva, implante e use aplicações e serviços estimulantes.

De laptops a datacenters, consoles de games a supercomputadores científicos, telefones celulares à Internet, o Java está em todos os lugares!

- 97% dos Desktops Corporativos executam o Java
- 89% dos Desktops (ou Computadores) nos EUA Executam Java
- 9 Milhões de Desenvolvedores de Java em Todo o Mundo
- A Escolha Nº 1 para os Desenvolvedores
- Plataforma de Desenvolvimento Nº 1

- 3 Bilhões de Telefones Celulares Executam o Java
- 100% dos Blu-ray Disc Players Vêm Equipados com o Java
- 5 bilhões de Placas Java em uso
- 125 milhões de aparelhos de TV executam o Java
- 5 dos 5 Principais Fabricantes de Equipamento Original Utilizam o Java ME

#### Porque os Desenvolvedores de Software Escolhem o Java

O Java foi testado, refinado, estendido e comprovado por uma comunidade dedicada de desenvolvedores, arquitetos e entusiastas do Java. O Java foi projetado para permitir o desenvolvimento de aplicações portáteis de alto desempenho para a mais ampla variedade possível de plataformas de computação. Ao disponibilizar aplicações entre ambientes heterogêneos, as empresas podem fornecer mais serviços e aumentar a produtividade, a comunicação e a colaboração do usuário final — além de reduzir drasticamente o custo de propriedade das aplicações da empresa e do consumidor. O Java tornou-se inestimável para os desenvolvedores, permitindo que eles:

- Gravem software em uma plataforma e o executem virtualmente em qualquer outra plataforma
- Criem programas que podem ser executados dentro em um web browser e acessem web services disponíveis
- Desenvolvam aplicações do servidor para fóruns on-line, armazenamentos, pesquisas, processamento de forms HTML e mais
- Combinem aplicações ou serviços usando a linguagem Java para criar aplicações ou serviços altamente personalizáveis.
- Crie aplicações potentes e eficientes para telefones celulares, processadores remotos, microcontroladores, módulos sem fio, sensores, gateways, produtos de consumo e praticamente qualquer outro dispositivo eletrônico

#### Algumas Maneiras Pelas Quais Desenvolvedores de Software Aprendem Java

Muitas faculdades e universidades oferecem cursos de programação para a plataforma Java. O Oracle Academy oferece um portfólio completo de software, conteúdo programático, tecnologia hospedada, treinamento acadêmico, suporte e recursos de certificação para instituições educacionais de ensino fundamental, formação profissional e ensino superior para utilização no ensino, incluindo uma oferta do Java que dará suporte a centenas de milhares de estudantes. Além disso, os desenvolvedores também podem aumentar suas habilidades de programação em Java lendo o site do desenvolvedor do Java da Oracle, inscrevendo-se para receber newsletters voltadas para a tecnologia Java e a Java Magazine, usando o Tutorial Java e o New to Java Programming Center e inscrevendo-se em cursos e certificações virtuais pela Web ou presenciais.

A Oracle Technology Network é a maior comunidade mundial de desenvolvedores de aplicações, administradores de banco de dados, administradores/desenvolvedores de sistema e arquitetos que usam tecnologias padrão da indústria em combinação com produtos Oracle. Ela também é a página inicial do [java.oracle.com](http://java.oracle.com) e a fonte mais recente, completa e oficial de informações técnicas sobre o Java. A associação é gratuita. Associe-se hoje! (Em seu Perfil, marque a caixa Oracle Technology Network em Minhas Associações da Comunidade.)

#### Jovens Desenvolvedores Aprendem Java

Os jovens estão aprendendo as linguagens de programação desde bem novos. As ferramentas visuais educacionais como Alice, Greenfoot e BlueJ ensinam aos jovens como programar usando a linguagem de programação Java e as linguagens com base em Java, desenvolvidas para facilitar o uso.

O que é o JavaFX?



A plataforma JavaFX é desenvolvida pelo Java. A plataforma JavaFX permite que desenvolvedores de aplicações criem e implantem facilmente RIA (Rich Internet Applications) que se comportam de forma consistente em várias plataformas. O JavaFX expande o poder do Java, permitindo que os desenvolvedores usem qualquer biblioteca Java em aplicações JavaFX. Os desenvolvedores podem expandir suas habilidades em Java e aproveitar a tecnologia de apresentação que o JavaFX fornece para criar experiências visuais envolventes.

### **Comece a programar: A Linguagem de Programação Java**

Portabilidade, sabe o que é? É programar em Windows, Linux, no Mac, pra Web, pra celular, em uma pedra...sem se preocupar com compatibilidade. Como é possível?  
Compatibilidade é o que mais atormenta os programadores!

Muito simples, Java não roda no computador! Roda em uma máquina virtual!

O que é o Java? Como surgiu? Para que serve? Onde posso utilizar? Que programas conhecidos são feitos em Java? É verdade que Java é só moda? Programar em Java é emprego garantido? É a mais fácil? Por que todo mundo tá estudando e falando de Java?

### **A linguagem de programação Java**

O site Programação Progressiva disponibiliza um curso completo de Java, o Java Progressivo.

Java é uma linguagem de programação orientada a objetos feita na Sun Microsystems, hoje Oracle Corporation, lançada em 1995.

A semelhança da sintaxe do Java com C e C++ não é coincidência, derivou dessas linguagens mesmo. Porém, programar em Java é mais simples, pois é alto nível. Isso quer dizer que não nos preocupamos tanto com detalhes baixo nível, como memória, processamento, ponteiros, lixo etc. O Java já provém um gerenciamento automático de memória e um coletor de lixo, que facilitam a vida do desenvolvedor, mas consomem mais processamento.

A diferença do Java é que os programas não são compilados diretamente na arquitetura do computadores. Ao invés disso, roda na JVM - Java Virtual Machine, uma máquina virtual, e esta é implementada nos mais diversos dispositivos, o que torna o Java referência quando o assunto é portabilidade.

Em outras linguagens de programação, como em C, o programa é convertido em código de máquina (Assembly) e rodará especificamente na sua máquina. Se tentar rodar em um celular, não irá conseguir, pois é outra 'máquina', outra arquitetura.

Porém, não existe esse 'código de máquina' em Java. O correspondente é o 'bytecode', que é um código que executa na JVM.  
Notou a sacada?

Atualmente, 2012, Java é uma das linguagens de programação mais famosas do mundo, principalmente pelas aplicações Web. Se você usa Internet, é quase que impossível não ter usado Java para ter tido acesso ao site daquele banco, ou usou para entrar numa rede social ou jogar um jogo online.

Java ajudou a desassociar a imagem de 'programação' com 'computador'. É a dita portabilidade. Hoje em dia, quando falamos em programação também nos referimos aos aparelhos móveis. Se você é programador e acha que computação é uma tela preta ou programar direto no hardware e não se importa com celulares, androids, iPhone, iPad, Tablet e outros: sinto muito, você é quem está perdendo.

As ATM, ou caixas-eletrônicos no Brasil, também estão usando e abusando de Java. Até a NASA já lançou robôs em outros planetas que usavam softs feitos em Java. Sim, Java, na verdade, é uma das linguagens mais usadas no Universo ;)

## O segredo do Java

A base da programação Java são as classes e seus objetos, que 'imita' o mundo real, o que facilita bastante a programação.

Por exemplo, os carros são uma classe, já um gol é um objeto da classe carro, um fusca também é um objeto da classe carro.

O poodle é um objeto da classe cachorro, assim como o maltês.

As classes possuem métodos e características que são comuns a todos os objetos. Por exemplo, todos os objetos da classe carro possuem motor e rodas. Porém, os tipos de motores podem variar (isto é uma característica específica de cada objeto, mas que possui motor, sempre possui).

Essa associação com o mundo real ajuda bastante na hora da abstração, de criar aplicações complexas.

O Java é bastante flexível por conta da possibilidade de expansão através das bibliotecas, ou APIs, além das extensões do Java, voltadas especificamente para desenvolvimento de aplicações para desktop, para celulares, para empresas, para áudio, para gráficos 3D, banco de dados, para aplicações de Internet, criptografia, computação/sistemas distribuídos, linguagem de marcação, infra estrutura peer-to-peer e várias outras.

Através dessas extensões, é possível desenvolver praticamente qualquer coisa que você se interesse, em Java, de uma maneira bem mais documentada e específica.

O que é preciso para rodar Java?

Para rodar aplicações em Java você precisa ter instalado a JRE, Java Runtime Environment.

Já para desenvolver aplicações, você vai precisar da JDK - Java Development Kit.

Para ajudar, use um ambiente de desenvolvimento, um IDE, como o NetBeans.

Tudo isso, além da documentação (todas as informações da linguagem Java), você pode encontrar aqui:

<http://java.sun.com/products/jfc/tsc/sightings/>

Lembrando que existe um curso completo de Java no Programação Progressiva.

Embora você possa criar aplicações para desktop, para empresas e para seus amigos, a principal utilidade do Java são as aplicações Web e mobile. Isso por conta da variedade de arquiteturas de celulares, ipad, iphone e computadores. Imagine se os sites tivesse que desenvolver um portal para cada tipo de sistema operacional ou máquina diferente? Haveriam dezenas de opções.

Ao invés disso, eles desenvolvem em Java, você baixa a JRE e todos usufruem das aplicações.

Um ponto fraco do Java, em relação a outras linguagens de programação, é o peso. É um pouco lento, principalmente se compararmos com a eficiência de linguagens como C e C++.

## Fundamentos da linguagem Java

Sobre este tutorial

O tutorial de duas partes Introdução à programação Java destina-se aos desenvolvedores de software que conhecem pouco da tecnologia Java. Execute as duas partes para o funcionamento com a programação orientada a objeto (OOP) e desenvolvimento de aplicativo real usando a linguagem e plataforma Java.

Esta primeira parte é uma introdução etapa a etapa à OOP, usando a linguagem Java. O tutorial começa com uma visão geral da plataforma e linguagem Java, seguida de instruções para configurar um ambiente de desenvolvimento que consiste de uma Java Development Kit (JDK) e Eclipse IDE. Depois da introdução aos componentes do ambiente de desenvolvimento, você começa a aprender a prática de sintaxe básica Java.

Parte 2 abrange recursos de linguagem mais avançados, incluindo expressões regulares, genéricas, E/S e serialização. Os exemplos de programação na Parte 2 criam o objeto Person que você começa a desenvolver na Parte 1.

### Objetivos

Ao concluir a Parte 1, você estará familiarizado com a sintaxe de linguagem básica Java e capaz de escrever programas Java simples. Acompanhe "Introdução à programação Java, Parte 2: Construções para aplicativos reais" para criar esta base.

### Pré-requisitos

Este tutorial destina-se a desenvolvedores de software que ainda não possuem experiência no código ou plataforma Java. O tutorial inclui uma visão geral dos conceitos de OOP.

### Requisitos do sistema

Para concluir os exercícios neste tutorial, instale e configure um ambiente de desenvolvimento que consiste de:

- JDK 8 em Oracle
- Eclipse IDE for Java Developers

As instruções de download e instalação para ambos estão incluídas no tutorial.

A configuração do sistema recomendada é:

- Um sistema que suporte Java SE 8 com pelo menos 2 GB de memória. Java 8 é suportado em Linux®, Windows®, Solaris® e Mac OS X.
- Pelo menos 200 MB de espaço em disco para instalar os componentes de software e exemplos.

### Visão geral da plataforma Java

A tecnologia Java é usada para desenvolver aplicativos para uma ampla variedade de ambientes, de dispositivos consumidores a sistemas corporativos heterogêneos. Nesta seção, obtenha uma visualização de alto nível da plataforma Java e seus componentes.

#### A linguagem Java

Como qualquer linguagem de programação, a linguagem Java tem sua própria estrutura, regras de sintaxe e paradigma de programação. O paradigma de programação da linguagem Java baseia-se no conceito de OOP, que os recursos da linguagem suportam.

A linguagem Java deriva da linguagem C, portanto suas regras de sintaxe assemelham-se às regras de C. Por exemplo, os blocos de códigos são modularizados em métodos e delimitados por chaves ({ e }) e variáveis são declaradas antes que sejam usadas.

Estruturalmente, a linguagem Java começa com pacotes. Um pacote é o mecanismo de namespace da linguagem Java. Dentro dos pacotes estão as classes e dentro das classes estão métodos, variáveis, constantes e mais. Neste tutorial você aprende sobre as partes da linguagem Java.

### O compilador Java

Quando você programa na plataforma Java, escreve seu código-fonte em arquivos.java e depois os compila.

O compilador verifica seu código nas regras de sintaxe da linguagem e depois grava bytecode em arquivos.class. Bytecode é um conjunto de instruções destinadas a executar em uma Java virtual machine (JVM). Ao incluir esse nível de abstração, o compilador Java difere-se de outros compiladores de linguagem, que escrevem instruções adequadas para o chipset de CPU no qual o programa é executado.

## A JVM

No tempo de execução, a JVM lê e interpreta arquivos.class e executa as instruções do programa na plataforma de hardware nativa para qual a JVM foi escrita. A JVM interpreta o bytecode como uma CPU interpretaria instruções de linguagem assembly. A diferença é que a JVM é uma parte do software escrita especificamente para uma determinada plataforma. A JVM é o núcleo do princípio "gravação única, execução em qualquer local" da linguagem Java. Seu código pode executar em qualquer chipset para o qual a implementação da JVM adequada está disponível. JVMs estão disponíveis para principais plataformas, como Linux e Windows, e subconjuntos de linguagem Java foram implementados nas JVMs para telefones celulares e chips hobbyist.

## O coletor de lixo

Em vez de forçá-lo a manter a alocação de memória (ou usar uma biblioteca de terceiros para isso), a plataforma Java fornece gerenciamento de memória fora do padrão. Quando seu aplicativo Java cria uma instância de objeto no tempo de execução, a JVM aloca automaticamente espaço de memória para esse objeto a partir de um conjunto de memória heap— reservado para uso de seu programa. O coletor de lixo Java é executado em segundo plano, mantendo o controle de quais objetos o aplicativo não necessita mais e recuperando memória deles. Essa abordagem para manipulação de memória é chamada de gerenciamento implícito de memória porque não exige a gravação de qualquer código de manipulação de memória. A coleta de lixo é um dos recursos essenciais para o desempenho da plataforma Java.

## O Java Development Kit

Ao fazer o download de um Java Development Kit (JDK), você obtém, — além do compilador e de outras ferramentas, — uma biblioteca de classe completa de utilitários de pré-construção que o ajuda a realizar tarefas de desenvolvimento de aplicativo mais comuns. A melhor forma de obter uma ideia do escopo dos pacotes e bibliotecas JDK é verificar a documentação da API JDK.

## O Java Runtime Environment

O Java Runtime Environment (JRE; também conhecido como o tempo de execução Java) inclui a JVM, bibliotecas de códigos e componentes necessários para executar programas que são escritos na linguagem Java. O JRE está disponível para diversas plataformas. É possível redistribuir livremente o JRE com seus aplicativos, de acordo com os termos da licença do JRE, para fornecer aos usuários do aplicativo uma plataforma na qual executar seu software. O JRE está incluído no JDK.

## Configurando seu ambiente de desenvolvimento Java

Nesta seção, você fará o download e instalará o JDK e a liberação atual do Eclipse IDE e configurará seu ambiente de desenvolvimento Eclipse.

Se você já tiver o JDK e Eclipse IDE instalados, poderá desejar pular para a seção "Introdução ao Eclipse" ou para aquela depois dela, "Conceitos de programação orientada a objeto."

## Seu ambiente de desenvolvimento

O JDK inclui um conjunto de ferramentas de linha de comandos para compilar e executar seu código Java, incluindo uma cópia completa do JRE. Embora seja possível usar essas ferramentas para desenvolver seus aplicativos, a maioria dos desenvolvedores apreciam a funcionalidade adicional, o gerenciamento de tarefas e a interface visual de um IDE.

Eclipse é um IDE de software livre popular para desenvolvimento Java. O Eclipse manipula tarefas básicas, como a compilação e depuração de códigos, portanto, você pode focar na escrita e teste de códigos. Além disso, é possível usar o Eclipse para organizar seus arquivos de código-fonte em projetos, compilar e testar esses projetos e armazenar arquivos de projetos em qualquer número de repositórios de origem. É necessário ter um JDK instalado para usar Eclipse para desenvolvimento Java. Se você fizer o download de um dos pacotes configuráveis Eclipse, ele já virá com o JDK.

## Instale o JDK

Siga estas etapas para fazer o download e instalar o JDK:

1. Navegue para Downloads do Java SE e clique na caixa Plataforma Java (JDK) para exibir a página de download da última versão do JDK.
2. Concorde com os termos da licença.
3. Em Java SE Development Kit, escolha o download que corresponda a seu sistema operacional e arquitetura de chip.

## Windows

1. Salve o arquivo em sua unidade de disco rígido quando solicitado.
2. Quando o download estiver concluído, execute o programa de instalação. Instale o JDK em sua unidade de disco rígido em um local fácil de lembrar, como C:\home\Java\jdk1.8.0\_60. É uma boa ideia codificar o número de atualização no nome do diretório de instalação escolhido.

## OS X

1. Quando o download estiver concluído, dê um clique duplo nele para montá-lo.
2. Execute o programa de instalação. Você não escolhe onde o JDK será instalado. É possível executar `/usr/libexec/java_home -1.8` para ver o local do JDK 8 em seu Mac. O caminho isdisplay é semelhante `/Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home`.

Consulte Instalação de JDK 8 e JRE 8 para obter mais informações.

Agora você possui um ambiente Java em seu computador. Depois, instalará o Eclipse IDE.

## Instale o Eclipse

Para fazer o download e instalar o Eclipse, siga estas etapas:

1. Navegue até Página de downloads do Eclipse IDE.
2. Clique em Eclipse IDE for Java Developers.
3. Em Links de download à direita, escolha sua plataforma (o site já pode ter encontrado seu tipo de sistema operacional).
4. Clique no espelho do qual deseja fazer download; depois salve o arquivo em sua unidade de disco rígido.
5. Extraia o conteúdo do arquivo.zip em um local em sua unidade de disco rígido do qual se lembre facilmente (como C:\home\eclipse no Windows ou ~/home/eclipse em Mac ou Linux).

## Configure o Eclipse

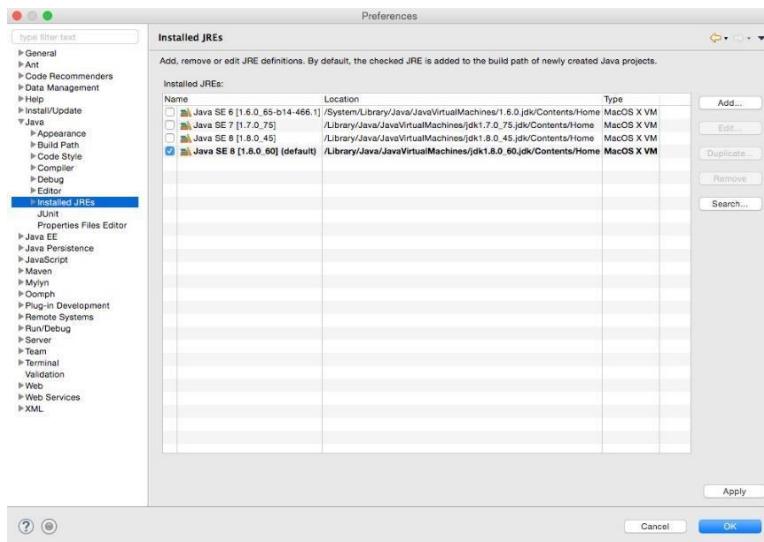
O Eclipse IDE opera sobre o JDK como um conceito de abstração útil, mas ainda precisa acessar o JDK e suas várias ferramentas. Antes de poder usar o Eclipse para escrever código Java, é necessário informar a ele onde o JDK está localizado.

Para configurar seu ambiente de desenvolvimento Eclipse:

1. Ative o Eclipse clicando duas vezes em eclipse.exe (ou no executável equivalente em sua plataforma).
2. O Ativador de área de trabalho é aberto, permitindo escolher uma pasta raiz para seus projetos Eclipse. Use uma pasta da qual possa se lembrar facilmente, como C:\home\workspace em Windows ou ~/home/workspace em Mac ou Linux.

3. Feche a janela Bem-vindo ao Eclipse.
4. Clique em Janela > Preferências > Java > JREs instalados. Figura 1 mostra esta seleção destacada na janela de configuração do Eclipse para o JRE.

Figura 1. Configurando o JDK que o Eclipse usa



Clique para ver a imagem maior

5. O Eclipse aponta para um JRE instalado. Você deve usar o JRE que transferiu por download com o JDK. Se o Eclipse não detectar automaticamente o JDK instalado, clique em Incluir... e, na próxima caixa de diálogo, clique em VM padrão e clique em Avançar.
6. Especifique o diretório inicial do JDK (como C:\home\jdk1.8.0\_60 em Windows) e clique em Concluir.
7. Confirme se o JDK que você deseja usar está selecionado e clique em OK.

O Eclipse está agora configurado e pronto para você criar projetos e compilar e executar código Java. A próxima seção o familiariza com Eclipse.

## Introdução ao Eclipse

O Eclipse é mais que um IDE; ele é um ecossistema de desenvolvimento completo. Esta seção é uma breve introdução prática para usar o Eclipse para desenvolvimento Java.

O ambiente de desenvolvimento Eclipse

O ambiente de desenvolvimento eclipse tem quatro principais componentes:

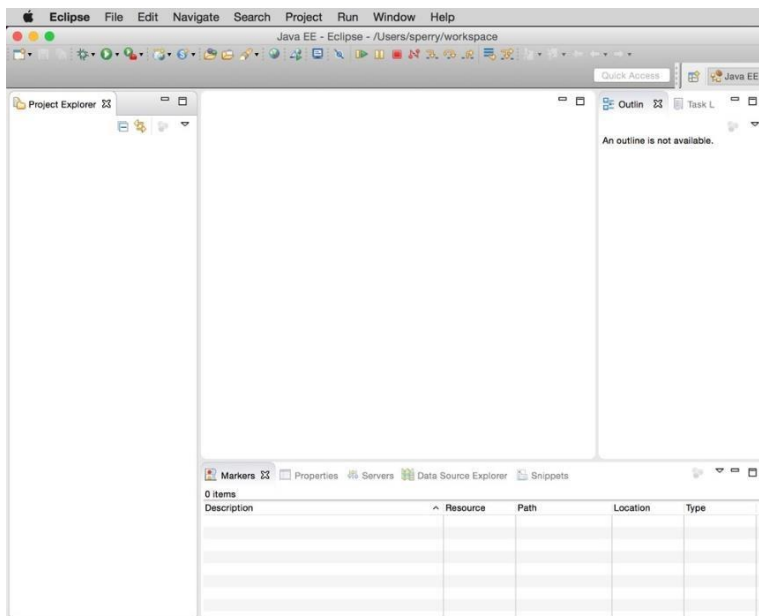
- Área de trabalho
- Projetos
- Perspectivas
- Visualizações

A unidade primária de organização no Eclipse é a área de trabalho. Uma área de trabalho contém todos os seus projetos. Uma perspectiva é uma forma de consulta a cada projeto (consequentemente o nome) e dentro de uma perspectiva há uma ou mais visualizações.

## A perspectiva Java

A Figura 2 mostra a perspectiva Java, que é a perspectiva padrão para Eclipse. Você deverá ver essa perspectiva ao iniciar o Eclipse.

Figura 2. Perspectiva Eclipse Java



Clique para ver a imagem maior

A perspectiva Java contém as ferramentas necessárias para começar a escrever aplicativos Java. Cada janela tabulada mostrada na Figura 2 é uma visualização da perspectiva Java. O Package Explorer e o Outline são duas visualizações especialmente úteis.

O ambiente Eclipse é altamente configurável. Cada visualização é adaptável, portanto, é possível movê-la na perspectiva Java e colocá-la onde desejar. Por enquanto, no entanto, aderiremos à perspectiva padrão e configuração de visualização.

## Crie um projeto

Siga estas etapas para criar um novo projeto Java:

1. Clique em Arquivo > Novo > Projeto Java... para iniciar o assistente para Novo projeto Java, mostrado da Figura 3.

Figura 3. Assistente de Novo projeto Java



2. Insira Tutorial como o nome do projeto e clique em Concluir.

3. Se desejar modificar as configurações padrão do projeto, clique em Avançar (recomendado apenas se você tiver experiência com o Eclipse IDE).

4. Clique em Concluir para aceitar a configuração do projeto e criá-lo.

Agora você criou um novo projeto Eclipse Java e uma pasta de origem. Seu ambiente de desenvolvimento está pronto para ação. No entanto, um entendimento do paradigma OOP — abrangido nas próximas duas seções deste tutorial — é essencial. Se você estiver familiarizado com os conceitos e princípios de OOP, pode desejar pular para a seção "Introdução à linguagem Java".

### Conceitos de programação orientada a objeto

A linguagem Java é (principalmente) orientada a objetos. Se você não usou uma linguagem orientada a objetos antes, os conceitos de OOP podem parecer estranhos à primeira vista. Esta seção é uma rápida introdução aos conceitos de linguagem OOP, usando programação estruturada como um ponto de contraste.

O que é um Objeto?

Linguagens de programação estruturadas, como C e COBOL, seguem um paradigma de programação diferente daquelas orientadas a objetos. O paradigma de programação estruturada é altamente orientado a dados: você possui estruturas de dados e instruções do programa atuam nesses dados. Linguagens orientadas a objetos, como a linguagem Java, combinam instruções de dados e programas em objetos.

Um objeto é uma entidade autocontida que contém atributos e comportamento, e nada mais. Em vez de ter uma estrutura de dados com campos (atributos) e transmite essa estrutura em toda a lógica do programa que atua nela (comportamento), em uma linguagem orientada a objetos, dados e lógica de programa são combinados. Essa combinação pode ocorrer em níveis completamente diferentes, de objetos com baixa granularidade, como Number, a objetos com alta granularidade, como um serviço FundsTransfer em um aplicativo financeiro amplo.

Objetos Pai e Filho

Um objeto pai é aquele que serve como base estrutural para derivação de objetos-filho mais complexos. Um objeto-filho assemelha-se ao seu pai, mas é mais especializado. Com o paradigma orientado a objetos, é possível reutilizar os atributos comuns e o comportamento do objeto pai, incluindo nesses objetos-filho atributos e comportamentos diferentes. (Você aprende mais sobre herança na próxima seção deste tutorial.)

Comunicação e coordenação de objetos

Os objetos se comunicam enviando mensagens (chamadas de métodos na linguagem Java). Além disso, em um aplicativo orientado a objetos, o código do programa coordena as atividades entre objetos para executar tarefas dentro do contexto do domínio de aplicativo específico.

Resumo de objeto

Um objeto bem escrito:

- Tem limites nítidos
- Executa um conjunto finito de atividades
- Conhece apenas sobre seus dados e quaisquer outros objetos necessários para realizar suas atividades

Em essência, um objeto é uma entidade discreta que possui apenas as dependências necessárias em outros objetos para executar suas tarefas.

Agora, você vê como um objeto se parece.

O objeto Person



Eu começo com um exemplo que baseia-se em um cenário de desenvolvimento de aplicativo comum: um indivíduo sendo representado por um objeto Person.

Voltando à definição de um objeto, você sabe que um objeto tem dois elementos primários: atributos e comportamento. Eu mostro como eles se aplicam a Person.

#### Atributos

Quais atributos uma pessoa pode ter? Alguns atributos comuns incluem:

- Name
- Age
- Height
- Weight
- Eye color
- Gender

Você provavelmente pensará em outros (e pode sempre incluir mais atributos posteriormente), mas esta lista é um bom começo.

#### Comportamento

Uma pessoa real pode fazer todo o tipo de ação, mas comportamentos de objetos geralmente estão relacionados a algum tipo de contexto de aplicativo. Em um contexto de aplicativo de negócios, por exemplo, você pode querer perguntar ao seu objeto Person : "Qual é sua idade?" Em resposta, Person informaria o valor de seu atributo Age.

Lógica mais complexa pode estar oculta dentro do objeto Person—, por exemplo calcular um Índice de massa corpórea (IMC) de uma pessoa para um aplicativo de saúde, — mas, por enquanto, suponha que Person tenha o comportamento de responder a estas questões:

- Qual é seu nome?
- Qual é sua idade?
- Qual é sua altura?
- Qual é seu peso?
- Qual é a cor de seus olhos?
- Qual é seu sexo?

#### Estado e sequência

Estado é um conceito importante em OOP. Um estado de objeto é representado a qualquer momento pelo valor de seus atributos.

No caso de Person, seu estado é definido por atributos como nome, idade, altura e peso. Se você quisesse apresentar uma lista de vários desses atributos, poderia fazer isso usando uma classe String, da qual eu falarei mais posteriormente no tutorial.

Usando os conceitos de estado e sequência juntos, é possível dizer para Person, "Diga-me quem você é fornecendo-me uma listagem (ou String) de seus atributos."

#### Princípios de OOP

Se você veio de um plano de fundo de programação estruturada, a proposição de valor de OOP pode ainda não estar muito clara. Afinal, os atributos de uma pessoa e qualquer lógica para recuperar (e converter) esses valores podem ser escritos em C ou COBOL. Esta seção esclarece os benefícios do paradigma de OOP explicando seus princípios de definição: encapsulamento, herança e polimorfismo.

### Encapsulamento

Lembre-se de que um objeto é, acima de tudo, discreto ou autocontido. Essa característica é o princípio do encapsulamento em funcionamento. Ocultação é outro termo que às vezes é usado para expressar a natureza autocontida e protegida de objetos.

Independentemente da terminologia, o que é importante é que o objeto mantém um limite entre seu estado e o comportamento e o mundo externo. Como objetos no mundo real, objetos usados na programação de computador possuem vários tipos de relacionamentos com diferentes categorias de objetos nos aplicativos que os usam.

Na plataforma Java, é possível usar modificadores de acesso (que eu apresento posteriormente no tutorial) para variar a natureza dos relacionamentos de objetos de público para privado. O acesso público é muito aberto, considerando que acesso privado significa que os atributos de objetos estão acessíveis apenas dentro do próprio objeto.

O limite público/privado impinge o princípio de encapsulamento orientado a objetos. Na plataforma Java, é possível variar a intensidade desse limite em uma base de objeto por objeto, dependendo de um sistema de confiança. O encapsulamento é um recurso poderoso da linguagem Java.

### Herança

Na programação estruturada, é comum copiar uma estrutura, nomeá-la e incluir ou modificar os atributos que torna a nova entidade (como um registro Account) diferente de sua fonte original. Com o tempo, essa abordagem gera uma grande questão de código duplicado, o que pode criar problemas de manutenção.

O OOP apresenta o conceito de herança, pelo qual classes especializadas — sem código adicional — podem "copiar" os atributos e o comportamento de classes de origem na qual elas se especializam. Se alguns desses atributos ou comportamentos precisarem mudar, você os substitui. O único código-fonte a ser mudado é o código necessário para criar classes especializadas. Como vimos na seção "Conceitos da programação orientada a objetos", o objeto de origem é chamado de pai e a nova especialização é chamada de filho.

### Herança em Funcionamento

Suponha que você esteja programando um aplicativo de recursos humanos e queira usar a classe Person como base (chamada de superclasse) para uma nova classe chamada Employee. Sendo filha de Person, Employee teria todos os atributos de uma classe Person, junto com aqueles adicionais, como:

- Número de identificação de contribuinte
- Número de matrícula
- Salário

A herança facilita a criação da nova classe Employee sem a necessidade de copiar todo o código Person manualmente.

Você verá muitos exemplos de herança na programação Java posteriormente no tutorial, especialmente na Parte 2.

### Polimorfismo

O polimorfismo é um conceito mais difícil de compreender do que o encapsulamento e a herança. Em essência, isso significa que objetos que pertencem à mesma ramificação de uma herança, quando

enviam a mesma mensagem (ou seja, quando efetuam a mesma ação), podem manifestar esse comportamento de forma diferente.

Para entender como o polimorfismo se aplica a um contexto de aplicativo de negócios, retorne ao exemplo de Person. Lembre-se de dizer a Person para formatar seus atributos em uma String? O polimorfismo torna possível que Person represente seus atributos em uma variedade de formas, dependendo do tipo que Person é.

O polimorfismo é um dos conceitos mais complexos que você encontrará no OOP na plataforma Java e está além do escopo de um tutorial introdutório.

### **Introdução à linguagem Java**

Seria impossível apresentar toda a sintaxe da linguagem Java em um único tutorial. O lembrete da Parte 1 foca nos fundamentos da linguagem, deixando você com conhecimento suficiente e prática para escrever programas simples. OOP trata exclusivamente de objetos, portanto, esta seção começa com dois tópicos especialmente relacionados a como a linguagem Java os manipula: palavras reservadas e a estrutura de um objeto Java.

#### **Palavras reservadas**

Como qualquer linguagem de programação, a linguagem Java designa determinadas palavras que o compilador reconhece como especiais. Por essa razão, você não pode usá-las para nomear suas construções Java. A lista de palavras reservadas é surpreendentemente curta:

- abstract
- assert
- boolean
- break
- byte
- case
- catch
- char
- class
- const
- continue
- default
- do
- double
- else
- enum
- extends
- final
- finally

- float
- para
- goto
- if
- implements
- import
- instanceof
- int
- interface
- long
- native
- new
- package
- private
- protected
- public
- return
- short
- static
- strictfp
- super
- switch
- synchronized
- this
- throw
- throws
- transient
- try
- void
- volatile
- while

Observe que true, false e null são tecnicamente palavras não reservadas. Embora sejam literais, eu as incluí nesta lista porque você não pode usá-las para nomear construções Java.

Uma vantagem da programação com um IDE é que ela pode usar a coloração de sintaxe para palavras reservadas, como eu mostro posteriormente neste tutorial.

### Estrutura de uma Classe Java

Uma classe é um blueprint para uma entidade discreta (objeto) que contém atributos e comportamentos. A classe define a estrutura básica do objeto e, no tempo de execução, seu aplicativo cria uma instância do objeto. Um objeto tem um limite e um estado nítidos e pode fazer ações quando corretamente solicitado. Cada linguagem orientada a objetos possui regras sobre como definir uma classe.

Na linguagem Java, classes são definidas conforme mostrado na Listagem 1:

Lista 1. Definição de classe

```
1 package packageName;
2 import ClassNameToImport; accessSpecifier class ClassName {
3     accessSpecifier dataType variableName [= initialValue];
4     accessSpecifier ClassName([argumentList]) {
5         constructorStatement(s)
6     }
7     accessSpecifier returnType methodName ([argumentList]) {
8         methodStatement(s)
9     }
10 // This is a comment
11 /* This is a comment too */
12 /* This is a
13 multiline
14 comment */
15 }
```

Listagem 1 contém vários tipos de construções, que eu diferenciei com a formatação de fonte. As construções mostradas em negrito (que você localizará na lista de palavras reservadas) são literais; em qualquer definição de objeto, elas devem ser exatamente o que estão na listagem. Os nomes fornecidos para outras construções descrevem os conceitos que eles representam. Eu expliquei todas as construções detalhadamente no restante desta seção.

Observação: Na Listagem 1 e em alguns outros exemplos de códigos nesta seção, colchetes indicam que as construções dentro deles não são requeridas. Os colchetes (diferente de { e }) não fazem parte da sintaxe Java.

#### Comentários no código

Observe que Listagem 1 também inclui algumas linhas de comentário:

```
1 // This is a comment
```

```
2  /* This is a comment too */  
3  /* This is a  
4  multiline  
5  comment */
```

Com informações apenas sobre cada linguagem de programação, os programadores podem incluir comentários para ajudar a documentar o código. A sintaxe Java permite comentários de única linha e multilinhas. Um comentário de única linha deve estar contido em uma linha, embora seja possível usar comentários de única linha adjacentes para formar um bloco. Um comentário de multilinhas começa com `/*`, deve terminar com `*/` e pode abranger qualquer número de linhas.

Você obtém mais informações sobre comentários quando chega à seção "Escrevendo bons códigos Java" deste tutorial.

### Empacotando Classes

Com a linguagem Java, é possível escolher os nomes de suas classes, como `Account`, `Person` ou `LizardMan`. Às vezes, é possível terminar usando o mesmo nome para expressar dois conceitos um pouco diferentes. Esta situação é chamada de colisão de nomes e ocorre com frequência. A linguagem Java usa pacotes para resolver esses conflitos.

Um pacote Java é um mecanismo que fornece um namespace— de uma área dentro da qual nomes são exclusivos, mas fora dela eles podem não ser. Para identificar uma construção exclusivamente, é necessário qualificá-la incluindo seu namespace.

Pacotes também fornecem a você uma boa forma de criar aplicativos mais complexos com unidades de funcionalidade discretas.

#### Definição de pacote

Para definir um pacote, você usa a palavra-chave `package` seguida de um nome de pacote legal, terminando com um ponto e vírgula. Geralmente os nomes de pacotes são separados por pontos e seguem este esquema padrão de fato :

```
1 package orgType.orgName.appName.compName;
```

Esta definição de pacote é dividida como a seguir:

- `orgType` é o tipo de organização, como `com`, `org` ou `net`.
- `orgName` é o nome do domínio da organização, como `makotojava`, `oracle` ou `ibm`.
- `appName` é o nome do aplicativo, abreviado.
- `compName` é o nome do componente.

A linguagem Java não força você a seguir esta convenção de pacote. De fato, não é necessário especificar um pacote, nesse caso, todas as suas classes deverão ter nomes exclusivos e residirão no pacote padrão. Como melhor prática, eu recomendo que você defina todas as suas classes Java em pacotes nomeados conforme descrito aqui. Você segue essa convenção em todo este tutorial.

#### Instruções de importação

Próxima à definição de classe (referindo-se novamente à Listagem 1) está a instrução de importação. Uma instrução de importação informa ao compilador Java onde localizar classes às quais você fez referência dentro de seu código. Qualquer classe não trivial usa outras classes para alguma funcionalidade e a instrução de importação é como você informa ao compilador Java sobre elas.

Uma instrução de importação geralmente assemelha-se ao seguinte:

```
1 import ClassNameToImport;
```

Você especifica a palavra-chave import, seguida pela classe que deseja importar, seguida por um ponto e vírgula. O nome da classe deve ser completo, o que significa que ele deve incluir seu pacote.

Para importar todas as classes dentro de um pacote, é possível inserir.\* após o nome do pacote. Por exemplo, essa instrução importa cada classe no pacote com.makotojava :

```
1 import com.makotojava.*;
```

A importação de um pacote integral pode tornar seu código menos legível, portanto, eu recomendo importar apenas as classes necessárias, usando seus nomes completos.

### Declaração de classe

Para definir um objeto na linguagem Java, você deve declarar uma classe. Novamente, pense em uma classe como um modelo para um objeto, como um cortador de cookie.

Listagem 1 inclui esta declaração de classe:

```
1 accessSpecifier class ClassName {
2     accessSpecifier dataType variableName [= initialValue];
3     accessSpecifier ClassName([argumentList]) {
4         constructorStatement(s)
5     }
6     accessSpecifier returnType methodName([argumentList]) {
7         methodStatement(s)
8     }
9 }
```

Um accessSpecifier da classe pode ter diversos valores, mas ele geralmente é public. Você logo consulta outros valores de accessSpecifier.

### Convenções de Nomenclatura de Classe

É possível nomear classes basicamente como você realmente deseja, mas a convenção é usar camel case: comece com uma letra maiúscula, altere para letras maiúsculas concatenada e torne todas as outras letras minúsculas. Nomes de classes devem conter apenas letras e números. A aderência a estas diretrizes garante que seu código seja mais acessível a outros desenvolvedores que estão seguindo as mesmas convenções.

Classes podem ter dois tipos de membros: variáveis e métodos.

#### Variáveis

Os valores de variáveis de uma classe específica distinguem cada instância daquela classe e definem seu estado. Esses valores geralmente são referidos como variáveis de instância. Uma variável possui:

- Um accessSpecifier
- Um dataType
- Um variableName
- Opcionalmente, um initialValue

Os valores possíveis de accessSpecifier são:

- public: qualquer objeto em qualquer pacote pode ver a variável. (Não use de forma alguma este valor; consulte a barra lateral Variáveis públicas.)
- protected: qualquer objeto definido no mesmo pacote ou uma subclasse (definida em qualquer pacote) pode ver a variável.
- Nenhum especificador (também chamado de acesso amigável ou pacote privado ): apenas objetos cujas classes são definidas no mesmo pacote podem ver a variável.
- private: apenas a classe contendo a variável pode vê-la.

Um dataType de variável depende do que a variável é, — ela pode ser um tipo primitivo ou outro tipo de classe (novamente, mais sobre isso posteriormente).

O variableName está ativo para você, mas, por convenção, nomes de variáveis usam a convenção de camel case que eu descrevi em "Convenções de nomenclatura de classe," exceto pelo fato de começarem com uma letra minúscula. (Este estilo é às vezes chamado de Camel Case iniciado por letra minúscula.)

Não se preocupe com o initialValue por enquanto; apenas saiba que é possível inicializar uma variável de instância ao declará-la. (Caso contrário, o compilador gerará um padrão para você que será configurado quando a classe for instanciada.)

Exemplo: definição de classe para Person

Antes de falarmos de métodos, veja um exemplo que resume o que você aprendeu até aqui. Listagem 2 é uma definição de classe para Person.

Lista 2. Definição básica de classe para Person

```
1 package com.makotojava.intro;
2
3 public class Person {
4     private String name;
5     private int age;
6     private int height;
7     private int weight;
8     private String eyeColor;
9     private String gender;
10 }
```

A definição básica de classe para Person não é útil neste ponto, pois define apenas seus atributos (e os atributos privados que estão nela).

Para que seja mais interessante, a classe Person precisa de comportamento — e isso significa métodos.

### Métodos

Os métodos de uma classe definem seu comportamento.



Métodos se enquadram em duas categorias principais: construtores ou todos os outros métodos, dos quais existem muitos tipos. Um método construtor é usado apenas para criar uma instância de uma classe. Outros tipos de métodos podem ser usados virtualmente para qualquer comportamento do aplicativo.

A definição de classe na Listagem 1 mostra a forma de definir a estrutura de um método, que inclui elementos como:

- accessSpecifier
- returnType
- methodName
- argumentList

A combinação desses elementos estruturais em uma definição de método é chamada de assinatura de método.

Depois, você consulta mais detalhadamente os dois tipos de métodos, começando com construtores.

### Métodos construtores

Você usa construtores para especificar como instanciar uma classe. Listagem 1 mostra a sintaxe de declaração do construtor de forma abstrata; novamente:

```
1 accessSpecifier ClassName([argumentList]) {  
2     constructorStatement(s)  
3 }
```

Um accessSpecifier do construtor é igual ao das variáveis. O nome do construtor deve corresponder ao nome da classe. Portanto, se você chamar sua classe de Person, o nome do construtor também deverá ser Person.

Para qualquer construtor que não seja o construtor padrão, você transmite argumentList, que é um ou mais dos seguintes:

```
1 argumentType argumentName
```

Argumentos em um argumentList são separados por vírgulas e nenhum dos dois argumentos pode ter o mesmo nome. argumentType é um tipo primitivo ou outro tipo de classe (igual com tipos de variáveis).

### Definição de Classe com um Construtor

Agora, você vê o que acontece quando inclui a capacidade para criar um objeto Person de duas formas: usando um construtor no-arg e inicializando uma lista parcial de atributos.

Listagem 3 mostra como criar construtores e também como usar argumentList:

Lista 3. Person : definição de classe com um construtor

```
1 package com.makotojava.intro;  
2 public class Person {  
3     private String name;  
4     private int age;  
5     private int height;
```

```
6     private int weight;
7     private String eyeColor;
8
9     private String gender;
10    public Person() {
11        // Nothing to do...
12    }
13
14    public Person(String name, int age, int height, int weight String eyeColor, String gender) {
15        this.name = name;
16        this.age = age;
17        this.height = height;
18        this.weight = weight;
19        this.eyeColor = eyeColor;
20        this.gender = gender;
21    }
22 }
```

Observe o uso da palavra-chave `this` ao fazer designações de variáveis na Listagem 3. A palavra-chave `this` é uma abreviação Java para "this object," e você deve usá-la ao fazer referência para duas variáveis com o mesmo nome. Neste caso, `age` é um parâmetro de construtor e uma variável de classe, portanto, a palavra-chave `this` ajuda o compilador a desambiguar a referência.

O objeto `Person` está ficando mais interessante, mas precisa de mais comportamento. E, para isso, você precisa de mais métodos.

#### Outros métodos

Um construtor é um tipo particular de método com uma função específica. Similarmente, muitos outros tipos de métodos executam funções específicas em programas Java. A exploração de outros tipos de métodos começa nesta seção e continua pelo tutorial.

De volta à Listagem 1, eu mostrei como declarar um método:

```
1  accessSpecifier returnType methodName ([argumentList]) {
2  methodStatement(s)
3  }
```

Outros métodos assemelham-se muito aos construtores, com algumas exceções. Primeiro, é possível nomear outros métodos como deseja (embora, claro, determinadas regras se apliquem). Eu recomendo as seguintes convenções:

- Comece com uma letra minúscula.
- Evite números, a menos que eles sejam absolutamente necessários.

- Use apenas caracteres alfabéticos.

Segundo, diferente dos construtores, outros métodos possuem um tipo de retorno opcional.

Outros métodos de Person

Munido destas informações básicas, é possível ver na Listagem 4 o que ocorre quando você inclui alguns métodos a mais em Person. (Para resumir, eu omiti os construtores.)

Lista 4. Person com alguns novos métodos

```
1 package com.makotojava.intro;
2
3 public class Person {
4     private String name;
5     private int age;
6     private int height;
7     private int weight;
8     private String eyeColor;
9     private String gender;
10
11     public String getName() { return name; }
12     public void setName(String value) { name = value; }
13     // Other getter/setter combinations...
14 }
```

Observe o comentário na Listagem 4 sobre "combinações de getter/setter." Você trabalhará mais com getters e setters posteriormente no tutorial. Por enquanto, tudo o que você precisa saber é que um getter é um método para recuperar o valor de um atributo e um setter é um método para modificar esse valor. A Listagem 4 mostra apenas uma combinação de getter/setter (para o atributo Name), mas é possível definir mais de uma forma semelhante.

Observe na Listagem 4 que se um método não retornar um valor, você deverá informar ao compilador especificando o tipo de retorno void em sua assinatura.

Métodos estáticos e de instância

Geralmente, dois tipos de métodos (sem construtor) são usados: métodos de instância e métodos estáticos. Métodos de instância dependem do estado de uma instância de objeto específica para seus comportamentos. Métodos estáticos também são às vezes chamados de métodos de classes, pois seus comportamentos não dependem de qualquer estado do objeto. O comportamento de um método estático ocorre no nível de classe.

Métodos estáticos são amplamente usados para utilidade; você pode pensar neles como sendo métodos globais (à la C) enquanto mantém o código para o método com a classe que o define.

Por exemplo, neste tutorial, você usa a classe JDK Logger para informações de saída para o console. Para criar uma instância da classe Logger, você não instancia uma classe Logger; ao contrário, você chama um método estático chamado getLogger().

A sintaxe para chamar um método estático em uma classe é diferente da sintaxe usada para chamar um método em um objeto. Você também usa o nome da classe que contém o método estático, conforme mostrado nesta chamada:

```
1 Logger l = Logger.getLogger("NewLogger");
```

Neste exemplo, Logger é o nome da classe e getLogger(...) é o nome do método. Assim, para chamar um método estático, não é necessário uma instância de objeto, apenas o nome da classe.

### Sua primeira classe Java

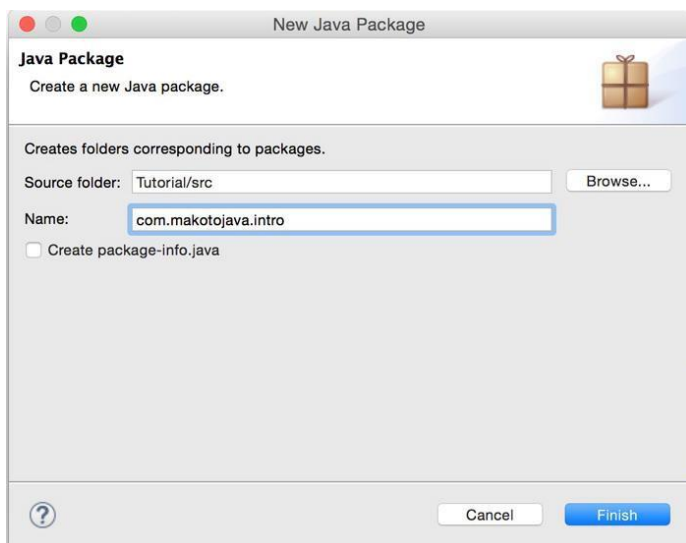
É hora de compilar tudo o que você aprender nas seções anteriores e começar a escrever algum código. Esta seção o orienta pela declaração de uma classe e inclui variáveis e métodos usando o Eclipse Package Explorer. Você aprende a usar a classe Logger para observar o comportamento de seu aplicativo e também aprende a usar um método main() como uma rotina de teste.

#### Criando um pacote

Se você ainda não estiver neste ponto, vá para a visualização do Package Explorer (na perspectiva Java) no Eclipse usando Janela > Perspectiva > Abrir perspectiva. Você obterá a configuração para criar sua primeira classe Java. A primeira etapa é criar um local para a classe residir. Pacotes são construções de namespace e também são mapeados de forma conveniente diretamente para a estrutura de diretórios do sistema de arquivos.

Em vez de usar o pacote padrão (quase sempre uma má ideia), você cria um especificamente para o código que está escrevendo. Clique em Arquivo > Novo > Pacote para iniciar o assistente Java Package, mostrado na Figura 4:

Figura 4. O assistente Eclipse Java Package



Digite com.makotojava.intro na caixa de texto Nome e clique em Concluir. Você verá o novo pacote criado no Package Explorer.

#### Declarando a classe

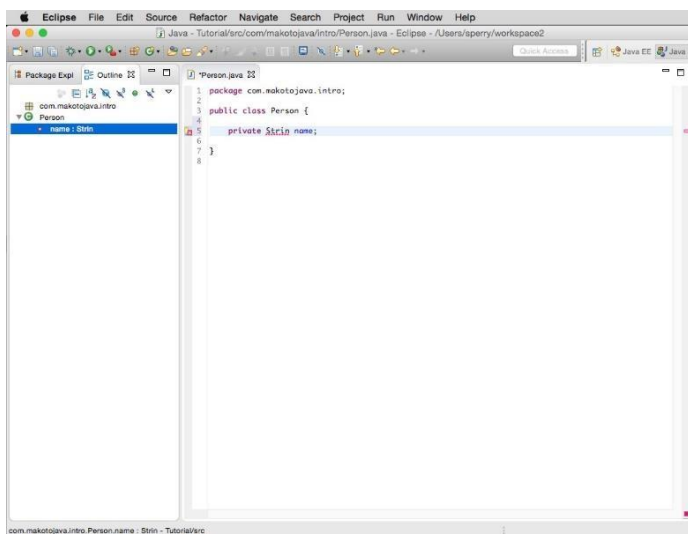
É possível criar uma classe a partir do Package Explorer de mais de uma forma, mas a forma mais fácil é clicar com o botão direito no pacote que acabou de criar e escolher Nova > Classe.... A caixa de diálogo Nova classe é aberta.

Na caixa de texto Nome, digite Person e clique em Concluir.

A nova classe é exibida em sua janela de edição. Eu recomendo fechar algumas das visualizações (Problemas, Javadoc e outras) abertas por padrão na perspectiva Java na primeira vez que abri-la para facilitar a visualização de seu código-fonte. (O Eclipse lembra que você não deseja ver aquelas

visualizações na próxima vez que abrir o Eclipse e vai para a perspectiva Java.) Figura 5 mostra uma área de trabalho com as visualizações essenciais abertas.

Figura 5. Uma área de trabalho bem ordenada



Clique para ver a imagem maior

O Eclipse gera uma classe shell para você e inclui a instrução package no início. Você só precisa implementar a classe agora. É possível configurar como o Eclipse gerará novas classes usando Janela > Preferências > Java > Estilo de código > Modelos de código. Para facilitar, acompanhe a geração de códigos simples de instalar do Eclipse.

Na Figura 5, observe o asterisco (\*) próximo ao novo nome de arquivo de código-fonte, indicando que eu fiz uma modificação. E observe que o código não foi salvo. Depois, observe que eu cometi um erro ao declarar o atributo Name : eu declarei o tipo de Name como se fosse Strin. O compilador não pôde localizar uma referência para esse tipo de classe e o sinalizou como um erro de compilação (a linha ondulada vermelha abaixo de Strin). Claro, eu posso corrigir meu erro incluindo um g ao final de Strin. Esta é uma pequena demonstração do poder do uso de um IDE em vez de ferramentas de linha de comandos para desenvolvimento de software. Continue e corrija o erro mudando o tipo para String.

Incluindo variáveis de classe

Na Listagem 3, você começa a implementar a classe Person, mas eu não expliquei muito da sintaxe. Agora, eu formalmente defino como incluir variáveis de classe.

Lembre-se de que uma variável possui um accessSpecifier, um dataType, um variableName e, opcionalmente, um initialValue. Anteriormente, você viu brevemente como definir accessSpecifier e variableName. Agora, verá o dataType que uma variável pode ter.

Um dataType pode ser um tipo primitivo ou uma referência para outro objeto. Por exemplo, observe que Age é um int (um tipo primitivo) e que Name é um String (um objeto). O JDK é fornecido cheio de classes úteis como java.lang.String e aquelas no pacote java.lang não precisam ser importadas (uma pequena cortesia do compilador Java). Mas se o dataType for uma classe JDK como String ou uma classe definida pelo usuário, a sintaxe será essencialmente a mesma.

A Tabela 1 mostra os oito tipos de dados primitivos que você provavelmente verá regularmente, incluindo os valores padrão que essas primitivas obtêm se você não inicializar explicitamente um valor de variável do membro.

Tabela 1. Tipos de dados primitivos

Tipo	Tamanho	Valor padrão	Faixa de valores
boolean	n/a	false	true ou false
byte	8 bits	0	-128 a 127
char	16 bits	(não designado)	\u0000' \u0000' a \uffff' ou 0 a 65535
short	16 bits	0	-32768 a 32767
int	32 bits	0	-2147483648 a 2147483647
long	64 bits	0	-9223372036854775808 a 9223372036854775807
float	32 bits	0,0	1.17549435e-38 a 3.4028235e+38
double	64 bits	0,0	4.9e-324 a 1.7976931348623157e+308

### Criação de log integrada

Antes de continuar na codificação, você precisa saber como seus programas informam o que estão fazendo.

A plataforma Java inclui o pacote `java.util.logging`, um mecanismo de criação de log integrado para reunir informações do programa de uma forma legível. Criadores de log são entidades nomeadas que você cria usando uma chamada de método estática para a classe `Logger` :

```

1 import java.util.logging.Logger;
2 //...
3 Logger l = Logger.getLogger(getClass().getName());
    
```

Ao chamar o método `getLogger()`, você transmite a ele um `String`. Por enquanto, apenas tenha o hábito de transmitir o nome da classe na qual o código que está escrevendo está localizado. De qualquer método regular (ou seja, não estático), o código anterior sempre faz referência ao nome da classe e o transmite para `Logger`.

Se você estiver fazendo uma chamada de Logger dentro de um método estático, faça referência ao nome da classe na qual está:

```
1 Logger l = Logger.getLogger(Person.class.getName());
```

Neste exemplo, o código no qual você está é a classe Person, portanto, você faz referência a um literal especial chamado class que recupera o objeto Class (mais sobre isso posteriormente) e obtém seu atributo Name.

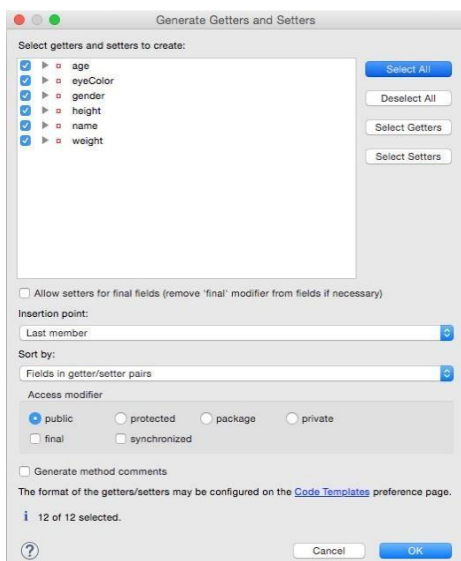
A seção "Escrevendo um bom código Java" deste tutorial inclui uma dica sobre como não efetuar a criação de log.

Antes de chegarmos ao corpo do teste, primeiro vá ao editor de código-fonte do Eclipse para Person e inclua esse código depois de public class Person { a partir da Listagem 3, de modo que fique semelhante ao seguinte:

```
1 package com.makotojava.intro;
2
3 public class Person {
4     private String name;
5     private int age;
6     private int height;
7     private int weight;
8     private String eyeColor;
9     private String gender;
10 }
```

O Eclipse tem um gerador de código útil para gerar getters e setters (entre outras coisas). Para experimentar o gerador de código, coloque o cursor do mouse na definição de classe Person (ou seja, na palavra Person na definição de classe) e clique em Origem > Gerar getters e setters.... Quando a caixa de diálogo abrir, clique em Selecionar tudo, conforme mostrado na Figura 6.

Figura 6. Eclipse gerando getters e setters



Para o ponto de inserção, escolha Último membro e clique em OK.

Agora, inclua um construtor em Person digitando o código a partir da Listagem 5 em sua janela de origem, logo abaixo da parte superior da definição de classe (a linha imediatamente abaixo de public class Person ()).

#### Lista 5. Construtor Person

```
1 public Person(String name, int age, int height, int weight, String eyeColor, String gender) {
2     this.name = name;
3     this.age = age;
4     this.height = height;
5     this.weight = weight;
6     this.eyeColor = eyeColor;
7     this.gender = gender;
8 }
```

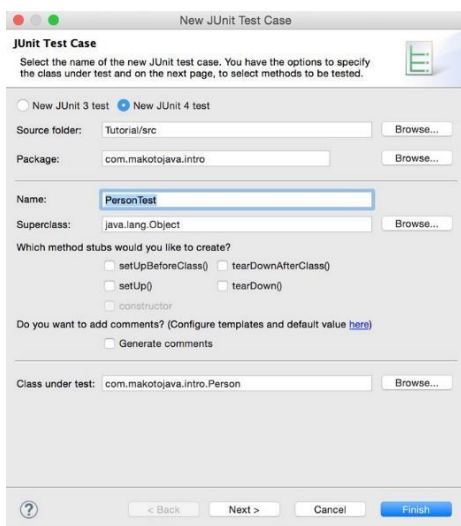
Certifique-se de não ter nenhuma linha ondulada indicando erros de compilação.

#### Gere um caso de teste JUnit

Agora você gera um caso de teste JUnit no qual instanciar um Person, usando o construtor na Listagem 5, e depois imprime o estado do objeto para o console. Nesse sentido, o "teste" certifica-se de que a ordem dos atributos na chamada do construtor esteja correta (ou seja, que eles estejam configurados para os atributos corretos).

No Package Explorer, clique com o botão direito em sua classe Person e depois clique em Novo > Caso de teste JUnit. A primeira página do assistente Novo caso de teste JUnit é aberta, conforme mostrado na Figura 7.

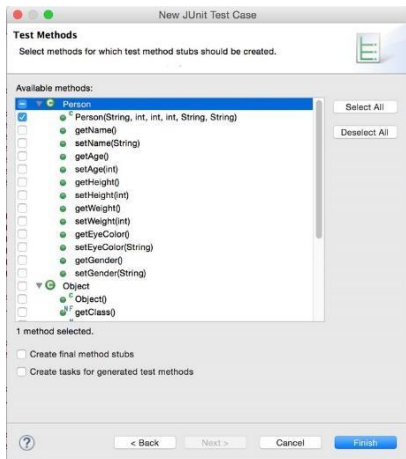
Figura 7. Criando um caso de teste JUnit



Aceite os padrões clicando em Avançar. Você vê a caixa de diálogo Métodos de teste, mostrada na Figura 8.

Figura 8. Selecione métodos para o assistente para gerar casos de teste





Nesta caixa de diálogo, selecione o método ou os métodos para os quais deseja que o assistente faça testes. Neste caso, selecione apenas o construtor, conforme mostrado na Figura 8. Clique em Concluir e o Eclipse gerará o caso de teste JUnit.

Depois, abra PersonTest, vá para o método testPerson() e deixe-o semelhante à Listagem 6.

Lista 6. O método testPerson()

```

1  @Test
2  public void testPerson() {
3      Person p = new Person("Joe Q Author", 42, 173, 82, "Brown", "MALE");
4      Logger l = Logger.getLogger(Person.class.getName());
5      l.info("Name: " + p.getName());
6      l.info("Age:" + p.getAge());
7      l.info("Height (cm):" + p.getHeight());
8      l.info("Weight (kg):" + p.getWeight());
9      l.info("Eye Color:" + p.getEyeColor());
10     l.info("Gender:" + p.getGender());
11     assertEquals("Joe Q Author", p.getName());
12     assertEquals(42, p.getAge());
13     assertEquals(173, p.getHeight());
14     assertEquals(82, p.getWeight());
15     assertEquals("Brown", p.getEyeColor());
16     assertEquals("MALE", p.getGender());
17 }

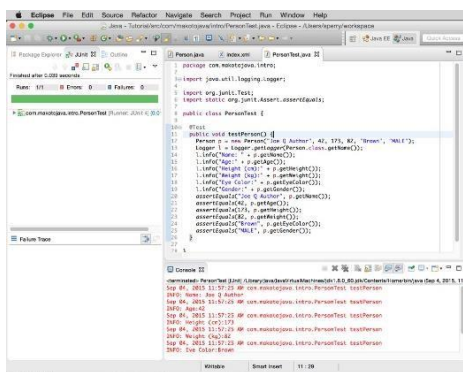
```

Não se preocupe com a classe Logger por enquanto. Apenas insira o código como o vê na Listagem 6. Agora você está pronto para executar seu primeiro programa Java (e caso de teste JUnit).

## Executando código em Eclipse

Para executar um aplicativo Java de dentro do Eclipse, selecione a classe que deseja executar, depois clique no ícone Executar (que está em verde e tem uma pequena seta triangular apontando para direita). O Eclipse é suficientemente inteligente para reconhecer que a classe que você deseja executar é um caso de teste JUnit e, portanto, ativa JUnit. Agora, sente-se e observe. A Figura 9 mostra o que acontece.

Figura 9. Veja a execução de Person



Clique para ver a imagem maior

A visualização Console abre automaticamente e mostra a saída Logger. A visualização JUnit também é aberta para mostrar os resultados do teste.

## Incluindo comportamento em uma classe Java

Person parece muito bom até agora, mas algum comportamento adicional pode ser usado para torná-lo mais interessante. Criando meios de comportamento incluindo métodos. Esta seção aparece mais detalhadamente em métodos acessadores—, os getters e setters já vistos em ação.

### Métodos acessadores

Para conter dados de uma classe de outros objetos, você declara suas variáveis como sendo private e depois fornece métodos acessadores. Como visto, um getter é um método acessador para recuperar o valor de um atributo; um setter é um método acessador para modificar esse valor. A nomenclatura dos acessadores segue uma convenção estrita conhecida como o padrão JavaBeans, pelo qual qualquer atributo Foo tem um getter chamado getFoo() e um setter chamado setFoo().

O padrão JavaBeans é tão comum que o suporte para ele é desenvolvido em Eclipse IDE. Você já o viu em ação — quando gerou getters e setters para Person na seção anterior.

Os acessadores seguem estas diretrizes:

- O próprio atributo é sempre declarado com acesso private.
- O especificador de acesso para getters e setters é public.
- Getters não obtêm nenhum parâmetro e retornam um valor cujo tipo é igual ao do atributo que eles acessam.
- As configurações apenas obtêm um parâmetro, do tipo de atributo, e não retornam um valor.

### Declarando acessadores

De longe a forma mais fácil de declarar acessadores é permitir que o Eclipse faça isso para você, como mostrado na Figura 6. Mas você deve também saber como codificar manualmente um par de getter e setter. Suponha que você tenha um atributo, Foo, cujo tipo seja java.lang.String. Uma declaração completa para ele (seguindo as diretrizes do acessador) seria:

```
1 private String foo;
2 public String getFoo() {
3     return foo;
4 }
5 public void setFoo(String value) {
6     foo = value;
7 }
```

É possível observar que o valor de parâmetro transmitido ao setter foi nomeado de forma diferente daquele gerado pelo Eclipse. A nomenclatura segue sua própria convenção, que eu recomendo para outros desenvolvedores. Nos raros casos que eu codifiquei manualmente um setter, sempre uso o nome value como o valor de parâmetro para o setter. Esse destaque me lembra que eu codifiquei manualmente o setter. Como geralmente eu permito que o Eclipse gere getters e setters para mim, quando isso não ocorre, é por uma boa razão. Usar value como o valor de parâmetro de setter lembra-me de que esse setter é especial. (Comentários de códigos também fazem isso.)

Chamando métodos

Chamar métodos é fácil. Você viu na Listagem 6 como chamar os diversos getters de Person para retornar seus valores. Agora, eu formalizo os mecanismos de chamadas de métodos.

Chamada de método com e sem parâmetros

Para chamar um método em um objeto, é necessário uma referência a esse objeto. A sintaxe de chamada de método abrange a referência de objeto, um ponto literal, o nome do método e todos os parâmetros que precisam ser transmitidos:

```
1 objectReference.someMethod();
2 objectReference.someOtherMethod(parameter);
```

A seguir, uma chamada de método sem parâmetros:

```
1 Person p = /*obtain somehow */;
2 p.getName();
```

E, aqui, uma chamada de método com parâmetros (acessando o atributo Name de Person):

```
1 Person p = new Person("Joe Q Author", 42, 173, 82, "Brown", "MALE");
```

Lembre-se de que construtores são métodos também. E é possível separar os parâmetros com espaços e novas linhas. O compilador Java não se importa. Essas duas próximas chamadas de métodos são idênticas:

```
1 new Person("Joe Q Author", 42, 173, 82, "Brown", "MALE");
1 new Person("Joe Q Author", // Name
2     42, // Age
3     173, // Height in cm
4     82, // Weight in kg
5     "Brown", // Eye Color
```

```
6 "MALE");// Gender
```

Observe como os comentários na segunda chamada do construtor fornece capacidade de leitura para o próximo desenvolvedor. Rapidamente essa pessoa pode dizer ao que cada parâmetro se destina.

Chamada de método aninhada

Chamadas de métodos também podem ser aninhadas:

```
1 Logger l = Logger.getLogger(Person.class.getName());
2 l.info("Name: " + p.getName());
```

Aqui você está transmitindo o valor de retorno `Person.class.getName()` para o método `getLogger()`. Lembre-se de que a chamada de método `getLogger()` é uma chamada de método estática, portanto sua sintaxe difere um pouco. (Não é necessário uma referência de `Logger` para fazer a chamada; no lugar, você usa o nome da classe do lado esquerdo da chamada.)

Isso é realmente tudo o que há para chamada de método.

## Strings e Operadores

O tutorial até agora apresentou várias variáveis do tipo `String`, mas sem muita explicação. Você aprendeu mais sobre variáveis `string` nesta seção e também descobriu quando e como usar operadores.

### Strings

Lidar com variáveis `string` em C é muito trabalhoso, pois elas são matrizes com terminação nula de caracteres de 8 bits que você deve manipular. Na linguagem Java, variáveis `string` são objetos de primeira classe do tipo `String`, com métodos que ajudam a manipulá-los. (O código Java mais próximo à linguagem C em relação a variáveis `string` é o tipo de dado primitivo `char`, que pode manter um único caractere Unicode, como a.)

Você já viu como instanciar um objeto `String` e configurar seu valor (voltando à Listagem 4), mas há várias outras formas de fazer isso. Há várias formas de criar uma instância `String` com um valor de `hello`:

```
1 String greeting = "hello";
1 greeting = new String("hello");
```

Como `Strings` são objetos de primeira classe na linguagem Java, você pode usar `new` para instanciá-los. A configuração de uma variável do tipo `String` tem o mesmo resultado, pois a linguagem Java cria um objeto `String` para manter o literal, depois designa esse objeto à variável de instância.

### Concatenando Strings

É possível efetuar várias ações com `String` e a classe tem muitos métodos úteis. Mesmo sem usar um método, você já fez algo interessante com dois `Strings` concatenando-os ou combinando-os:

```
1 l.info("Name: " + p.getName());
```

O sinal de soma (+) é uma abreviação para concatenar `Strings` na linguagem Java. (Pode haver penalidade no desempenho ao fazer esse tipo de concatenação dentro de um `loop`, mas por enquanto, não é necessário se preocupar com isso.)

### Exemplo de Concatenação

Agora, você pode tentar concatenar `Strings` dentro de `Person`. Neste ponto, você possui uma variável de instância `name`, mas seria adequado ter um `firstName` e `lastName`. É possível então concatená-los quando outro objeto solicitar o nome completo de `Person`.

A primeira coisa a fazer é incluir as novas variáveis de instância (no mesmo local do código-fonte no qual name está atualmente definido):

```
1 //private String name;  
2 private String firstName;  
3 private String lastName;
```

Não é mais necessário ter name ; ele foi substituído por firstName e lastName.

Encadeando chamadas de métodos

Agora, é possível gerar getters e setters para firstName e lastName (conforme mostrado na Figura 6), remover o método setName() e mudar getName() para se assemelhar ao seguinte:

```
1 public String getName() {  
2     return firstName.concat(" ").concat(lastName);  
3 }
```

Este código ilustra o encadeamento de chamadas de métodos. O encadeamento é uma técnica comumente usada com objetos imutáveis como String, na qual uma modificação para um objeto imutável sempre retorna a modificação (mas não muda o original). Você opera então com o valor retornado, mudado.

Operadores

Como você pode esperar, a linguagem Java pode ser aritmética e você já viu como designar variáveis. Agora, daremos uma rápida olhada em alguns operadores de linguagem Java que você precisará à medida que suas habilidades melhoram. A linguagem Java usa dois tipos de operadores:

- Unário: apenas um operando é necessário.
- Binário: dois operandos são necessários.

Os operadores aritméticos da linguagem Java estão resumidos na Tabela 2.

Tabela 2. Operadores aritméticos da linguagem Java

Operador	Uso	Descrição
+	a + b	Inclui a e b
+	+a	Promove a para int se ele for um byte, short ou char
-	a - b	Subtrai b de a
-	-a	Nega aritmeticamente a

Operador	Uso	Descrição
*	a * b	Multiplica a e b
/	a / b	Divide a por b
%	a % b	Retorna o restante da divisão de a por b (o operador de módulo)
++	a++	Incrementa a por 1; calcula o valor de a antes de incrementar
++	++a	Incrementa a por 1; calcula o valor de a depois de incrementar
--	a--	Decrementa a por 1; calcula o valor de a antes do decremento de
--	--a	Decrementa a por 1; calcula o valor de a após decremento de
+=	a += b	Abreviação de a = a + b
-=	a -= b	Abreviação de a = a - b
*=	a *= b	Abreviação de a = a * b
%=	a %= b	Abreviação de a = a % b

#### Operadores adicionais

Além dos operadores na Tabela 2, você viu diversos outros símbolos que são chamados de operadores na linguagem Java, incluindo:

- Ponto (.), que qualifica nomes de pacotes e chama métodos
- Parênteses (), que delimita uma lista de parâmetros separados por vírgula para um método

- new, que (quando seguido por um nome de construtor) instancia um objeto

A sintaxe de linguagem Java também inclui diversos operadores que são usados especificamente para programação condicional — ou seja, programas que respondem de forma diferente com base na entrada diferente. Você vê sobre eles na próxima seção.

### Operadores condicionais e instruções de controle

Nesta seção, você aprende sobre as diversas instruções e operadores que pode usar para informar aos programas Java como deseja que eles atuem com base em entrada diferente.

#### Operadores relacionais e condicionais

A linguagem Java fornece operadores e instruções de controle que podem ser usados para a tomada de decisões em seu código. Mais frequentemente, uma decisão no código inicia com uma expressão booleana (ou seja, uma que seja avaliada como true ou false). Essas expressões usam operadores relacionais, que comparam um operando ou uma expressão a outra, e operadores condicionais.

Tabela 3 lista os operadores relacionais e condicionais da linguagem Java.

Tablela 3. Operadores relacionais e condicionais

Operador	Uso	Retorna true se...
>	a > b	a for maior que b
>=	a >= b	a é maior que ou igual a b
<	a < b	a é menor que b
<=	a <= b	a é menor que ou igual a b
==	a == b	a é igual a b
!=	a != b	a não é igual a b

Operador	Uso	Retorna true se...
&&	a && b	a e b são true, condicionalmente avalia b (se a for false, b não será avaliado)
	a    b	a ou b é true; condicionalmente avalia b (se a for true, b não será avaliado)
!	!a	a é false
&	a & b	a e b são true; sempre avalia b
	a   b	a ou b é true; sempre avalia b
^	a ^ b	a e b são diferentes

#### A instrução if

Agora que você tem alguns operadores, é hora de usá-los. Este código mostra o que ocorre quando você inclui alguma lógica para o acessador getHeight() do objeto Person :

```

1 public int getHeight() {
2     int ret = height;
3     // If locale of the machine this code is running on is U.S.,
4     if (Locale.getDefault().equals(Locale.US))
5         ret /= 2.54; // convert from cm to inches
6     return ret;
7 }
```

Se o código de idioma atual for Estados Unidos (em que o sistema de métrica não é usado), pode fazer sentido converter o valor interno de height (em centímetros) para polegadas. Este exemplo (um pouco controverso) ilustra o uso da instrução if, que avalia uma expressão booleana entre parênteses. Se essa expressão for avaliada como true, ela executa a próxima instrução.

Neste caso, você só precisa executar uma instrução se o Locale da máquina na qual o código está executando for Locale.US. Se for necessário executar mais de uma instrução, será possível usar



chaves para formar uma instrução composta. Uma instrução composta agrupa muitas instruções em um — e instruções compostas também podem conter outras instruções compostas.

### Escopo da variável

Cada variável em um aplicativo Java tem escopo, ou namespace localizado, que você pode acessá-lo por nome dentro do código. Fora desse espaço, a variável está fora do escopo e você obtém um erro de compilação se tenta acessá-la. Níveis de escopo na linguagem Java são definidos por onde uma variável é declarada, conforme mostrado na Listagem 7.

#### Lista 7. Escopo da variável

```
1 public class SomeClass {
2     private String someClassVariable;
3     public void someMethod(String someParameter) {
4         String someLocalVariable = "Hello";
5
6         if (true) {
7             String someOtherLocalVariable = "Howdy";
8         }
9         someClassVariable = someParameter; // legal
10        someLocalVariable = someClassVariable; // also legal
11        someOtherLocalVariable = someLocalVariable; // Variable out of scope!
12    }
13    public void someOtherMethod() {
14        someLocalVariable = "Hello there"; // That variable is out of scope!
15    }
16 }
```

Dentro de `SomeClass`, `someClassVariable` é acessível por todos os métodos de instância (ou seja, não estáticos). Dentro de `someMethod`, `someParameter` é visível, mas fora desse método não é, e o mesmo ocorre para `someLocalVariable`. Dentro do bloco `if`, `someOtherLocalVariable` é declarado e, fora desse bloco `if`, ele está fora do escopo. Por essa razão, dizemos que Java tem escopo de bloco, pois blocos (delimitados por `{ e }`) definem os limites do escopo.

O escopo tem muitas regras, mas Listagem 7 mostra aquelas mais comuns. Gaste alguns minutos para se familiarizar com elas.

### A instrução else

Às vezes, em um fluxo de controle de programa, você deseja efetuar ação apenas se uma determinada expressão falhar ao ser avaliada como `true`. É quando `else` entre em cena:

```
1 public int getHeight() {
2     int ret;
3     if (gender.equals("MALE"))
```

```
4     ret = height + 2;
5     else {
6         ret = height;
7         Logger.getLogger("Person").info("Being honest about height...");
8     }
9     return ret;
10 }
```

A instrução else funciona da mesma forma que if, ou seja, executa apenas a próxima instrução na qual é executada. Neste caso, duas instruções são agrupadas em uma instrução composta (observe as chaves), que o programa então executa.

Também é possível usar else para executar uma verificação de if adicional:

```
1  if (conditional) {
2      // Block 1
3  } else if (conditional2) {
4      // Block 2
5  } else if (conditional3) {
6      // Block 3
7  } else {
8      // Block 4
9  } // End
```

Se conditional for avaliado como true, então Block 1 será executado e o programa irá para a próxima instrução depois da chave final (que é indicada por // End). Se conditional não for avaliado como true, então conditional2 será avaliado. Se conditional2 for true, então Block 2 será executado e o programa irá para a próxima instrução depois da chave final. Se conditional2 não for true, o programa irá para conditional3 e assim por diante. Apenas se todos os três condicionais falharem, Block 4 será executado.

O operador ternário

A linguagem Java fornece um operador útil para efetuar verificações simples da instrução if / else. Sua sintaxe é:

```
1  (conditional) ? statementIfTrue : statementIfFalse;
```

Se conditional for avaliado como true, então statementIfTrue será executado; caso contrário, statementIfFalse será executado. Instruções compostas não são permitidas para nenhuma instrução.

O operador ternário será útil quando você souber o que precisa para executar uma instrução como o resultado da avaliação condicional como true e outra caso não saiba. Operadores ternários são usados mais frequentemente para inicializar uma variável (como um valor de retorno), como a seguir:

```
1  public int getHeight() {
2      return (gender.equals("MALE")) ? (height + 2) : height;
```

```
3 }
```

Os parênteses após o ponto de interrogação não são estritamente necessários, mas eles tornam o código mais legível.

## Loops

Além de ser capaz de aplicar condições para seus programas e ver diferentes resultados com base nos vários cenários if/then, às vezes você deseja que seu código efetue a mesma ação mais e mais vezes, até que a tarefa seja concluída. Nesta seção, aprenda sobre duas construções usadas para iteração por código ou execute-o mais de uma vez: loops for e while.

O que é um loop?

Um loop é uma construção de programação que executa repetidamente enquanto alguma condição (ou conjunto de condições) é atendida. Por exemplo, você pode solicitar a um programa que leia todos os registros até o final de um arquivo ou pode efetuar loop de todos os elementos em uma matriz, processando cada um. (Você aprende sobre matriz na seção "Coleções Java" deste tutorial.)

Loops for

A construção básica de loop na linguagem Java é a instrução for, que você pode usar para iterar em um intervalo de valores para determinar quantas vezes executar um loop. A sintaxe abstrata de um loop for é:

```
1 for (initialization; loopWhileTrue; executeAtBottomOfEachLoop) {  
2     statementsToExecute  
3 }
```

No início do loop, a instrução de inicialização é executada (diversas instruções de inicialização podem ser separadas por vírgulas). Desde que loopWhileTrue (uma expressão condicional Java que deve ser avaliada como true ou false) seja true, o loop será executado. Na parte inferior do loop, executeAtBottomOfEachLoop é executado.

Exemplo de um loop for

Se você quiser mudar um método main() para executar três vezes, pode usar um loop for, conforme mostrado na Listagem 8.

Lista 8. Um loop for

```
1 public static void main(String[] args) {  
2     Logger l = Logger.getLogger(Person.class.getName());  
3     for (int aa = 0; aa < 3; aa++) {  
4         Person p = new Person("Joe Q Author", 42, 173, 82, "Brown", "MALE");  
5         l.info("Loop executing iteration# " + aa);  
6         l.info("Name: " + p.getName());  
7         l.info("Age:" + p.getAge());  
8         l.info("Height (cm):" + p.getHeight());  
9         l.info("Weight (kg):" + p.getWeight());  
10        l.info("Eye Color:" + p.getEyeColor());
```

```
11    l.info("Gender:" + p.getGender());
12  }
13 }
```

A variável local aa é inicializada como zero no início da Listagem 8. Essa instrução executa apenas uma vez, quando o loop é inicializado. O loop continua então três vezes, e sempre que aa é incrementado por um.

Como você verá posteriormente, uma sintaxe de loop for alternativa está disponível para loop em construções que implementam a interface Iterable (como matrizes e outras classes do utilitário Java). Por enquanto, apenas observe o uso da sintaxe de loop for na Listagem 8.

#### Loops while

A sintaxe para um loop while é:

```
1  while (condition) {
2    statementsToExecute
3  }
```

Como você pode suspeitar, a condição while é avaliada como true, portanto, o loop é executado. Na parte superior de cada iteração (ou seja, antes da execução de qualquer instrução), a condição é avaliada. Se a condição for avaliada como true, o loop será executado. Portanto, é possível que um loop while nunca seja executado se sua expressão condicional não for true pelo menos uma vez.

Consulte novamente o loop for na Listagem 8. Para comparação, Listagem 9 usa um loop while para obter o mesmo resultado.

#### Lista 9. Um loop while

```
1  public static void main(String[] args) {
2    Logger l = Logger.getLogger(Person.class.getName());
3    int aa = 0;
4    while (aa < 3) {
5      Person p = new Person("Joe Q Author", 42, 173, 82, "Brown", "MALE");
6      l.info("Loop executing iteration# " + aa);
7      l.info("Name: " + p.getName());
8      l.info("Age:" + p.getAge());
9      l.info("Height (cm):" + p.getHeight());
10     l.info("Weight (kg):" + p.getWeight());
11     l.info("Eye Color:" + p.getEyeColor());
12     l.info("Gender:" + p.getGender());
13     aa++;
14   }
15 }
```

Como você pode ver, um loop while requer um pouco mais de manutenção que um loop for. Você deve inicializar a variável aa e também lembrar de incrementá-la na parte inferior do loop.

#### Loops do...while

Se quiser um loop que sempre seja executado uma vez e depois verifique sua expressão condicional, tente usar um loop do...while, conforme mostrado na Listagem 10.

#### Lista 10. Um loop do...while

```
1  int aa = 0;
2  do {
3      Person p = new Person("Joe Q Author", 42, 173, 82, "Brown", "MALE");
4      l.info("Loop executing iteration# " + aa);
5      l.info("Name: " + p.getName());
6      l.info("Age:" + p.getAge());
7      l.info("Height (cm):" + p.getHeight());
8      l.info("Weight (kg):" + p.getWeight());
9      l.info("Eye Color:" + p.getEyeColor());
10     l.info("Gender:" + p.getGender());
11     aa++;
12 } while (aa < 3);
```

A expressão condicional (aa < 3) não é verificada até o término do loop.

#### Ramificação de loop

Às vezes, é necessário resgatar um loop antes que a expressão condicional seja avaliada como false. Esta situação pode ocorrer se você estiver procurando uma matriz de Strings para um determinado valor e, depois de encontrá-la, não se importa com os outros elementos da matriz. Para os momentos nos quais você deseja resgatar, a linguagem Java fornece a instrução break, mostrada na Listagem 11.

#### Lista 11. Uma instrução break

```
1  public static void main(String[] args) {
2      Logger l = Logger.getLogger(Person.class.getName());
3      int aa = 0;
4      while (aa < 3) {
5          if (aa == 1)
6              break;
7          Person p = new Person("Joe Q Author", 42, 173, 82, "Brown", "MALE");
8          l.info("Loop executing iteration# " + aa);
9          l.info("Name: " + p.getName());
```

```
10    l.info("Age:" + p.getAge());
11    l.info("Height (cm):" + p.getHeight());
12    l.info("Weight (kg):" + p.getWeight());
13    l.info("Eye Color:" + p.getEyeColor());
14    l.info("Gender:" + p.getGender());
15    aa++;
16 }
17 }
```

A instrução break o leva para a próxima instrução executável fora do loop no qual ela está localizada.

#### Continuação de loop

No exemplo (simplista) na Listagem 11, você só deseja executar o loop apenas uma vez e resgatar. Também é possível ignorar uma única iteração de um loop, mas continuar executando o loop. Para esse propósito, é necessária a instrução continue, mostrada na Listagem 12.

#### Lista 12. Uma instrução continue

```
1    public static void main(String[] args) {
2        Logger l = Logger.getLogger(Person.class.getName());
3        int aa = 0;
4        while (aa < 3) {
5            if (aa == 1)
6                continue;
7            else
8                aa++;
9        Person p = new Person("Joe Q Author", 42, 173, 82, "Brown", "MALE");
10       l.info("Loop executing iteration# " + aa);
11       l.info("Name: " + p.getName());
12       l.info("Age:" + p.getAge());
13       l.info("Height (cm):" + p.getHeight());
14       l.info("Weight (kg):" + p.getWeight());
15       l.info("Eye Color:" + p.getEyeColor());
16       l.info("Gender:" +
17           p.getGender());
18     }
19 }
```

Na Listagem 12, você ignora a segunda iteração de um loop, mas continua na terceira. continue será útil quando você estiver, digamos, processando registros e se depara com um registro que definitivamente não deseja processar. É possível ignorar esse registro e ir para o próximo.

## Coleções Java

A maioria dos aplicativos reais lidam com coleções de itens como arquivos, variáveis, registros de arquivos ou conjuntos de resultados do banco de dados. A linguagem Java possui uma Estrutura de coleções sofisticada que permite criar e gerenciar coleções de objetos de vários tipos. Esta seção não ensinará sobre Coleções Java, mas apresentará as classes de coleções mais usadas e o introduzirá a elas.

### Matrizes

A maioria das linguagens de programação inclui o conceito de uma matriz para manter uma coleção de itens e a linguagem Java não é exceção. Uma matriz não é nada mais que uma coleção de elementos do mesmo tipo.

Observação: os colchetes nestes exemplos de código da seção fazem parte da sintaxe requerida para Coleções Java, não são indicadores de elementos opcionais.

É possível declarar uma matriz de uma das duas formas:

- Crie-a com um determinado tamanho, que é fixo pela vida útil da matriz.
- Crie-a com um determinado conjunto de valores iniciais. O tamanho desse conjunto determina o tamanho da matriz —, se ele é suficientemente largo para manter todos esses valores, e seu tamanho é fixo pela vida útil da matriz.

### Declarando uma matriz

No geral, você declara uma matriz como a seguir:

```
1 new elementType [arraySize]
```

É possível criar uma matriz de número inteiro de elementos de duas formas. Essa instrução cria uma matriz que possui espaço para cinco elementos, mas está vazia:

```
1 // creates an empty array of 5 elements:  
2 int[] integers = new int[5];
```

Esta instrução cria a matriz e a inicializa de uma só vez:

```
1 // creates an array of 5 elements with values:  
2 int[] integers = new int[] { 1, 2, 3, 4, 5 };
```

Os valores iniciais estão entre chaves e são separados por vírgulas.

Outra forma de criar uma matriz é criá-la e depois codificar um loop para iniciá-la:

```
1 int[] integers = new int[5];  
2 for (int aa = 0; aa < integers.length; aa++) {  
3     integers[aa] = aa+1;  
4 }
```

O código anterior declara uma matriz de número inteiro de cinco elementos. Se você tentar inserir mais que cinco elementos na matriz, o Java Runtime lançará uma exceção. Você aprenderá sobre exceções e como lidar com elas na Parte 2.

Carregando uma matriz

Para carregar a matriz, efetue o loop usando números inteiros de 1 ao comprimento da matriz (que você obterá chamando `length` na matriz — ; mais sobre isso a seguir). Neste caso, você para quando chega a 5.

Depois que a matriz é carregada, você pode acessá-la como antes:

```
1  Logger l = Logger.getLogger("Test");
2  for (int aa = 0; aa < integers.length; aa++) {
3    l.info("This little integer's value is: " + integers[aa]);
4  }
```

Esta sintaxe mais recente (disponível desde o JDK 5) também funciona:

```
1  Logger l = Logger.getLogger("Test");
2  for (int i : integers) {
3    l.info("This little integer's value is: " + i);
4  }
```

Eu acho a sintaxe mais recente mais fácil de trabalhar e a uso nesta seção.

O índice de elemento

Pense em uma matriz como uma série de depósitos e em cada um deles há um elemento de um determinado tipo. O acesso a cada depósito é obtido usando um índice:

```
1  element = arrayName [elementIndex];
```

Para acessar um elemento, é necessário fazer referência à matriz (seu nome) e ao índice no qual o elemento que você deseja reside.

O método `length`

Um método útil, como já foi visto, é `length`. Ele é um método integrado, portanto sua sintaxe não inclui os parênteses usuais. Apenas digite a palavra `length` e ele retornará, — conforme esperado, — o tamanho da matriz.

Matrizes na linguagem Java são baseadas em zero. Portanto, para algumas matrizes denominadas `array`, o primeiro elemento sempre reside em `array[0]` e o último reside em `array[array.length - 1]`.

Uma matriz de objetos

Você viu como as matrizes podem manter tipos primitivos, mas vale mencionar que elas também podem manter objetos. Nesse sentido, a matriz é a coleção mais utilitária da linguagem Java.

Criar uma matriz de objetos `java.lang.Integer` não é muito diferente de criar uma matriz de tipos primitivos. Novamente, você tem duas formas de fazer isso:

```
1  // creates an empty array of 5 elements:
2  Integer[] integers = new Integer[5];
1  // creates an array of 5 elements with values:
2  Integer[] integers = new Integer[] { Integer.valueOf(1),
```



```
3 Integer.valueOf(2)
4 Integer.valueOf(3)
5 Integer.valueOf(4)
6 Integer.valueOf(5);
```

#### Boxing e unboxing

Cada tipo primitivo na linguagem Java tem uma classe de contraparte JDK, que você pode ver na Tabela 4.

Tablela 4. Primitivas e contrapartes JDK

Primitiva	Contraparte JDK
boolean	java.lang.Boolean
byte	java.lang.Byte
char	java.lang.Character
short	java.lang.Short
int	java.lang.Integer
long	java.lang.Long
float	java.lang.Float
double	java.lang.Double

Cada classe JDK fornece métodos para analisar e converter de sua representação interna em um tipo de primitiva correspondente. Por exemplo, este código converte o valor decimal 238 em um Integer:

```
1 int value = 238;
2 Integer boxedValue = Integer.valueOf(value);
```

Esta técnica é conhecida como boxing, pois você está colocando a primitiva em um wrapper ou caixa.

De modo semelhante, para converter a representação Integer novamente a sua contraparte int, você efetua unbox dela:

```
1 Integer boxedValue = Integer.valueOf(238);
2 int intValue = boxedValue.intValue();
```

Autoboxing e auto-unboxing

Falando estritamente, não é necessário efetuar box e unbox das primitivas explicitamente. Ao contrário, é possível usar os recursos de autoboxing e auto-unboxing da linguagem Java:

```
1 int intValue = 238;
2 Integer boxedValue = intValue;
3 //
4 intValue = boxedValue;
```

Entretanto, eu recomendo que você evite autoboxing e auto-unboxing, pois isso pode levar a problemas de leitura de código. O código nos fragmentos de boxing e unboxing é mais óbvio e, portanto, mais legível que o código com autoboxing; eu acredito que isso compensa o esforço extra.

Analisando e convertendo tipos com boxing

Você viu como obter um tipo com boxing, mas o que dizer da análise de um String que você suspeita ter um tipo com boxing em sua caixa correta? As classes de wrapper JDK possuem métodos para isso também:

```
1 String characterNumeric = "238";
2 Integer convertedValue = Integer.parseInt(characterNumeric);
```

Também é possível converter o conteúdo de um tipo de wrapper JDK em um String:

```
1 Integer boxedValue = Integer.valueOf(238);
2 String characterNumeric = boxedValue.toString();
```

Observe que ao usar o operador de concatenação em uma expressão String (você já viu isso em chamadas de Logger), o tipo de primitiva tem autoboxing efetuado e os tipos de wrapper automaticamente têm toString() chamado. Muito útil.

Listas

List é uma construção de coleções que é, por definição, uma coleção ordenada, também conhecida como sequência. Como List é ordenado, você possui controle total sobre o local para onde os itens de List irão. Uma coleção Java List pode ter apenas objetos e define um contrato estrito sobre seus comportamentos.

List é uma interface, portanto, não é possível instanciá-la diretamente. Você trabalhará com sua implementação mais comumente usada, ArrayList. Há duas formas de fazer a declaração. Primeiro, usando a sintaxe explícita:

```
1 List<String> listOfStrings = new ArrayList<String>();
```

Segundo, usando o operador "diamond", apresentado no JDK 7:

```
1 List<String> listOfStrings = new ArrayList<>();
```

Observe que o tipo de objeto na instanciação ArrayList não é especificado. Isso ocorre porque o tipo de classe à direita da expressão deve corresponder ao tipo do lado esquerdo. No restante deste tutorial, eu uso os dois tipos, pois você provavelmente verá os dois usos na prática.

Observe que eu designei o objeto ArrayList para uma variável do tipo List. Com programação Java, é possível designar uma variável de um tipo para outro, desde que a variável que está sendo designada seja uma superclasse ou interface implementada pela variável da qual ela está sendo designada. É possível saber mais sobre como as designações de variáveis são afetadas na Parte 2 na seção "Herança".

Tipo formal

O <Object> no fragmento de código precedente é chamado de tipo formal. <Object> informa ao compilador que List contém uma coleção do tipo Object, o que significa que você pode inserir o que desejar em List.

Se você quiser deixar as restrições mais rigorosas sobre o que é possível ou não inserir em List, poderá definir o tipo formal de forma diferente:

```
1 List<Person> listOfPersons = new ArrayList<Person>();
```

Agora List poderá conter apenas instâncias Person.

Usando Lists

Usar Lists é super fácil, como as coleções Java no geral. A seguir, algumas das tarefas que se pode fazer com Lists:

- Inserir algo em List.
- Perguntar a List qual seu tamanho no momento.
- Retirar algo de List.

Agora, você pode tentar algumas dessas tarefas. Você já viu como criar uma instância de List instanciando seu tipo de implementação ArrayList, portanto, pode começar daqui.

Para inserir algo em List, chame o método add() :

```
1 List<Integer> listOfIntegers = new ArrayList<>();
2 listOfIntegers.add(Integer.valueOf(238));
```

O método add() inclui o elemento ao final de List.

Para perguntar a List seu tamanho, chame size():

```
1 List<Integer> listOfIntegers = new ArrayList<>();
2
3 listOfIntegers.add(Integer.valueOf(238));
4 Logger l = Logger.getLogger("Test");
5 l.info("Current List size: " + listOfIntegers.size());
```

Para recuperar um item de List, chame get() e o transmita ao índice do item que deseja:

```
1 List<Integer> listOfIntegers = new ArrayList<>();
2 listOfIntegers.add(Integer.valueOf(238));
3 Logger l = Logger.getLogger("Test");
4 l.info("Item at index 0 is: " + listOfIntegers.get(0));
```

Em um aplicativo real, List conteria registros ou objetos de negócios, e você possivelmente desejaria consultá-los como parte de seu processamento. Como fazer isso de uma forma genérica? Você deseja fazer a iteração da coleção, o que pode ser feito porque List implementa a interface `java.lang.Iterable`. (Você aprendeu sobre interfaces na Parte 2.)

### Iterable

Se uma coleção implementar `java.lang.Iterable`, ela será chamada de coleção com possível iteração. É possível iniciar em uma extremidade e percorrer a coleção item por item até que os itens se esgotem.

Você já viu a sintaxe especial para iteração de coleções que implementam a interface `Iterable`, na seção "Loops". Novamente aqui:

```
1 for (objectType varName : collectionReference) {
2     // Start using objectType (via varName) right away...
3 }
```

### Iterando em List

Esse exemplo anterior era abstrato; agora, há um exemplo mais realista:

```
1 List<Integer> listOfIntegers = obtainSomehow();
2 Logger l = Logger.getLogger("Test");
3 for (Integer i : listOfIntegers) {
4     l.info("Integer value is : " + i);
5 }
```

Este pequeno fragmento de código efetua a mesma ação do fragmento maior:

```
1 List<Integer> listOfIntegers = obtainSomehow();
2 Logger l = Logger.getLogger("Test");
3 for (int aa = 0; aa < listOfIntegers.size(); aa++) {
4     Integer i = listOfIntegers.get(aa);
5     l.info("Integer value is : " + i);
6 }
```

O primeiro fragmento usa sintaxe abreviada: não há nenhuma variável index (`aa` neste caso) para inicializar e nenhuma chamada para `get()` de List.

Como List estende `java.util.Collection`, que implementa `Iterable`, é possível usar a sintaxe abreviada para iteração em qualquer List.

### Sets

Um Set é uma construção de coleções que, por definição, contém elementos exclusivos — ou seja, nenhuma duplicata. Considerando que List pode conter o mesmo objeto centenas de vezes, um Set só pode conter uma determinada instância uma vez. Uma coleção Java Set pode ter apenas objetos e define um contrato estrito sobre seus comportamentos.

Como Set é uma interface, ela não pode ser instanciada diretamente, portanto, aqui está uma das minhas implementações favoritas: `HashSet`. `HashSet` é fácil de usar e é semelhante a List.

A seguir, algumas das tarefas que se pode fazer com Set:

- Inserir algo em Set.
- Perguntar a Set qual seu tamanho no momento.
- Retirar algo de Set.

Usando Sets

Um atributo de distinção de Set é o que garante exclusividade entre seus elementos, mas a ordem dos elementos não é importante. Considere o seguinte código:

```
1 Set<Integer> setOfIntegers = new HashSet<Integer>();
2 setOfIntegers.add(Integer.valueOf(10));
3 setOfIntegers.add(Integer.valueOf(11));
4 setOfIntegers.add(Integer.valueOf(10));
5 for (Integer i : setOfIntegers) {
6     l.info("Integer value is: " + i);
7 }
```

É possível esperar que Set tenha três elementos, mas ele só possui dois porque o objeto Integer que contém o valor 10 é incluído apenas uma vez.

Mantenha este comportamento em mente ao fazer iteração com Set, como a seguir:

```
1 Set<Integer> setOfIntegers = new HashSet();
2 setOfIntegers.add(Integer.valueOf(10));
3 setOfIntegers.add(Integer.valueOf(20));
4 setOfIntegers.add(Integer.valueOf(30));
5 setOfIntegers.add(Integer.valueOf(40));
6 setOfIntegers.add(Integer.valueOf(50));
7 Logger l = Logger.getLogger("Test");
8 for (Integer i : setOfIntegers) {
9     l.info("Integer value is : " + i);
10 }
```

Os objetos impressos em uma ordem diferente da ordem que você os incluiu são casuais, pois Set garante exclusividade, não ordem. É possível ver isso ao colar o código anterior no método main() de sua classe Person e executá-lo.

Maps

Um Map é uma construção de coleção útil que você pode usar para associar um objeto (a chave) a outro (o valor). Como você pode imaginar, a chave para Map deve ser exclusiva e ela é usada para recuperar o valor posteriormente. Uma coleção Java Map pode ter apenas objetos e define um contrato estrito sobre seus comportamentos.

Como Map é uma interface, ela não pode ser instanciada diretamente, portanto, aqui está uma das minhas implementações favoritas: HashMap.

A seguir, algumas das tarefas que se pode fazer com Maps:

- Inserir algo em Map.
- Retirar algo de Map.
- Obter um Set de chaves para Map—, para iteração nele.

Usando Maps

Para inserir algo em Map, é necessário ter um objeto que represente sua chave e um objeto que represente seu valor:

```
1 public Map<String, Integer> createMapOfIntegers() {
2     Map<String, Integer> mapOfIntegers = new HashMap<>();
3     mapOfIntegers.put("1", Integer.valueOf(1));
4     mapOfIntegers.put("2", Integer.valueOf(2));
5     mapOfIntegers.put("3", Integer.valueOf(3));
6     //...
7     mapOfIntegers.put("168", Integer.valueOf(168));
8 }
```

Neste exemplo, Map contém Integer s, encadeados por String, que pode ser suas representações de String. Para recuperar um valor Integer específico, é necessário sua representação de String :

```
1 mapOfIntegers = createMapOfIntegers();
2 Integer oneHundred68 = mapOfIntegers.get("168");
```

Usando Set com Map

No momento, é possível que você tenha uma referência a Map e queira percorrer todo seu conjunto de conteúdo. Nesse caso, será necessário um Set de chaves para Map:

```
1 Set<String> keys = mapOfIntegers.keySet();
2 Logger l = Logger.getLogger("Test");
3 for (String key : keys) {
4     Integer value = mapOfIntegers.get(key);
5     l.info("Value keyed by '" + key + "' is '" + value + "'");
6 }
```

Observe que o método toString() de Integer recuperado de Map é automaticamente chamado quando usado na chamadaLogger. Map não retorna List de suas chaves, pois Map está encadeado e cada chave é exclusiva. A exclusividade é a característica de distinção de um Set.

Fazendo archive do código Java

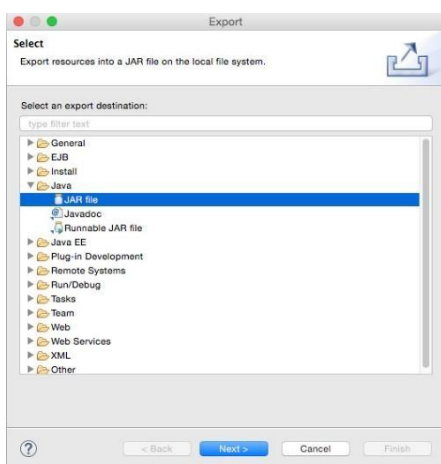
Agora que você aprendeu um pouco sobre como escrever aplicativos Java, pode desejar saber como empacotá-los para que outros desenvolvedores possam usá-los ou como importar o código de outro desenvolvedor em seus aplicativos. Esta seção mostra como fazer isso.

## JARs

O JDK é fornecido com uma ferramenta chamada JAR, que é responsável pelo Java archive. Você usa esta ferramenta para criar arquivos JAR. Depois de empacotar seu código em um arquivo JAR, outros desenvolvedores podem descartar o arquivo JAR em seus projetos e configurar seus projetos para usar seu código.

Criar um arquivo JAR em Eclipse é fácil. Em sua área de trabalho, clique com o botão direito no pacote com `makotojava.intro` e clique em Arquivo > Exportar. Você verá a caixa de diálogo mostrada na Figura 10. Escolha Java > Arquivo JAR e clique em Avançar.

Figura 10. Exportar caixa de diálogo



Quando a próxima caixa de diálogo for aberta, navegue até o local no qual deseja armazenar seu arquivo JAR e nomeie o arquivo da forma que desejar. A extensão `.jar` é o padrão, que eu recomendo usar. Clique em Concluir.

Você verá o arquivo JAR no local selecionado. É possível usar as classes nele a partir de seu código se você inserir o JAR em seu caminho de construção em Eclipse. Fazer isso também é fácil, como verá a seguir.

## Usando aplicativos de terceiros

À medida que você se sente mais confortável ao escrever aplicativos Java, pode desejar usar mais e mais aplicativos de terceiros para suportar seu código. Embora o JDK seja muito bom, ele não fornece tudo o que você precisa para escrever um grande código Java. A comunidade de software livre Java fornece muitas bibliotecas para ajudá-lo a eliminar essas lacunas. A título de exemplo, suponha que você deseje usar Commons Lang, uma biblioteca de substituição de JDK para manipular as classes principais Java. As classes fornecidas por Commons Lang o ajudam a manipular matrizes, criar números aleatórios e executar manipulação de sequência.

Assumiremos que você já tenha efetuado o download de Commons Lang, que está armazenada em um arquivo JAR. Para usar as classes, a primeira etapa é criar um diretório `lib` em seu projeto e eliminar o arquivo JAR:

1. Clique com o botão direito na pasta raiz `Intro` na visualização Explorador de Projetos.
2. Clique em Nova > Pasta e chame a pasta `lib`.
3. Clique em Concluir.





## Spring Framework

Muitos já ouviram falar nesse framework (Spring), mas a complexidade inicial afasta ou assusta bastante os iniciantes do mundo Java. Sim, a princípio e principalmente para um iniciante o Spring pode parecer nada simples, mas para desenvolvedores mais experientes, logo pode-se perceber o encanto e a mágica do Spring Framework. Nesse artigo vamos desvendar um pouco desse completíssimo framework.

Spring é um framework de código aberto (open source), criado por Rod Johnson, em meados de 2002, e apresentado no seu livro Expert One-on-One: JEE Design and Development. Foi criado com o intuito simplificar a programação em Java, possibilitando construir aplicações que antes só era possível utilizando EJB's.

O Spring atualmente possui diversos módulos como Spring Data (trata da persistência), Spring Security (trata da segurança da aplicação) entre outros módulos. Mas o principal (core) pode ser utilizado em qualquer aplicação Java, as principais funcionalidades são a injeção de dependência (CDI) e a programação orientada a aspectos (AOP), cabe ao desenvolvedor dizer ao Spring o que quer usar. O que faz dele uma poderosa ferramenta, pois não existe a necessidade de se arrastar todas as ferramentas do framework para criar uma aplicação simples.

## Configurando o Spring

Para esse artigo vamos utilizar o Eclipse e o Tomcat 7, certifique-se de tê-los instalados antes de continuar.

Primeiramente criaremos um “Dynamic Web Project” com o nome de “hello-spring”, como mostra a figura 1.

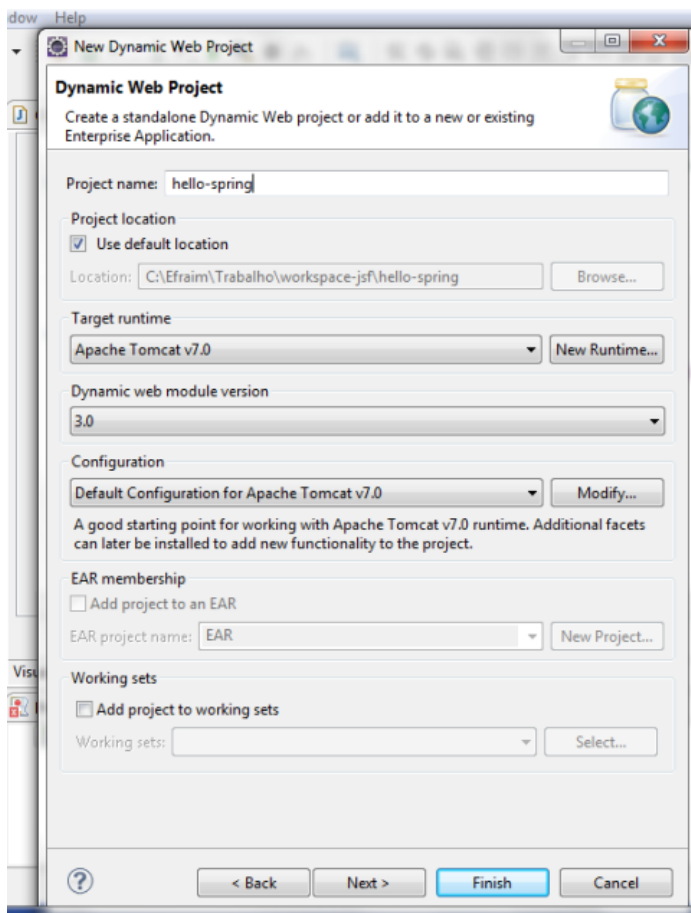


Figura 1: Criando o projeto

Agora vem a parte mais “complicada”, configura o Spring, a princípio algo bastante complexo para aqueles que ainda não possuem muita pratica, lembrando que todo framework é um bicho de sete cabeças a primeira vista.

Para o Spring funcionar, vamos precisar de suas libs, acessando os links abaixo, vamos encontrar tudo o que precisamos.

No canto direito da tela teremos as últimas versões lançadas, usaremos a mais atual. Após baixar o zip do framework, vamos descomprimi-lo, e acessar a pasta libs do framework, não se assuste, não usaremos todos esses jar’s , vamos copiar “spring-webmvc-X-X.jar” , “spring-web- X-X.jar”, “spring-expression-X-X.jar”, “spring-core- X-X.jar”, “spring-context- X-X.jar”, “spring-beans- X-X.jar” e colar na pasta “WEB-INF/lib”. Também é necessária a biblioteca “commons-logging-1.1.1.jar” para nosso projeto, ver figura 2.

Observação: X-X trata-se da versão do framework que foi baixada.

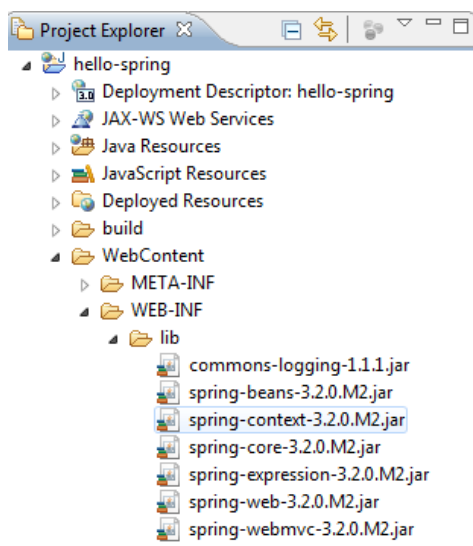


Figura 2: Monstrando JAR’s do projeto

Com posse da biblioteca do framework, agora vamos criar uma pasta chamada “spring” dentro do diretório WEB-INF, e dentro da pasta criaremos um arquivo xml chamado “application-context.xml”, lembrando que o Spring é um framework “container-based”, ou seja, ele vai conter e carregar o que você informá-lo. A raiz do nosso xml é tag <beans></beans> e dentro conterá toda a configuração do Spring

Observação: não é obrigatória a configuração do Spring em apenas um “.xml”, é possível separar vários arquivos de configuração, exemplo: persistence-context.xml mvc-context.xml etc... , porém nesse artigo isso não será abordado.

**Listagem 1:** Como deve ficar nosso “application-context.xml”

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans xmlns="http://www.springframework.org/schema/beans"
```

```
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
    xmlns:context="http://www.springframework.org/schema/context"
```

```
    xmlns:mvc="http://www.springframework.org/schema/mvc"
```

```
    xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd
```

```
        http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
```

<http://www.springframework.org/schema/context> <http://www.springframework.org/schema/context/spring-context-3.0.xsd>>

```

<!-- Informa o pacote onde o Spring ira buscar as classes anotadas (@Controller, @Service...) -->
    <context:component-scan base-package="br.com.devmedia" />
    <!-- Diz ao Spring que ele deve usar a configuração das annotations -->
    <mvc:annotation-driven />
    <!-- Define pagina inicial (ignora a configuração do web.xml)-->
    <mvc:view-controller path="/" view-name="helloworld"/>
    <!-- Define onde está localizada as views da aplicação, e qual a extensão das mesmas -->
    <!--
    Estão configuradas dentro da WEB-INF para que o usuário não possa acessalas, se
    não por meio do mapeamento
    -->
    <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/views"/>
        <property name="suffix" value=".jsp"/>
    </bean>
</beans>
    
```

Arquivo de configuração bem simples não? Claro, é uma aplicação simples, e a medida que sua aplicação cresce, basta ir acrescentando módulos de acordo com sua necessidade, evitando assim um consumo gigantesco de memória com coisas que você nunca vai utilizar na sua aplicação.

Agora vamos dizer a nossa aplicação Web para carregar o Spring, mas como? O bom e velho arquivo "web.xml", será necessária apenas a configuração do Servlet do Spring, nada sobrenatural, veja como deve ficar o nosso arquivo.

#### Listagem 2: Arquivo web.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/ja-
vae/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

    <display-name>hello-spring</display-name>

    <!-- Configura o Spring Servlet -->
    <servlet>
        <servlet-name>Spring-Servlet</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>

        <init-param>
    
```

```

        <param-name>contextConfigLocation</param-name>
        <param-value><!-- Especifica que arquivo de configuração sera cha-
mado, junto a instanciação -->
            /WEB-INF/spring/application-context.xml
        </param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping><!-- Mapeia o Servlet -->
    <servlet-name>Spring-Servlet</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>
    
```

### Exibindo uma pagina

Então é só isso? Podemos rodar a aplicação? Calma, ainda não existe nenhuma página, então criaremos uma dentro da pasta “WEB-INF/view” com o nome de “helloworld.jsp”. Veja que é o mesmo nome o qual configuramos como página inicial no “application-context.xml”.

#### Listagem 3: Definindo a página inicial

```

<!-- Define pagina inicial (ignora a configuração do web.xml)-->
    <mvc:view-controller path="/" view-name="helloworld"/>
    
```

Nossa página ficará assim.

#### Listagem 4: Página helloworld.jsp

```

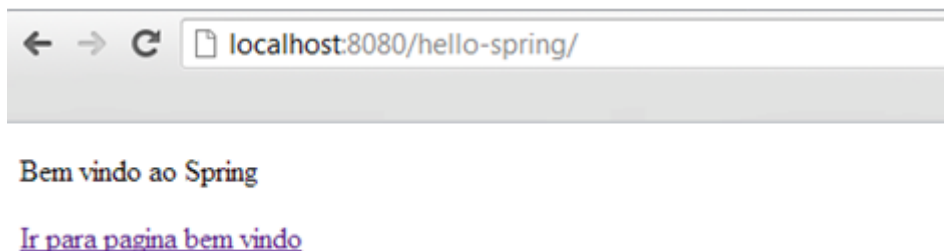
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Dev Media Hello Spring</title>
</head>
<body>
    <p>Hello World Spring</p>
    <br/><br/>
    
```

```
<a href="bemvindo">Ir para pagina bem vindo</a>
```

```
</body>
```

```
</html>
```

Agora sim, você já pode rodar o projeto no tomcat, veja o resultado na figura 3.



**Figura 3:** Exibindo um Hello World

### Controllers

Exibir uma página é muito simples, mas qual a diferença do Spring? Vamos falar sobre o controle de navegação da aplicação. O Spring utiliza os chamados Controllers, que são classes mapeadas através de annotations que servem para dizer à aplicação o que exibir quando for requisitada uma página, ou envio de informações, uma espécie de Servlet do JSP, mas bem mais fácil de se trabalhar.

Vamos a um exemplo, criaremos uma classe chamada “HelloController” no pacote “br.com.devmedia.controllers”, se o pacote não existir, crie-o. Vamos anotar essa classe com o @Controller, que diz ao Spring que essa classe vai funcionar como uma espécie de Servlet para a aplicação, ou seja, vai receber requisições tratá-las e responder ao usuário.

#### Listagem 5: Anotação @Controller

```
@Controller
```

```
public class HelloController
```

Espera ai, mas como minha aplicação vai saber qual requisição deve ser tratada pelo meu @Controller? Simples, através da anotação @RequestMapping, ela vai verificar qual url está sendo solicitada e enviar para o @Controller que contém a anotação com a url específica, veja.

#### Listagem 6: Método bemVindo

```
@Controller
```

```
public class HelloController {
```

```
    @RequestMapping("/bemvindo")
```

```
    public ModelAndView bemVindo(Model model){
```

```
        model.addAttribute("bemvindo" , "Olha só que facil dizer bem vindo");
```

```
        return new ModelAndView("bemvindo");
```

Explicando o código, o @RequestMapping diz que quando for requisitada a url /bemvindo, será executado o método “bemVindo()” da classe “HelloController”. Mas o que é esse “Model” e de onde ele vem? Por que ele está ali? O Model não é um parâmetro obrigatório, e com um pouco mais de experiência verá que muita coisa não é. O “Model” vai servir para adicionar atributos para serem usados na tela, não se preocupe com detalhes, o Spring vai fazer isso por você. Devem ter notado também o



**Spring Boot:**

Este artigo é útil para entendermos o que é uma arquitetura baseada em micro serviços e aprender como implementar uma aplicação baseada neste padrão com a ferramenta que será apresentada, o Spring Boot. Além disso, também será apresentado um novo modo de desenvolvimento voltado para a plataforma Java EE que possivelmente influenciará toda a comunidade de desenvolvedores Java.

Apesar de ser uma ferramenta poderosa e que aumenta significativamente a produtividade na escrita de aplicações corporativas, o Spring Framework ainda é alvo de críticas a respeito do tempo necessário para se iniciar o desenvolvimento de novos projetos. Pensando nisso, foi introduzido no Spring 4.0 um novo projeto que tem, dentre seus objetivos, responder a estas críticas e, como veremos neste artigo, também mudar bastante nossa percepção acerca do desenvolvimento de aplicações para a plataforma Java EE. Este projeto é o Spring Boot.

A principal crítica feita ao Spring é sobre o modo como configuramos o seu container de injeção de dependências e inversão de controle usando arquivos de configuração no formato XML. Artefatos estes que, conforme aumentam de tamanho, se tornam cada vez mais difíceis de serem mantidos, muitas vezes se transformando em um gargalo para a equipe de desenvolvimento. No decorrer da história do framework vimos que este problema foi sendo tratado a partir de uma série de melhorias no modo como declaramos nossos beans a cada novo release: namespaces na versão 1.2, anotações na versão 2.0 e, finalmente, passamos a poder tratar arquivos XML como um artefato opcional no lançamento da versão 3.0, que nos trouxe a possibilidade de declarar nossos beans usando apenas código Java e anotações.

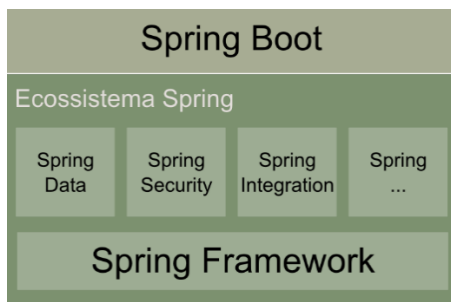
Outra crítica relevante diz respeito à complexidade na gestão de dependências. Conforme nossos projetos precisam interagir com outras bibliotecas e frameworks como, por exemplo, JPA, mensagem, frameworks de segurança e tantos outros, garantir que todas as bibliotecas estejam presentes no classpath da aplicação acaba se tornando um pesadelo. Não é raro encontrarmos em projetos baseados em Maven arquivos POM nos quais 90% do seu conteúdo sejam apenas para gestão de dependências. E esta é apenas a primeira parte do problema. O grande desafio surge quando precisamos integrar todos estes componentes.

O projeto Spring Boot (ou simplesmente Boot) resolve estas questões e ainda nos apresenta um novo modelo de desenvolvimento, mais simples e direto, sem propor novas soluções para problemas já resolvidos, mas sim alavancando as tecnologias existentes presentes no ecossistema Spring de modo a aumentar significativamente a produtividade do desenvolvedor.

**O Que É O Spring Boot?**

Trata-se de mais um framework, mas talvez a melhor denominação seja micro framework. Como mencionado na introdução deste artigo, seu objetivo não é trazer novas soluções para problemas que já foram resolvidos, mas sim reaproveitar estas tecnologias e aumentar a produtividade do desenvolvedor. Como veremos mais à frente, trata-se também de uma excelente ferramenta que podemos adotar na escrita de aplicações que fazem uso da arquitetura de micro serviços.

Se pudéssemos desenhar um diagrama arquitetural do Spring Boot, este seria muito similar ao que vemos no Grails: uma fina camada sobre tecnologias já consagradas pelo mercado, tal como podemos verificar na **Figura 1**. A grande mudança está no modo como agora empacotamos e acessamos estas soluções.



**Figura 1.** Posicionamento do Spring Boot no ecossistema Spring.

O desenvolvedor não precisa se preocupar em aprender novas tecnologias, pois todo o conhecimento adquirido sobre o ecossistema Spring é reaproveitado. A principal diferença se dá no modo como configuramos, organizamos nosso código e executamos a aplicação, tal como veremos neste artigo.

Como sabemos, todo framework se baseia em alguns princípios. No caso do Boot, são quatro:

1. Prover uma experiência de início de projeto (getting started experience) extremamente rápida e direta;
2. Apresentar uma visão bastante opinativa (opinionated) sobre o modo como devemos configurar nossos projetos Spring, mas ao mesmo tempo flexível o suficiente para que possa ser facilmente substituída de acordo com os requisitos do projeto;
3. Fornecer uma série de requisitos não funcionais já pré-configurados para o desenvolvedor como, por exemplo, métricas, segurança, acesso a base de dados, servidor de aplicações/servlet embarcado, etc.;
4. Não prover nenhuma geração de código e minimizar a zero a necessidade de arquivos XML.

### Uma Visão Opinativa Sobre A Configuração?

Na documentação oficial do projeto Boot, assim como em posts a seu respeito, encontraremos muitas vezes o termo *opinionated view* (visão opinativa). Ao citá-lo, os responsáveis pelo desenvolvimento da ferramenta na realidade estão se referindo ao conceito de convenção sobre configuração, porém levemente modificado.

O conceito de convenção sobre configuração é o grande motor por trás do ganho de produtividade do Spring Boot, porém não foi algo introduzido por ele. Frameworks como Ruby on Rails e Grails já o aplicam há bastante tempo. A ideia é bastante simples: dado que a maior parte das configurações que o desenvolvedor precisa escrever no início de um projeto são sempre as mesmas, por que já não iniciar um novo projeto com todas estas configurações já definidas?

Pense no modo como estamos habituados a trabalhar, por exemplo, em um projeto baseado em Spring MVC. Nossos primeiros passos serão incluir as dependências necessárias no projeto, adequar o arquivo `web.xml` e organizar a estrutura do nosso código fonte para que possamos iniciar o desenvolvimento. E isto é feito no início de todo projeto. Sendo assim, por que não já começar com isto pronto?

Voltando nossa atenção para esse termo, é importante prestar atenção na palavra **sobre** em “convenção **sobre**configuração”. Note que não é “convenção **ao invés** de configuração”. Não temos uma imposição aqui, mas sim sugestões. Deste modo, se seu projeto requer um aspecto diferente daquele definido pelas convenções do framework, o programador precisa alterar apenas aqueles locais nos quais a customização se aplica.

Indo além na análise dos termos mencionados, percebemos que no Spring Boot não foi usado o termo “convenção sobre configuração”, mas sim “visão opinada sobre configuração”. Dito isso, você pode estar se perguntando: Qual a diferença entre eles? A diferença está no fato da equipe de desenvolvimento do Spring Boot assumir que algumas escolhas tomadas baseiam-se em aspectos subjetivos e não estritamente técnicos. Um exemplo é a escolha da biblioteca de log. Foi adotado o Log4J, no entanto o Commons Logging seria uma opção igualmente competente. Sendo assim, por que um ao invés do outro? Em grande parte, preferências pessoais.

Há outra grande vantagem na adoção deste princípio. A partir do momento em que a equipe conhece as convenções, torna-se menor o tempo necessário para que novos membros se adaptem ao projeto e a manutenção passa a ser mais simples.

### Spring Scripts

O projeto Spring Boot nos permite criar dois tipos de aplicações: as tradicionais, escritas primariamente em Java; e uma segunda forma, chamada pela equipe de desenvolvimento do Boot de “Spring Scripts”, que nada mais são do que códigos escritos em Groovy – o que não é de se estranhar, dado



que desde o lançamento da versão 4.0 do framework há uma forte tendência da Pivotal em abraçar esta linguagem de programação e torná-la cada vez mais presente no dia a dia do programador Java.

Para ilustrar esse tipo de aplicação, vamos escrever um “Olá mundo!” bastante simples que nos conduzirá na apresentação de alguns conceitos fundamentais por trás do Spring Boot. Todo o código fonte do nosso projeto pode ser visto na **Listagem 1**.

**Listagem 1.** “Olá mundo” com Spring Scripts – OlaMundo.groovy.

```
@RestController
class OlaMundo {
    @RequestMapping("/")
    String home() {
        "olá mundo!"
    }
}
```

Conhecendo o Spring Boot!

### O que é Spring Boot

Se você programa em JAVA provavelmente já usou ou ouviu falar sobre o Spring Framework. Atualmente na versão 4.x é uma ferramenta poderosa, madura e com uma comunidade gigantesca por trás. Mas e o Spring Boot ? Bom, já vou chegar lá.

Conforme o Spring foi evoluindo, suas configurações necessárias foram crescendo, o que antes era um simples XML começou a parecer a jornada de Frodo(Senhor dos Anéis) para ser configurado. Outro problema comum era a quantidade de dependências que era preciso gerenciar, aquele arquivo pom.xml cada vez maior não era nada legal.

A própria equipe do Spring junto com feedbacks da comunidade perceberam a necessidade de tornar o seu poderoso framework mais ágil, e então surgiu o Spring Boot!

Em versões anteriores do Spring Framework, mesmo sem o “Boot” já era possível criar suas Beans e até mesmo sua aplicação inteira sem a necessidade de XML's. Tudo poderia ser feito por anotações e código Java.

O que o Spring Boot fez foi juntar todas essas facilidades que encontraram ao longo de sua evolução em um só lugar. Podemos dizer que o Spring juntou suas próprias tecnologias de forma a reaproveitá-la e tornar a sua vida mais produtiva na hora de desenvolver.

### Rápido E Leve

Me lembro que à alguns anos atrás, mesmo desenvolvendo com Spring MVC, algo que muito me incomodava era a hora de subir meu serviço para testar(mesmo localmente). Jboss, Glassfish, Weblogic e tantos outros Server's JAVA EE. Muitas vezes era aquela tortura para configurar, cada um com seu “jeito” de configuração, XML's gigantescos, e etc... Bom, esses tempos acabaram depois que conheci o Spring Boot! Um dos principais pontos que gosto do Spring Boot é o fato dele ser “Embedded”, ou seja, o nosso server application está dentro da nossa aplicação, e não se assuste, esse server pode ser um Tomcat embedded ou até mesmo um Jetty. Sendo assim fica tão leve e rápido como uma Robbit!

Não tenho nada contra Server's JAVA EE, mas a complexidade desses server's pode acabar atrapalhando a agilidade de desenvolvimento, e o seu uso pode não ser necessário dependendo da arquitetura que iremos trabalhar(será que sempre vou precisar de um server EE ? Acho que não rsrs).

Talvez você se pergunte, “mas meu Jboss eu configuro do jeito que eu quero!”, “No Weblogic eu parametrizo o que eu quiser” e muitas outras justificativas. Agora, será que precisa configura tudo isso pra começar a desenvolver? Será que o único modo de escalar bem seus serviços é usando Servers EE? Tantas configurações ajudam ou atrapalham? É de se pensar...

Claro que existe uma curva de aprendizado para utilizar o Spring Boot, e o que quero mostrar aqui é que essa curva não é tão curva assim rsrs, vai valer a pena acredite! E alias, essas configurações que fazemos em outros servers também conseguimos fazer no Jetty ou Tomcat(lembra que são embedded, então não é mágica, existe um server rodando ali). A diferença é que essas configurações são muito mais enxutas e sem tantas complexidades.

Outro ponto bem legal, como podemos usar nosso server embedded, basta dar um start em nosso JAR e sua aplicação estará de pé. Sim, JAR e não WAR, podemos gerar um WAR e colocar ele dentro de um server qualquer se quisermos, mas porque faríamos isso se podemos simplesmente dar o comando `java -jar gandalf.jar` e tudo funciona, pois nosso server está junto com nossa aplicação.

E Se Eu Precisar Usar Outras Bibliotecas, Acessar Dados, Exportar Relatórios E Etc ?

É legal frisar que por traz do Spring Boot existe uma comunidade chamada "SPRING" e esses caras são poderosos. Existem inúmeros projetos da Spring.io e todos se conversam muito bem entre si, seria super tranquilo fazer uma aplicação com Spring Boot usando o Spring Data para acessar um MongoDB por exemplo (O Alexandre Queiroz escreveu sobre Spring Data e MongoDB aqui no blog recentemente, vale conferir). Não gosta do Spring Data? Sem problemas, você pode usar as bibliotecas que quiser dentro do seu projeto, alias, muitas vezes você vai precisar.

### Mão Na Massa

Agora que já falei bastante sobre Spring Boot, que tal construirmos algo? (E sem XML, a não ser nosso pom é claro)

Vamos montar uma calculadora Rest, teremos os serviços para somar, subtrair, multiplicar e dividir. A logica é bem simples não é? Agora, se precisasse configurar seu container, gerar seu arquivo WAR para fazer deploy, configurar seus XML's, properties e tudo mais... Acredito que toda essa configuração levaria muito mais tempo do que implementar a própria calculadora. Então vamos ver como fica com Spring Boot.

Vou usar o eclipse para construir o projeto, mas podem usar a IDE que melhor preferirem.

Vamos começar criando um projeto Maven simples, e então vamos mexer no arquivo pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLoc
<modelVersion>4.0.0</modelVersion>
<groupId>com.lucas.spring.calculadora</groupId>
<artifactId>CalcRest</artifactId>
<version>0.0.1-SNAPSHOT</version>

<properties>
<java.version>1.8</java.version>
</properties>

<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>1.4.1.RELEASE</version>
</parent>

<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>
```

vejam que adicionei o parent para o spring Boot:

```
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>1.4.1.RELEASE</version>
</parent>
```

E a dependência spring-boot-starter-web do Spring Boot

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

Essa é a dependência que contem tudo o que precisamos para iniciar nosso projeto Web.

OBS: Coloquei a versão 8 do Java para nosso projeto e a tag build com o plugin do Spring Boot para o maven.

Vamos criar 2 pacotes em nosso projeto, um para conter nossa Main Application e outro para conter nosso Controller onde iremos mapear os serviços da nossa calculadora. No meu caso criei o pacote com.lucas.spring.calculadora e com.lucas.spring.calculadora.controller

Dentro do pacote com.lucas.spring.calculadora vamos criar a classe Application

```
package com.lucas.spring.calculadora;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;

@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        ApplicationContext ctx = SpringApplication.run(Application.class, args)
    }

}
```

Repare que é uma simples classe main Java, e não precisamos mais do que isso para disponibilizar nossa aplicação.

A anotação @SpringBootApplication configura algumas coisas defaults do Spring para nós, ela inclui automaticamente as anotações:

@Configuration – classifica a classe como uma bean de configurações para a aplicação.

@EnableAutoConfiguration – habilita a auto configuração do Spring baseado em suas convenções.

@ComponentScan – Busca componentes, serviços, configurações que estão em nosso pacote(ou sub-pacote em nosso caso)

@EnableWebMvc – seta nossa aplicação como uma Web Application para o Spring

Por fim com uma única linha de código damos um “RUN” em nossa própria classe. Basta rodar nossa classe Main como uma aplicação JAVA comum, e por default o spring Boot vai iniciar o Tomcat Embedded na porta 8080 (é muito rápido, acredite!).

No pacote com.lucas.spring.calculadora.controller vamos criar a classe CalcController

```
package com.lucas.spring.calculadora.controller;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/calculadora")
public class CalcController {

    @RequestMapping(value = "/soma", method = RequestMethod.GET)
    public Double soma(@RequestParam("valA") Double valA, @RequestParam("valB") Double valB) {
        return valA + valB;
    }

    @RequestMapping(value = "/subtrai", method = RequestMethod.GET)
    public Double subtrai(@RequestParam("valA") Double valA, @RequestParam("valB") Double valB) {
        return valA - valB;
    }

    @RequestMapping(value = "/multiplica", method = RequestMethod.GET)
    public Double multiplica(@RequestParam("valA") Double valA, @RequestParam("valB") Double valB) {
        return valA * valB;
    }

    @RequestMapping(value = "/divide", method = RequestMethod.GET)
    public Double divide(@RequestParam("valA") Double valA, @RequestParam("valB") Double valB) {
        return valA / valB;
    }

}
```

Aqui implementamos a logica da nossa calculadora e mapeamos as requisições, poderíamos dividir entre regras de negócio e mapeamento do nosso resource mas não é nosso foco.

A anotação `@RestController` habilita minha classe como controller e faz com que cada método seja anotado como `@ResponseBody` (Podemos falar melhor dessas anotações em outro artigo mais detalhado, por enquanto o importante é saber que essa anotação é responsável por habilitar o mapeamento de nossos métodos).

A anotação `@RequestMapping` mapeia literalmente nosso método, passamos o nome com que ele sera chamado e qual método de request ele é (GET, POST...), no nosso caso mapeei todos os métodos como GET.

Por fim a anotação `@RequestParam` apenas mapeia os parâmetros que quero receber na requisição(valor A e valor B) para efetuar minhas operações(Podemos detalhar melhor essa anotação em outro artigo também).

Pronto, Calculadora Construída! Vamos Subir E Testar ?

Para testar vamos executar por linha de comando(no meu caso estou usando linux). Para isso, precisamos estar com o JAVA 8 e o Maven instalado.

Na pasta onde se encontra o projeto vamos executar o comando: `mvn clean install`

Se tudo ocorrer com sucesso será gerado o arquivo `CalcRest-0.0.1-SNAPSHOT.jar` na pasta `target`, é esse jar que iremos usar. Ainda na pasta do seu projeto entre com o comando `java -jar target/CalcRest-0.0.1-SNAPSHOT.jar`

O Spring Boot vai iniciar nossa aplicação com o Tomcat Embedded na porta 8080, feito isso teremos nossos calculadora mapeada, basta acessar pelo browser mesmo e nossas operações serão executadas, exemplo:

- `http://localhost:8080/calculadora/soma?valA=2&valB=3`
- `http://localhost:8080/calculadora/subtrai?valA=2&valB=3`
- `http://localhost:8080/calculadora/multiplica?valA=2&valB=3`
- `http://localhost:8080/calculadora/divide?valA=2&valB=3`

### Começando Com A Magia Do Spring Boot

O primeiro passo, como não poderia ser diferente, é a criação do projeto. Você pode realizar este passo no próprio `SetupMyProject`. O zip vai vir com o mínimo configurado, mas é aí que você já vai perceber a diferença. Para conseguir subir o servidor e acessar a primeira rota, só precisamos de uma classe configurada no projeto.

```
@SpringBootApplication
@Controller
public class CasadocodigoSpringBootApplication {

    @RequestMapping("/")
    @ResponseBody
    public String index(){
        return "funciona?";
    }

    public static void main(String[] args) {
        SpringApplication.run(CasadocodigoSpringBootApplication.class, args);
    }
}
```

Aqui já temos um pouco de magia. Você vai rodar sua aplicação web a partir de um bom e velho método `main`. A classe `SpringApplication` vai ler as configurações da classe passada como argumento e pronto, você tem a sua aplicação no ar. A outra parte da magia acontece por conta da anotação `@SpringBootApplication`. Ela é um Sterotype do Spring Boot que já encapsula algumas outras

annotations, como a `@EnableAutoConfiguration`. Essa última, por sua vez, carrega a `AutoConfigurationPackages` que é responsável por configurar os pacotes que devem ser escaneados, baseados na localização da sua classe.

Um outro ponto bem impressionante é o `pom.xml` criado, você vai perceber que ele possui pouquíssimas dependências! Vamos dar uma rápida olhada.

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.3.0.BUILD-SNAPSHOT</version>
</parent>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
</dependencies>
```

Perceba que você adiciona algumas dependências que eles chamam de starters. A grande sacada é fazer com esses artefatos já baixem tudo que você precisa. E aqui, eu preciso dizer que, pelo menos para mim, eles tomaram uma decisão bastante acertada. Ao invés de te dar a opção de escolher entre vários frameworks, servidores web, libs específicas e etc, eles tomaram algumas decisões e empurraram pra gente. Para a maioria dos projetos isso não muda e, se for preciso, você pode sobreescrever tudo que eles fizeram. A motivação deles foi a mesma que tivemos ao construir o `SetupMyProject`. Já estamos cheios de frameworks, precisamos é de uma maneira mais fácil de usá-los.

Por mais que já tenhamos o projeto configurado, provavelmente esse não é bem o modelo que vamos seguir. Os nossos controllers não vão ser declarados dentro da mesma classe de configuração. O normal é criar uma outra classe que representa este seu controller.

```
@Controller
@Transactional
@RequestMapping("/produtos")
public class ProductsController {

    @Autowired
    private ProductDAO products;
    @Autowired
    private FileSaver fileSaver;
    @Autowired
    private ServletContext ctx;
    @Autowired
    private InternalResourceViewResolver resolver;

    @RequestMapping(method=RequestMethod.GET)
    @Cacheable(value="lastProducts")
    public ModelAndView list() throws ServletException, IOException{
        System.out.println("ola");
        ModelAndView modelAndView = new ModelAndView("products/list");
        modelAndView.addObject("products", products.findAll());
        return modelAndView;
    }

    ...
}
```

E agora é só pedir para escanear o pacote... Opa, não precisa mais! Como eu criei esse classe em um pacote abaixo do pacote da classe de configuração, todas as classes já vão ser escaneadas de maneira automática. Agora é necessário configurar o acesso a JPA e, nesse momento, nossos corações vão ficar felizes novamente. O nosso único trabalho é adicionar um starter no `pom.xml`.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

Quase tudo que você acha que deveria fazer para configurar o acesso ao banco com a JPA, vai ser feito automaticamente. Não será necessário configurar `persistence.xml`, `EntityManagerFactory`, `EntityManager`, `PlatformTransactionManager` nem nada, tudo vai ser feito pra você. A única coisa que você precisa entregar a ele é o `DataSource` com os dados de acesso ao banco. Podemos criar um método com essa configuração na própria classe que contém nosso `main`.

```
@SpringBootApplication
public class CasadocodigoSpringbootApplication {
    @Bean
    public DataSource dataSource(Environment environment) {
        DriverManagerDataSource dataSource = new DriverManagerDataSource();
        dataSource.setDriverClassName("com.mysql.jdbc.Driver");
        dataSource.setUrl("jdbc:mysql://localhost:3306/casadocodigo");
        dataSource.setUsername("root");
        dataSource.setPassword("");
        return dataSource;
    }

    public static void main(String[] args) {
        SpringApplication.run(CasadocodigoSpringbootApplication.class, args);
    }
}
```

Além disso, ele também já inclui as dependências para a Spring Data JPA, para facilitar a criação dos seus DAOs. Para completar essa primeira parte da migração do projeto, foi necessário fazer uma pequena configuração extra para os jsps. O Spring Boot sugere que você use outra tecnologia de view, mas como o jsp era a utilizada no projeto do livro, eu resolvi ir com ela mesmo. Como o tomcat está sendo executado de forma `embedded`, é necessário que adicionemos uma nova dependência no pom, que é a do compilador de jsps para este tipo de cenário.

```
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
  <artifactId>tomcat-embed-jasper</artifactId>
  <scope>provided</scope>
</dependency>
```

Além disso, como é procedimento normal, precisamos configurar um prefixo e um sufixo para busca da nossa view. Podemos fazer isso do jeito tradicional, adicionando um método que retorna um `InternalResourceViewResolver`. Só que podemos aproveitar a oportunidade para já apresentar um outro detalhe que você pode tirar proveito, que são os arquivos de configuração externos. Você pode criar um arquivo chamado `application.properties` na raiz do seu classpath, geralmente em `src/main/resources`. Lá você pode colocar configurações específicas das tecnologias utilizadas no projeto, abaixo segue o meu exemplo.

```
spring.mvc.view.prefix=/WEB-INF/jsp/
spring.mvc.view.suffix=.jsp

#hibernate
spring.jpa.hibernate.ddl-auto=update
```

Perceba que ele é bem diferente de um `spring-context.xml` da vida. Você não tem que declarar um bean para ser usado, simplesmente configura as propriedades do bean já escolhido.

Para fechar com chave de ouro, pelo menos para mim :), eles deram mais um passo pensando na vida fácil do usuário. Nessa semana eles lançaram um novo starter chamado de DevTools. Uma facilidade adicionada foi a de recarregamento do projeto para cada mudança que você faz no código. Só que, como eles mesmos disseram, esse recarregamento é mais esperto do que um simples hot deploy de um servidor web tradicional. Além disso, ainda adicionaram o suporte para o Live Reload, que possibilita o recarregamento automático das páginas no seu navegador sempre que você alterar uma classe sua no servidor!



## Web Services

**SOAP (Simple Object Access Protocol**, em português **Protocolo Simples de Acesso a Objetos**) é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída. Ele se baseia na Linguagem de Marcação Extensível (XML) para seu formato de mensagem, e normalmente baseia-se em outros protocolos da camada de aplicação, mais notavelmente em chamada de procedimento remoto (RPC) e Protocolo de transferência de hipertexto (HTTP), para negociação e transmissão de mensagens.

SOAP pode formar a camada base de uma pilha de protocolos de serviços Web, fornecendo um arcabouço básico de mensagens sob o qual se podem construir os serviços Web.

Este protocolo baseado em XML consiste de três partes: um envelope, que define o que está na mensagem e como processá-la, um conjunto de regras codificadas para expressar instâncias do tipos de dados definidos na aplicação e uma convenção para representar chamadas de procedimentos e respostas.

Sua especificação define um arcabouço que provê maneiras para se construir mensagens que podem trafegar através de diversos protocolos e que foi especificado de forma a ser independente de qualquer modelo de programação ou outra implementação específica. Por não se tratar de um protocolo de acesso a objetos, o acrônimo não é mais utilizado.

Geralmente servidores SOAP são implementados utilizando-se servidores HTTP, embora isto não seja uma restrição para funcionamento do protocolo. As mensagens SOAP são documentos XML que aderem a uma especificação W3C.

O primeiro esforço do desenvolvimento do SOAP foi implementar RPCs sobre XML.

### Definição

Envelope das mensagens, regras de codificação, convenção RPC, ligação com protocolos subjacentes.

O SOAP tem:

Mecanismo para definir a unidade de comunicação,

Mecanismo para lidar com erros,

Mecanismo de extensão que permite evolução,

Mecanismo entre as mensagens SOAP e o HTTP, que permite representar tipos de dados em XML.

### Concepção

Simplicidade, independente de vendedor, independente da linguagem, independente do modelo de objetos, independente do transporte.

Introdução às tecnologias Web Services: SOA, SOAP, WSDL e UDDI

Antes de nos aprofundarmos nos conceitos e tecnologia de web services, vejamos um pouco sua evolução. No ano de 2000, a W3C (World Wide Web Consortium) aceitou a submissão do Simple Object Access Protocol (SOAP).

Este formato de mensagens baseado em XML estabeleceu uma estrutura de transmissão para comunicação entre aplicações (ou entre serviços) via HTTP. Sendo uma tecnologia não amarrada a fornecedor, o SOAP disponibilizou uma alternativa atrativa em relação aos protocolos proprietários tradicionais, tais como CORBA e DCOM.

No decorrer do ano seguinte, o W3C publicou a especificação WSDL. Uma nova implementação do XML, este padrão forneceu uma linguagem para descrever a interface dos web services. Posteriormente suplementada pela especificação UDDI (Universal Description, Discovery and Integration), que proporcionou um mecanismo padrão para a descoberta dinâmica (dynamic discovering) de descrições



de serviço, a primeira geração da plataforma de Web services foi estabelecida. A **Figura 1** ilustra em alto nível o relacionamento entre estes padrões.

**Figura 1:** O relacionamento entre especificações de primeira geração

Is accessed using: é acessado utilizando;

Enables discovery of: permite a descoberta de;

Describes: descreve;

Enables communication between: permite a comunicação entre;

Binds to: ligação para.

Desde então, os web services foram adotados por vendedores e fabricantes num ritmo considerável. Suporte amplo da indústria seguiu-se à popularidade e importância desta plataforma e de princípios de projeto orientados a serviço. Isto levou à criação de uma segunda geração de especificação de Web services.

### **Web services e a arquitetura orientada a serviços (SOA)**

#### **Entendendo serviços**

O conceito de serviços em uma aplicação existe faz algum tempo. Serviços, assim como componentes, são considerados blocos de construção independentes, os quais coletivamente representam um ambiente de aplicação.

No entanto, diferente de componentes tradicionais, serviços têm algumas características únicas que lhes permitem participar como parte de uma arquitetura orientada a serviços.

Uma destas características é a completa autonomia em relação a outros serviços. Isto significa que cada serviço é responsável por seu próprio domínio, o que tipicamente significa limitar seu alcance para uma função de negócio específica (ou um grupo de funções relacionadas).

Este enfoque de projeto resulta na criação de unidades isoladas de funcionalidades de negócio ligadas fracamente entre si. Isto é possível por causa da definição de uma estrutura padrão de comunicação.

Devido à independência que esses serviços desfrutam dentro desta estrutura, a lógica de programação que encapsulam não tem necessidade de obedecer a nenhuma outra plataforma ou conjunto de tecnologias.

#### **XML Web services**

O tipo de serviço mais largamente aceito e bem-sucedido é o XML Web service, que será daqui em diante chamado apenas de web service, ou simplesmente service. Este tipo de serviço possui dois requisitos fundamentais:

Comunica-se via protocolos internet (normalmente HTTP);

Envia e recebe dados formatados como documentos XML.

A ampla aceitação do web service resultou no surgimento de um conjunto de tecnologias suplementares que se tornaram um padrão de fato. Assim ao desenvolver nossa web services devemos considerar o uso de tecnologias que:

Forneça uma descrição de serviço que, no mínimo, consista de um documento WSDL;

Seja capaz de transportar documentos XML utilizando SOAP sobre HTTP.

Estas tecnologias não modificam a funcionalidade do núcleo de um serviço web, tanto como o faz sua habilidade para se representar e comunicar num modo padrão.

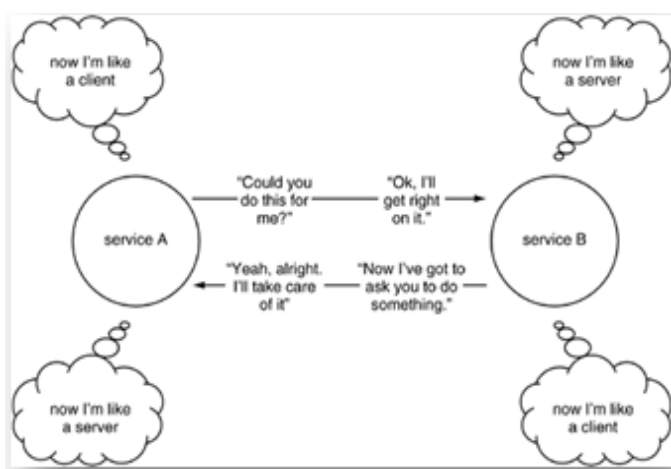
Muitas das convenções de arquitetura expressadas neste artigo assumem que SOAP e WSDL fazem parte da estrutura de web services descrita.

Além disto, é normal que um Web service seja:

Capaz de agir como o solicitante e o provedor de um serviço;

Registrado com um discovery agent através do qual possam ser localizados.

Numa conversação típica com um web service, o cliente iniciador do pedido é um web service também. Como mostrado na **Figura 2**, qualquer interface exposta por este client service também o qualifica como um serviço a partir do qual outros serviços podem solicitar informação. Posto isto, web services não se encaixam no modelo clássico de cliente-servidor. Na verdade, eles tendem a estabelecer um sistema ponto-a-ponto, onde cada serviço pode atuar como cliente ou servidor.



**Figura 2:** Troca de papéis do Web services durante uma conversação

Now I'm like a client: agora sou um cliente;

Now I'm like a server: agora sou um servidor;

Service: serviços;

Could you do this for me: poderia fazer isso por mim?;

Ok, I'll get right on it: Ok;

Now I've got to ask you to do something: agora eu gostaria de lhe pedir algo;

Yeah, alright. I'll take care of it: Tudo bem, é só falar.

### **Service-oriented architecture (SOA)**

Como mencionado anteriormente, adicionar uma aplicação com uns poucos web services não é nenhum problema.

Esta integração limitada pode ser apropriada para uma experiência de aprendizado, ou para complementar a arquitetura de uma aplicação existente com uma peça de funcionalidade baseada em serviços que atende a um requisito específico do projeto. No entanto, isto não estabelece uma arquitetura orientada a serviço. Existe uma clara diferença entre:

Uma aplicação que usa web service;

Uma aplicação baseada numa arquitetura orientada a serviços.

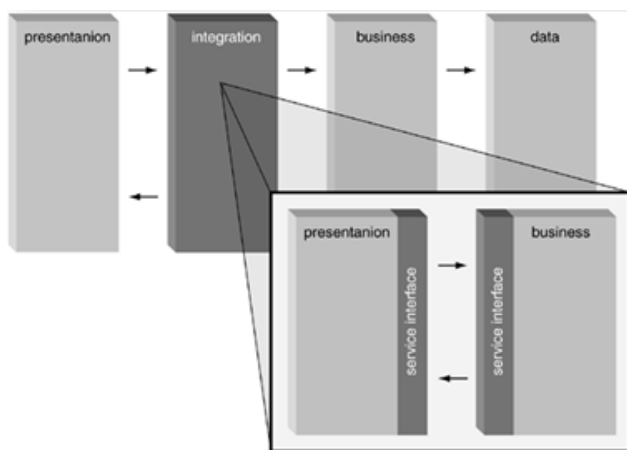
Uma SOA é um modelo de projeto com um conceito profundamente amarrado à questão do encapsulamento de aplicação.

A arquitetura resultante estabelece essencialmente um paradigma de projeto, no qual web services são os blocos de construção chave. Isto quer dizer que ao migrar a arquitetura da sua aplicação para uma SOA, estabelece-se um compromisso com os princípios de projeto de web services e a tecnologia correspondente, como partes fundamentais do seu ambiente técnico.

Uma SOA baseada em XML web service é construída sobre camadas de tecnologia XML estabelecidas, focada em expor a lógica de aplicação existente como um serviço fracamente acoplado. Para apoiar este modelo, uma SOA promove o uso de um mecanismo de discovery por serviços via um service broker ou discovery agent.

A **Figura 3** mostra como uma SOA altera a arquitetura multicamada existente, ao introduzir uma camada lógica que, através do uso de interfaces programáticas padrão (providas pelo web services), estabelece um ponto comum de integração.

Esta camada de integração de serviços constitui a base para um novo modelo que pode se estender além do escopo de uma única aplicação, unificando plataformas legadas disparees em um ambiente aberto. Quando web services são utilizados para integração cruzada de aplicações (ver **Figura 4**), elas se estabelecem como parte da infra-estrutura do sistema.



**Figura 3:** Uma representação lógica de uma arquitetura orientada a serviços

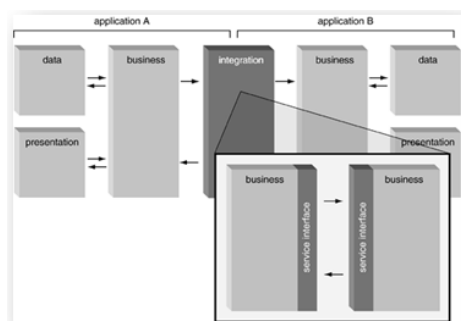
Presentation: apresentação;

Integration: integração;

Business: negócio;

Data: dado;

Service interface: interface de serviço.



**Figura 4:** Uma representação lógica de uma arquitetura de integração orientada a serviço

É importante se conscientizar quanto ao acréscimo de complexidade de projeto introduzido pelo SOA. Mais ainda do que em um ambiente n-camada, projetistas de aplicação devem considerar de uma forma completa como a introdução de serviços vai afetar dados existentes e modelos de negócio.

Na medida em que a utilização de serviços se diversifica, o significado dos requisitos de segurança e escalabilidade são amplificados. Ambientes orientados a serviço bem projetados tentarão vencer estes desafios com infra-estrutura adequada, ao invés de utilizar soluções sob medida, específicas de aplicação

Os papéis e cenários ilustrados nas próximas duas seções estão limitados somente ao assunto web service. A estrutura de mensagens SOAP subjacente será explicada em separado, no próximo artigo desta série.

### **Papéis Web Service**

Serviços podem assumir diferentes papéis quando envolvidos em diversos cenários de interação. Dependendo do contexto pelo qual é visualizado, assim como o estado da tarefa rodando no momento, o mesmo web service pode trocar de papéis ou ser designado para múltiplos papéis simultâneos:

#### **Provedor de Serviços**

Agindo como um provedor de serviços, um web service expõe uma interface pública através da qual pode ser chamado por solicitantes do serviço. Um provedor de serviços disponibiliza esta interface publicando uma descrição do serviço. Num modelo cliente-servidor, o provedor de serviço pode ser comparado ao servidor.

O termo “provedor de serviço” pode também ser usado para descrever a organização ou ambiente que hospeda (provê) o web service.

Um provedor de serviço pode também agir como um solicitante de serviço. Por exemplo, um web service pode atuar como um provedor de serviço quando um solicitante de serviço lhe pede para executar uma função.

Pode então atuar como um solicitante de serviço quando mais tarde contata o solicitante de serviço original (agora agindo como um provedor de serviço) para solicitar informação de status.

#### **Solicitante de Serviço**

Um solicitante de serviço é o remetente de uma mensagem web service ou o programa de software solicitando uma web service específico. O solicitante de serviço é comparável ao cliente dentro de um modelo cliente-servidor padrão. Solicitantes de serviços são às vezes chamados de consumidores de serviços.

Um solicitante de serviço pode também ser um provedor de serviço. Por exemplo, num modelo de solicitação e resposta, a web service iniciador primeiro age como um solicitante de serviço ao requerer informações do provedor de serviço.

A mesma web service então, faz o papel de um provedor de serviço ao responder à solicitação original.

#### **Intermediário**

O papel de intermediário é assumido pelo web service quando ele recebe a mensagem de um solicitante de serviço e a passa adiante para o provedor de serviço. Neste caso, ele pode também agir como um provedor de serviço (recebendo a mensagem) e como um solicitante de serviço (passando adiante a mensagem).

Intermediários podem existir em muitas formas diferentes. Alguns são passivos e simplesmente retransmitem ou roteam as mensagens, enquanto outros processam ativamente uma mensagem antes de repassá-la.

Tipicamente, aos intermediários só é permitido o processamento e modificação do cabeçalho da mensagem. Para preservar a integridade da mensagem, seus dados não devem ser alterados.

**Remetente Inicial**

Como o web service responsável por iniciar a transmissão da mensagem, remetentes iniciais também podem ser considerados solicitantes de serviço. Este termo existe para ajudar a diferenciar o primeiro web service que envia uma mensagem, dos intermediários também qualificados como solicitantes de serviço.

**Receptor Final**

O último Web service a receber uma mensagem é o receptor final. Estes serviços representam o destino final de uma mensagem e também podem ser considerados provedores de serviço.

**Interação Web Service**

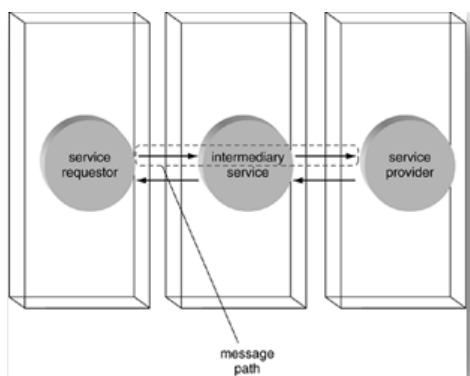
Quando mensagens são passadas entre dois ou mais web services, uma variedade de cenários de interação pode acontecer. A seguir, termos comuns utilizados para identificar e etiquetar estes cenários serão apresentados.

**Caminho da Mensagem**

A rota pela qual a mensagem viaja é o caminho da mensagem. Deve consistir de um remetente inicial e um receptor final e pode conter nenhum, um, ou mais de um intermediários. A **Figura 5** ilustra um caminho de mensagem simples.

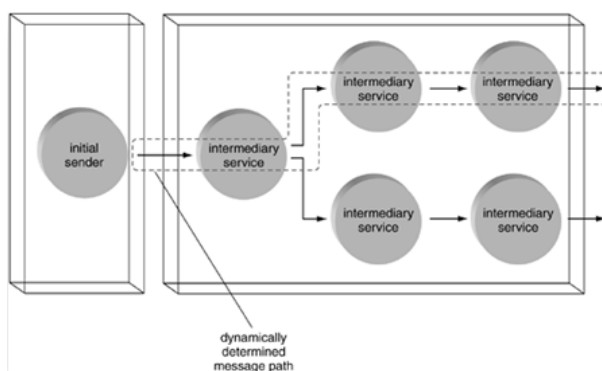
O caminho de transmissão atual percorrido por uma mensagem pode ser dinamicamente determinado por roteadores intermediários.

A lógica de roteamento pode ser ativada em resposta à carga de requisitos de balanceamento, ou pode ser baseada nas características da mensagem e outras variáveis lidas e processadas pelo intermediário em tempo de execução. A **Figura 6** descreve como uma mensagem é enviada via um ou dois caminhos de mensagem possíveis, tal como é determinado pelo roteador intermediário.



**Figura 5:** Um caminho de mensagem formado por três Web services

Message path: caminho da mensagem.



**Figura 6:** Uma mensagem dinamicamente determinada por um roteador intermediário

Dynamically determined message path: caminho da mensagem determinada dinamicamente.

### Padrão de troca de mensagens

Serviços que interagem em um ambiente orientado a serviços tipicamente se enquadram em determinados padrões de troca de mensagens. Padrões típicos incluem:

Solicite e responda (request and response);

Publique e subscreva (publish and subscribe);

fire and forget - um para um;

fire and forget - um para muitos ou difusão;

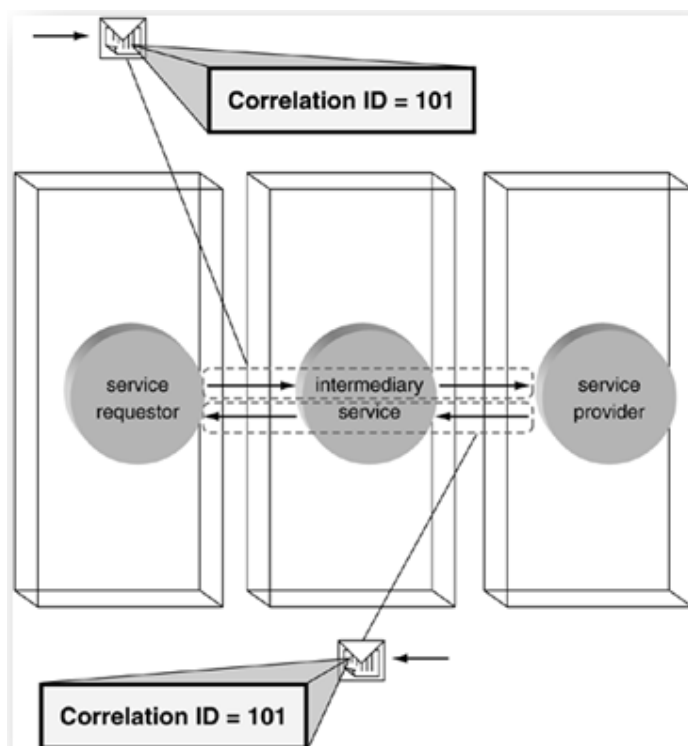
O padrão pedido e resposta é o mais comum quando se está simulando intercâmbio de dados sincronizados. Os demais padrões são usados principalmente para facilitar transferência de dados assíncrona.

### Correlação

Correlação (Correlation) é a técnica utilizada para casar mensagens enviadas através de caminhos de mensagem diferentes. É comumente empregada num padrão de intercâmbio de pedido e resposta de mensagem, onde a mensagem de resposta deve estar associada à mensagem original que iniciou a solicitação. Embutir valores de ID sincronizados dentro de mensagens relacionadas é uma técnica frequentemente utilizada para conseguir correlação.

### Coreografia

Regras que governam características de comportamento relacionadas à forma como um grupo de web services interagem podem ser aplicadas como uma coreografia (choreography). Estas regras incluem a seqüência na qual web services podem ser chamados, condições que se aplicam à seqüência que está sendo transportada e o padrão de uso que irá definir os cenários de interação permitidos. O escopo de uma coreografia está tipicamente amarrado ao de uma atividade ou tarefa (ver exemplo na **Figura 7**).



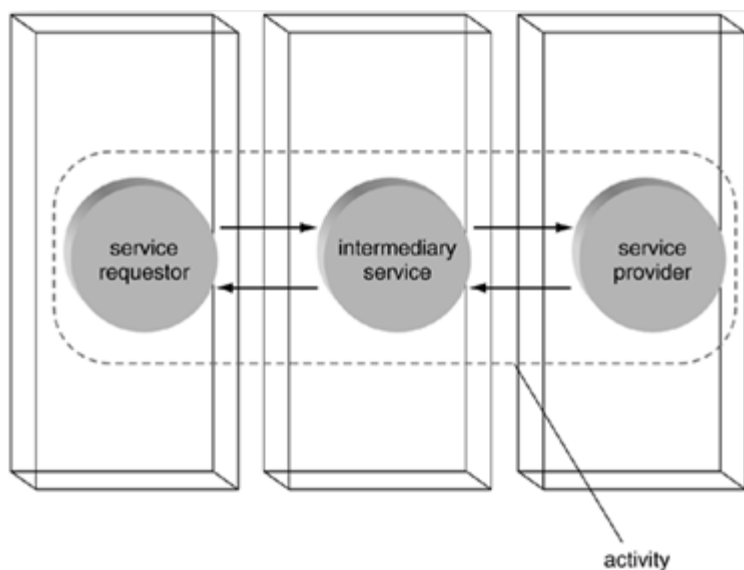
**Figura 7:** A seqüência de interação de um grupo de serviços sendo comandados por uma coreografia

Correlation: correlação.

**Atividade**

Padrões de intercâmbio de mensagens formam a base para atividades de serviços (também conhecidos como tarefas). Uma atividade consiste de um grupo de web services que interagem e colaboram para realizar uma função ou um grupo lógico de funções.

A **Figura 8** mostra uma atividade de serviço simples. A diferença entre uma coreografia e uma atividade está no fato de que a atividade é geralmente associada com uma função de aplicação específica, tal como a execução de uma tarefa de negócio.



**Figura 8:** Uma atividade de serviço envolvendo três serviços

Activity: atividade.

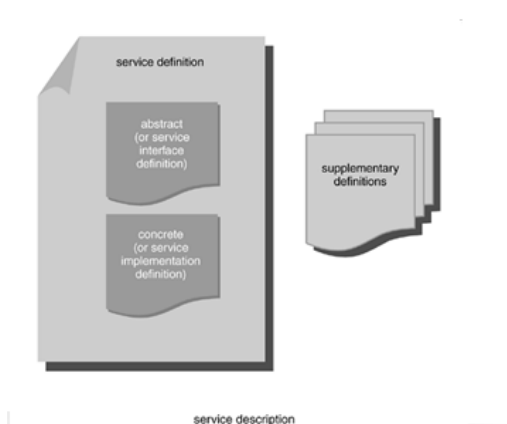
**Descrição da estrutura de web services**

Um web service é descrito através de uma coleção de documentos de definição. Estes atuam como blocos de construção para uma descrição de serviço:

Abstrato + Concreto = Definição de Serviço;

Definição de Serviço + Definições Complementares = Descrição de Serviço.

A **Figura 9** ilustra a relação entre esses documentos.



**Figura 9:** Conteúdo de uma descrição de serviço

Service description: descrição do serviço;

Service definition: definição do serviço;

Supplementary definitions: definições suplementares;

Abstract: abstrato;

Concret: concreto;

service interface definition: definição da interface do serviço;

service implementation definition: definição da implementação do serviço.

### **Abstrato**

A descrição de uma interface web service, independente dos detalhes de implementação, é chamada de abstrato (abstract). Descrições deste elemento são fornecidas adiante neste artigo, como parte do tutorial do WSDL.

### **Concreto**

Localização e informação de implementação específicas sobre um web service constituem as partes concretas (concrete) de um documento WSDL. Elas são representadas pelos elementos de ligação (binding), serviço (service) e ponto-de-término (endpoint ou port).

### **Definição de serviço**

Geralmente, o conteúdo de um documento WSDL constitui uma definição de serviço (service definition) que inclui as definições da interface (abstrato) e da implementação (concreto).

### **Descrição de Serviço**

Frequentemente, uma descrição de serviço é um único documento WSDL que fornece uma definição de serviço. No entanto, pode também incluir vários documentos de definição adicionais que irão fornecer informações complementares.

### **Introdução Aos Web Services de Primeira Geração**

A estrutura W3C para web services está fundamentada em três especificações XML fundamentais:

Linguagem para definição de web service (Web Services Definition Language - WSDL);

Simple Object Access Protocol (SOAP);

Universal Description, Discovery, and Integration (UDDI).

Estes padrões de tecnologia, acoplados aos princípios de projeto orientado a serviço, formam um SOA fundamentado na tecnologia XML. Esta arquitetura de web services de primeira geração permite a criação de web services independentes capazes de encapsular unidades isoladas de funcionalidades de negócio.

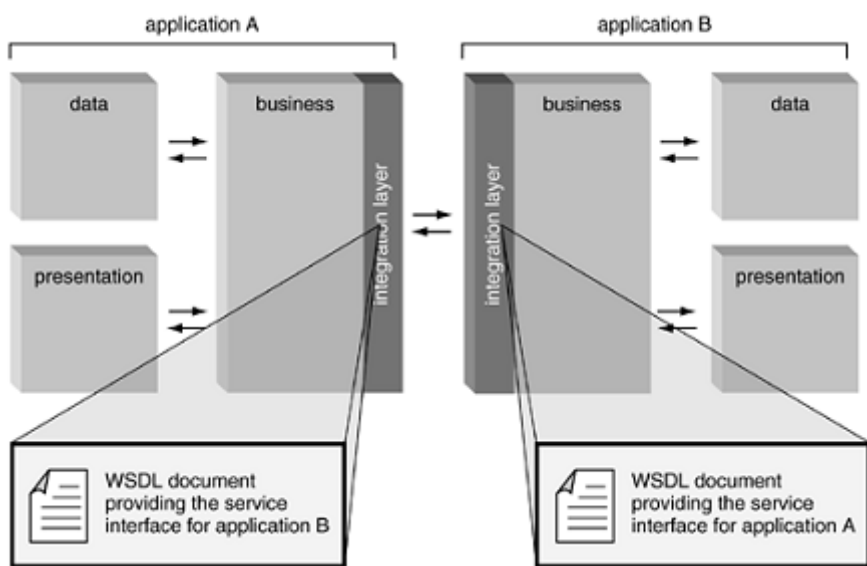
Esta tecnologia tem também algumas limitações, que tem sido contempladas numa segunda geração de especificações. A seção a seguir, fornece um tutorial introdutório para a tecnologia WSDL. SOAP e UDDI serão vistos no próximo artigo da série.

### **Linguagem para definição de Web Services (WSDL)**

Web services devem ser definidos numa forma consistente para que possam ser descobertos e “interfaceados” com outros serviços e aplicações. A WSDL é uma especificação W3C que fornece a linguagem mais avançada para a descrição de definições de web services.



A camada de integração introduzida pela estrutura de web services estabelece um padrão, universalmente reconhecido e com interface programática suportada. Tal como mostrado na **Figura 10**, WSDL permite a comunicação entre essas camadas ao fornecer descrições padronizadas.



**Figura 10:** Documentos WSDL representando aplicações web services

Application: aplicação;

Integration layer: camada de integração;

WSDL document providing the service interface for application b: documento WSDLI provendo a interface para o serviço da aplicação b;

WSDL document providing the service interface for application a: documento WSDL provendo a interface para o serviço da aplicação a.

A melhor forma de entender como é definido um web service e como ele é expresso por um documento WSDL, é caminhar através de cada construtor que coletivamente representa essa definição. Começemos com o elemento definitions raiz, o qual age como o container para a definição do serviço (ver **Listagem 1**).

**Listagem 1:** Uma definição de serviço, tal como é expressa pelo construtor definitions

```
<definitions>
  <interface name="Catalog">
    ...
  </interface>
  <message name="BookInfo">
    ...
  </message>
  <service>
    ...
  </service>
```

```
<binding name="Binding1">
```

```
...
```

```
</binding>
```

```
</definitions>
```

Uma definição WSDL pode conter coleções dos seguintes construtores primários:

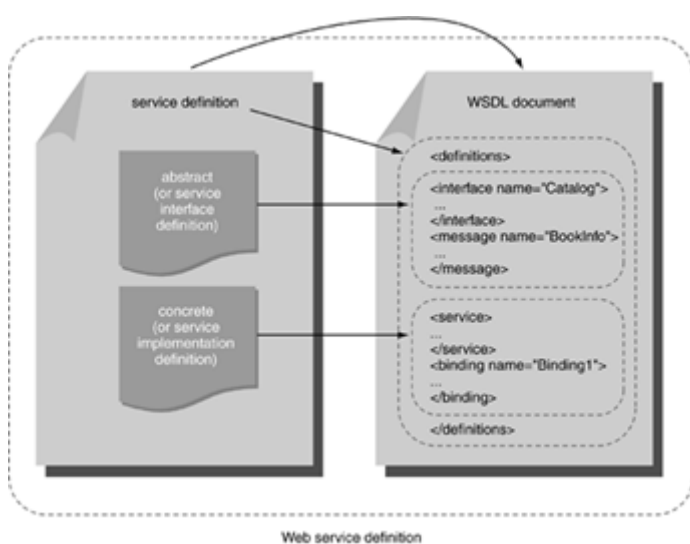
Interface;

Message;

Service;

Binding.

A **Figura 11** ilustra como os primeiros dois construtores representam a definição da interface de serviço e os últimos dois fornecem os detalhes de implementação do serviço.



**Figura 11:** O conteúdo de um documento WSDL, tal como se relaciona com uma definição de serviço

Web service definition: definição do web service;

WSDL document: documento WSDL.

### Definição de interface abstrata

Interfaces de web services individuais são representadas por elementos interface WSDL. Estes construtores contêm um grupo de operações lógicas correlatas. Numa arquitetura baseada em componentes, um elemento interface WSDL é análogo à interface do componente. Uma operação, portanto, é equivalente a um método de componente, por representar uma única ação ou função. A **Listagem 2** apresenta um exemplo de interface.

**Listagem 2:** Uma interface representada pelo elemento interface

```
<definitions>
```

```
<interface name="Catalog">
```

```
<operation name="GetBook">
```

```
...
```

```

        </operation>
    </interface>
</definitions>

```

Um elemento operation típico consiste de um grupo de mensagens de entrada e saída correlatas. A execução de uma operation requer a transmissão ou intercâmbio destas mensagens entre o serviço solicitante e o provedor de serviço.

Mensagens operation são representadas por construtores message que são declarados sob os elementos definitions. Os nomes das mensagens são então referenciados nos elementos filho das operation input ou output (ver **Listagem 3**).

**Listagem 3:** O elemento input dentro do construtor operation referenciando um bloco de mensagem

```

<definitions>
    <message name="BookInfo">
        ...
    </message>
    <interface name="Catalog">
        <operation name="GetBook">
            <input name="Msg1" message="BookInfo"/>
        </operation>
    </interface>
</definitions>

```

Um elemento message pode conter um ou mais parâmetros input ou output que pertencem a uma operation. Cada elemento part define um destes parâmetros. Ele fornece um conjunto nome/valor (name/value), junto com o tipo de dado associado. Em uma arquitetura baseada em componentes, um part WSDL equivale a um parâmetro de input ou output (ou um valor de retorno) de um método de componente (ver **Listagem 4**).

**Listagem 4:** Um bloco de mensagem com um construtor part representando parâmetros operation

```

<definitions>
    <message name="BookInfo">
        <part name="title" type="xs:string">
            Field Guide
        </part>
        <part name="author" type="xs:string">
            Mr. T
        </part>
    </message>
</definitions>

```

A seguir, um breve resumo dos construtores fundamentais que podem ser montados para estabelecer uma definição de interface abstrata:

Interfaces: representam interfaces de serviço e podem conter múltiplos operations;

Operations: representam uma função web service e podem referenciar múltiplas messages;

Messages: representam uma coleção de parâmetros input ou output e podem conter múltiplas parts;

Parts: representam parâmetros de dados operation tanto de chegada como de partida.

### Definição concreta (implementação)

Sobre os detalhes de implementação, usando os elementos descritos nesta seção, um documento WSDL pode estabelecer detalhes "concrete" de ligação para protocolos, tais como SOAP e HTTP.

Dentro de um documento WSDL, o elemento service representa um ou mais pontos-de-término nos quais o web service pode ser acessado.

Estes pontos-de-término consistem de informações de localização e protocolo e são armazenados numa coleção de elementos endpoint (ver **Listagem 5**).

**Listagem 5:** O elemento ponto-de-término

```
<definitions>
  <service name="Service1">
    <endpoint name="EndPoint1" binding="Binding1">
      ...concrete implementation details...
    </endpoint>
  </service>
</definitions>
```

Agora que descrevemos como um web service pode ser acessado, precisamos definir os requisitos de chamada para cada uma das suas operations.

Os elementos binding associam às operations informações de formato de protocolo e mensagem. O construtor operation que reside dentro do bloco binding é semelhante a sua contrapartida na seção interface (ver **Listagem 6**).

**Listagem 6:** O elemento binding representando uma operation existente

```
<definitions>
  <service name="Service1">
    <binding name="Binding1">
      <operation>
        <input name="Msg1" message="book"/>
      </operation>
    </binding>
  </service>
</definitions>
```

A descrição de informação concreta dentro de um documento WSDL pode ser resumida assim:

elementos service: hospedam coleções de ponto-de-término representados individualmente por elementos endpoint;

elementos endpoint: contem dados endpoint, incluindo endereço físico e informação de protocolo;

elementos binding: se auto-associam a construtores operation;

cada endpoint pode referenciar um elemento binding e, portanto, fornecer informação endpoint para a operation subordinada.

Vimos nesse primeiro artigo o princípio de funcionamento dos web services bem como duas de suas tecnologias de base, SOA e WSDL. No próximo artigo, complementaremos o assunto apresentado aqui analisando as tecnologias SOAP e UDDI. Até lá

### **REST e SOAP: Usar um dos dois ou ambos?**

Desenvolvedores web têm uma grande quantidade de tecnologias que podem escolher, de ferramentas para acesso simples a bancos de dados, integração com serviços em middleware, a softwares do lado do cliente.

A quantidade de opções em si já é um desafio, e escolher uma abordagem específica para construir partes de uma applicativa web exacerba o problema.

Neste breve artigo, vamos nos concentrar em uma dessas escolhas: SOAP ou REST. Ambas possuem vantagens e desvantagens e fica na mão do desenvolvedor determinar a melhor abordagem para cada caso em particular.

A maioria dos desenvolvedores tem exposto seus serviços utilizando REST, que faz uso de um padrão de URI (Uniform Resource Identifier), fazendo uma chamada para um serviço web como em:

O REST é simples de entender e pode ser adotado em praticamente qualquer cliente ou servidor com suporte a HTTP/HTTPS. Os desenvolvedores que o utilizam citam, como principais vantagens a facilidade no desenvolvimento, o aproveitamento da infraestrutura web existente e um esforço de aprendizado pequeno.

Por outro lado, o SOAP, avô das interfaces de serviços web, não deixará de ser usado tão cedo. Com o SOAP v 1.2, muitas das deficiências percebidas nessa tecnologia foram corrigidas e aumentou a facilidade de uso.

Além disso, a sigla SOAP deixou de representar "Simple Object Access Protocol". Na especificação 1.2 da W3C, SOAP é apenas o nome da especificação.

Utilizar o SOAP 1.2 traz uma carga adicional não encontrada ao usar REST, mas há também vantagens.

Primeiramente o SOAP é baseado em XML, de três formas: o envelope, que define o conteúdo da mensagem e informa como processá-la; um conjunto de regras de codificação para os tipos de dados; e o layout para os procedimentos de chamadas e respostas.

Esse "envelope" é enviado por meio de (por exemplo) HTTP/HTTPS. E uma RPC (Remote Procedure Call) é executada, e o envelope retorna com as informações do documento XML formatado.

Uma das vantagens do SOAP é o uso de um método de transporte "genérico". Enquanto que o REST faz uso de HTTP/HTTPS, o SOAP pode usar qualquer meio de transporte existente para enviar sua requisição, desde SMTP até mesmo JMS (Java Messaging Service).

No entanto, uma desvantagem percebida no uso de XML é a sua natureza prolixa e o tempo necessário para analisar o resultado apresentado.

A boa notícia para os desenvolvedores web é que ambas as tecnologias são muito viáveis no mercado atual. Ambos REST e o SOAP conseguem resolver um grande número de problemas e desafios

na web, e em muitos casos tanto um como o outro podem ser utilizados para fazer o que querem os desenvolvedores.

Mas uma história não contada é que ambas as tecnologias podem ser misturadas e combinadas. O REST é fácil de entender e extremamente acessível, porém faltam padrões, e a tecnologia é considerada apenas uma abordagem arquitetural. Em comparação, o SOAP é um padrão da indústria, com protocolos bem definidos e um conjunto de regras bem estabelecidas.

Pode-se afirmar, então, que casos onde o REST funciona bem são:

Situações em que há limitação de recursos e de largura de banda: A estrutura de retorno é em qualquer formato definido pelo desenvolvedor e qualquer navegador pode ser usado. Isso porque a abordagem REST usa o padrão de chamadas GET, PUT, POST e DELETE. O REST também pode usar objetos XMLHttpRequest (a base do velho AJAX) que a maioria dos navegadores modernos suporta.

Operações totalmente sem-estado: se uma operação precisa ser continuada, o REST não será a melhor opção. No entanto, se forem necessárias operações de CRUD stateless (Criar, Ler, Atualizar e Excluir), o REST seria a melhor alternativa.

Situações que exigem cache: se a informação pode ser armazenada em cache, devido à natureza da operação stateless do REST, esse seria um cenário adequado para a tecnologia.

Essas três situações abrangem muitas soluções. Então por que ainda precisamos considerar o uso do SOAP? Mais uma vez, o SOAP é bastante maduro e bem definido e vem com uma especificação completa. Já a abordagem REST é apenas isso: uma abordagem. Está totalmente aberta. Por isso ao se encontrar uma das situações abaixo, o SOAP pode ser uma ótima solução:

Processamento e chamada assíncronos: se o aplicativo precisa de um nível garantido de confiabilidade e segurança para a troca de mensagens, então o SOAP 1.2 oferece padrões adicionais para esse tipo de operação como por exemplo o WSRM (WS-Reliable Messaging).

Contratos formais: se ambos os lados (fornecedor e consumidor) têm que concordar com o formato de intercâmbio de dados, então o SOAP 1.2 fornece especificações rígidas para esse tipo de interação.

Operações stateful: para o caso de o aplicativo precisar de informação contextual e gerenciamento de estado com coordenação e segurança, o SOAP 1.2 possui uma especificação adicional em sua estrutura que apoia essa necessidade (segurança, transações, coordenação etc.). Comparativamente, usar o REST exigiria que os desenvolvedores construíssem uma solução personalizada.

Como se vê, cada uma das abordagens tem sua utilidade. Ambas têm problemas nos quesitos de segurança, camadas de transporte etc.; mas ambas podem realizar o trabalho necessário e trazem sua contribuição para o desenvolvimento de aplicações web.

Portanto, a melhor abordagem é a flexibilidade, pois não importa qual seja o problema, no mundo de hoje do desenvolvimento web, conta-se com excelentes resultados ao fazer uso de um desses padrões.

Qual a diferença entre REST e SOAP?

Bom, vamos lá.. Quem é que nunca leu algum artigo, presenciou ou de fato iniciou uma discussão sobre REST e SOAP? Acho que muitos... certo?!

Então... Logo após de ler o título deste post você deve estar se perguntando “Esse cara vai ser mais um desses que fica defendendo um lado e detonando o outro com argumentos esquerdistas e radicais?” De ante mão te respondo:

“Não!”. Minha intenção com este post é apenas explicar um pouco sobre o assunto e tentar desfazer algumas confusões que as pessoas fazem com REST, SOAP e outros assuntos adjacentes.

Antes de mais nada, vamos a definição básica destes dois:

**REST – Representational State Transfer** é um estilo arquitetural usado no projeto de aplicações da Web que contam com recursos nomeados (URL, URI, URN) e engenhosamente utiliza mais profundamente o protocolo HTTP, seu cabeçalho, seus métodos (GET, POST, PUT, DELETE, HEAD) e toda a infraestrutura web já bem estabelecida, reconhecida e utilizada por todos.

**SOAP – Simple Object Access Protocol** é um protocolo para troca de informações estruturadas geralmente em uma plataforma descentralizada e distribuída. Ele se baseia em XML para seu formato de mensagem, ou seja, uma mensagem SOAP encapsula o conteúdo e pode ser trafegada via HTTP, JMS ou outro protocolo.

### Definição de Interfaces

Primeiramente, pensando em serviços que utilizam mensagens no protocolo SOAP, o mais comum é descrevermos a interface do mesmo com WSDL (Web Services Description Language). Basicamente, cada serviço terá um arquivo .wsdl que terá a definição de suas operações, estrutura de dados que são usadas nas requisições e respostas, Endpoints (endereços de rede do serviço).

Uma associação que ajuda o entendimento deste ponto é pensarmos nas operações como métodos de uma classe, a assinatura do método como as mensagens definidas, e o tipo de cada campo da assinatura a definição da estrutura de dados que será usada nas mensagens.

Agora, nos serviços RESTful, se seguirmos de fato o estilo, parte da descrição da interface é desnecessária já que a forma de interagir com os serviços é sempre a mesma, por meio dos métodos http.

E ainda mais com o uso do Método Option podemos saber quais outros métodos são permitidos naquele serviço. Agora com relação aos dados continua a ser interessante a definição de que dados vão trafegar e suas restrições.

Agora, tem pessoas que mesmo assim preferem de uma definição tanto dados quanto de operações, e estas podem fazer o seguinte:

a) Criar um documento para ser lido por humanos que pode conter, por exemplo, a estrutura de dados que seu serviço recebe e responde, lista de quais operações HTTP estão sendo suportadas naquele serviço, qual formato de mensagem é suportado, etc.

Na minha opinião, esta abordagem faz mais sentido quando a estrutura de dados da mensagem é simples e/ou utiliza notações simples como JSON por exemplo, devido a facilidade da implementação do código que irá acessar ao serviço.

b) A segunda forma, é utilizar o WADL (Web Application Description Language), ele é similar ao WSDL só que mais simples, com atributos de configuração que facilitam a compreensão, por exemplo, de qual notação/formato é utilizada nas mensagens de request e response, e o mais importante, é todo orientado aos recursos e ao protocolo HTTP.

Obs:

WSDL e WADL serve para que aplicações clientes/consumidoras possam gerar o código automaticamente a partir destas definições. No entanto, nada impede que transformações .XSL sejam aplicadas a estes arquivos e eles fiquem mais “legíveis” para leigos, afinal os dois são XML com schema e com vocabulário bem definido.

### SOAP e SOA

Preciso começar este tópico com uma frase clássica, “uma coisa é uma coisa outra coisa é outra coisa”. Nem adianta falar que a diferença entre SOAP e SOA é a letra “P” essa já está velha.

Primeiro, SOAP é um protocolo para troca de informações estruturada, como vimos no início deste post. SOA, por sua vez, é uma arquitetura que se baseia no paradigma de orientação a serviço que molda o desenvolvimento de funcionalidades de negócio, buscando desacoplamento, reutilização, produtividade e alinhamento entre objetivos de negócio e as estratégias de TI. (SOA Manifesto)

Além do nome ser parecido, o que faz com que as pessoas troquem um pelo outro em conversas, e leigos que estão escutando multiplicarem esta confusão por aí..., existe também uma questão relacionada tanto aos big vendors de ferramentas, quanto aos consultores tradicionais atrelados à SOA, que focam muito em Web services, WSDL, WS-\* Extensions e SOAP.

Ainda tem o fato adicional de que a maioria deles estão engatinhando em REST. Ou seja..., tem muitos profissionais de TI, inclusive que se dizem entendidos de SOA, que acham que para se ter uma Arquitetura Orientada a serviço precisamos de Webservices, WSDLs e SOAP, o que está 100% errado.

### **Algumas questões que podem desfazer algumas amarras entre estes conceitos**

a) Podemos ter uma empresa com uma maturidade elevada em SOA, apenas com serviços RESTful? SIM!

b) Podemos ter serviços RESTful com a interface descrita com WSDL.? SIM! É! com WSDL.. WSDL foi criado para descrever interfaces e tem uma boa flexibilidade, podemos usar SOAP sobre HTTP, ou simplesmente HTTP com mensagens em formato JSON, ou XML. (obs: Apartir do WSDL 2.0 isto ficou mais fácil.)

c) Podemos ter serviços disponíveis em HTTP e não serem serviços RESTful? SIM! Formas de utilizar o cabeçalho HTTP, como modelar as URLs, e usar operações GET, POST, PUT, DELETE entre outras coisas vão determinar se o serviço é RESTful ou não.

### **SOAP ou REST ?**

Podemos elencar alguns pontos interessantes:

#### **Rest**

- É mais elegante, pois utiliza ao máximo o protocolo HTTP, evitando a construção de protocolos adicionais
- Tem o potencial de ser bem mais simples que uma implementação com WSDL/SOAP
- Tende a ser mais performático
- ~ 80% das integrações utilizam o protocolo HTTP.
- A possibilidade de ter diferentes representações de um mesmo recurso, por exemplo, uma dada entidade pode ser representada em diferentes formatos como Json, xml, html e text/plain, dependendo da requisição feita pelo cliente(Content-Negotiation)
- Possibilidade de navegar entre relacionamentos (Links web) de vários recursos de forma dinamica. seguindo a usabilidade de qualquer sistema web. HATEOAS (Hypermedia as the Engine of Application State).

#### **SOAP**

- É um padrão que combinado a as especificações WS-\* podem garantir questões de QoS(Quality of Service), Segurança, transação e outras questões presentes em integrações mais complexas.
- Uma mensagem SOAP pode ser propagada por diferentes protocolos, o que flexibiliza bastante várias integrações.
- É um padrão que está muito maduro no mercado, qualquer ferramenta de integração e Framework tem várias funcionalidades para manipular as mensagens que seguem este padrão.

Com certeza existem muitos outros pontos a serem levantados, mas de qualquer forma.. Minha resposta para a pergunta tão popular “Qual o melhor?” seria “Depende”.. depende de qual problema você quer resolver.

Particularmente, acredito que cada demanda tem que ser analisada, com calma, por pessoas que conheçam as 2 abordagens e possam chegar a uma solução adequada ao problema em questão. Isto me faz lembrar de um artigo que o Marcelo Fernandes (companheiro de equipe), me encaminhou há algum tempo que falava sobre uma metodologia ágil muito extremista mas que tinha uma frase interessante.. algo como “Tudo aquilo que você considera verdade, está errado em algum contexto” ou seja, achar uma solução antes de entender o problema é algo que pode funcionar algumas vezes, mas tem um risco, e com certeza em algum, ou vários contextos aquilo não irá se aplicar.. então.. analise cada caso e tire proveito de cada um dos dois.



## SOAP e WebServices

O SOAP é um protocolo elaborado para facilitar a chamada remota de funções via Internet, permitindo que dois programas se comuniquem de uma maneira tecnicamente muito semelhante à invocação de páginas Web.

O SOAP é um protocolo elaborado para facilitar a chamada remota de funções via Internet, permitindo que dois programas se comuniquem de uma maneira tecnicamente muito semelhante à invocação de páginas Web.

O protocolo SOAP tem diversas vantagens sobre outras maneiras de chamar funções remotamente como DCOM, CORBA ou diretamente no TCP/IP:

É simples de implementar, testar e usar.

É um padrão da indústria, criado por um consórcio da qual a Microsoft é parte, adotado pela W3C (<http://www.w3.org/TR/SOAP/>) e por várias outras empresas.

Usa os mesmos padrões da Web para quase tudo: a comunicação é feita via HTTP com pacotes virtualmente idênticos; os protocolos de autenticação e encriptação são os mesmos; a manutenção de estado é feita da mesma forma; é normalmente implementado pelo próprio servidor Web.

Atravessa “firewalls” e roteadores, que “pensam” que é uma comunicação HTTP.

Tanto os dados como as funções são descritas em XML, o que torna o protocolo não apenas fácil de usar como também muito robusto.

É independente do sistema operacional e CPU.

Pode ser usado tanto de forma anônima como com autenticação (nome/senha).

Os pedidos SOAP podem ser feitos em três padrões: GET, POST e SOAP. Os padrões GET e POST são idênticos aos pedidos feitos por navegadores Internet. O SOAP é um padrão semelhante ao POST, mas os pedidos são feitos em XML e permitem recursos mais sofisticados como passar estruturas e arrays. Independente de como seja feito o pedido, as respostas são sempre em XML.

O XML descreve perfeitamente os dados em tempo de execução e evita problemas causados por inadvertidas mudanças nas funções, já que os objetos chamados têm a possibilidade de sempre validar os argumentos das funções, tornando o protocolo muito robusto.

O SOAP define também um padrão chamado WSDL, que descreve perfeitamente os objetos e métodos disponíveis, através de páginas XML acessíveis através da Web. A idéia é a seguinte: quem publicar um serviço, cria também estas páginas.

Quem quiser chamar o serviço, pode usar estas páginas como “documentação” de chamada e também usadas antes de chamar as funções para verificar se alguma coisa mudou.

O SOAP pode ser facilmente implementado em virtualmente qualquer ambiente de programação. Existem atualmente diversos “kits” de desenvolvimento SOAP para vários sistemas operacionais e linguagens de alto nível. A própria Microsoft tem um “kit” para o Visual Studio 6 em <http://msdn.microsoft.com/soap/default.asp>.

O SOAP é uma parte importante da arquitetura .NET da Microsoft e tem um extenso suporte no Visual Studio.NET. Um WebService é um conjunto de métodos WebMethods logicamente associados e chamados através de SOAP.

Os WebMethods são funções chamadas remotamente através de SOAP. Cada WebService tem dois arquivos associados: um com extensão “asmx” e outro com extensão “cs” se você estiver usando a linguagem C#.

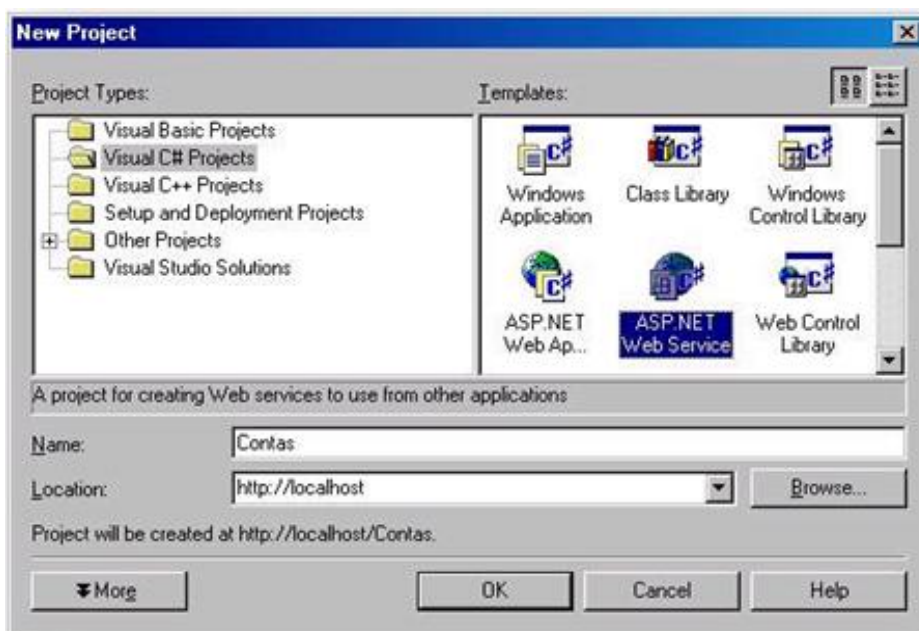
Na arquitetura .NET os WebServices são implementados sempre em uma classe derivada de “System.Web.Services.WebService”. Nesta classe adicionamos as funções (métodos) que serão chamados via SOAP. A diferença entre um WebMethod e um método comum é a presença de um “atributo

WebMethod”, uma espécie de diretiva de compilação. A página SDL é gerada automaticamente pelas ferramentas de programação.

Do ponto de vista do programador, um Webservice é uma página ASP.NET “glorificada”, que mapeia automaticamente pedidos via Web a métodos de uma linguagem de alto-nível.

**Criando um Web Service**

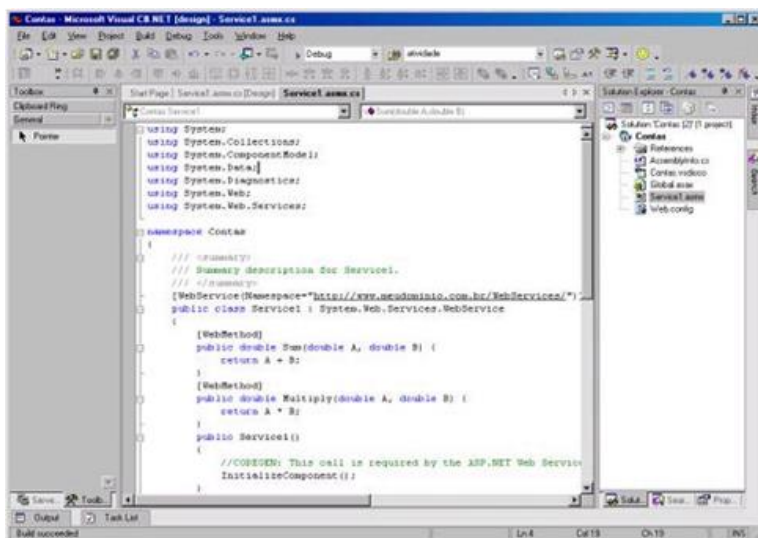
Veja a seguir uma seqüência de criação de um Webservice que faz algumas contas simples. Inicialmente criamos um aplicativo Webservice no Visual Studio.NET Beta 2:



**Figura 1:** Criando novo web service

Este é um projeto ASP.NET comum. Na prática você provavelmente irá também acrescentar páginas ASP.NET comuns.

A seguir escrevemos o código do serviço:



**Figura 2:** Código no editor

Veja o código digitado:

**Listagem 1:** Código do serviço

```
1[WebMethod]
2public double Sum(double A, double B) {
3return A + B;
4}
5[WebMethod]
6public double Multiply(double A, double B) {
7return A * B;
8}
```

Note o atributo [WebMethod], que sinaliza ao sistema de runtime que este é um método chamado via HTTP. Nem todos os métodos precisam ser WebMethods.

Todo WebService deve ser identificado de forma única no Universo. A maneira de fazer isto é fornecer uma URI baseada em um domínio Internet registrado por você ou pela sua empresa. Esta URI deve ser fornecida em um atributo antes da declaração da classe:

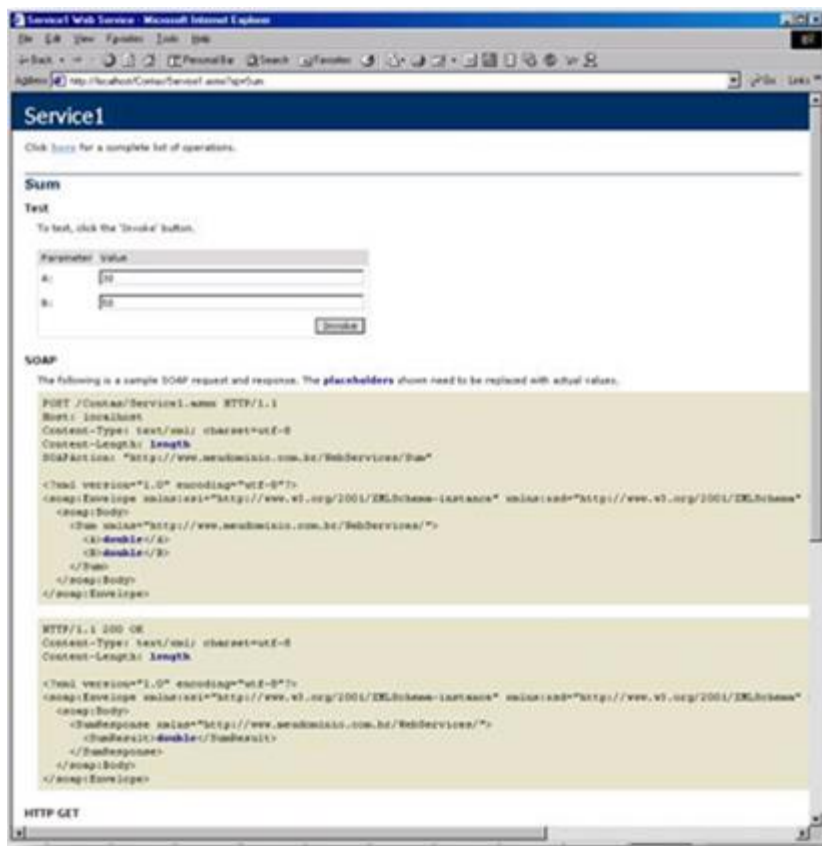
```
[WebService(Namespace="http://www.meudominio.com.br/WebServices/")]
```

Após pedirmos "Build", podemos imediatamente acessar o serviço através de um navegador Web. Peça "Debug | Start Without Debugging". As rotinas de suporte a WebServices irão criar automaticamente uma página para testar o WebService:



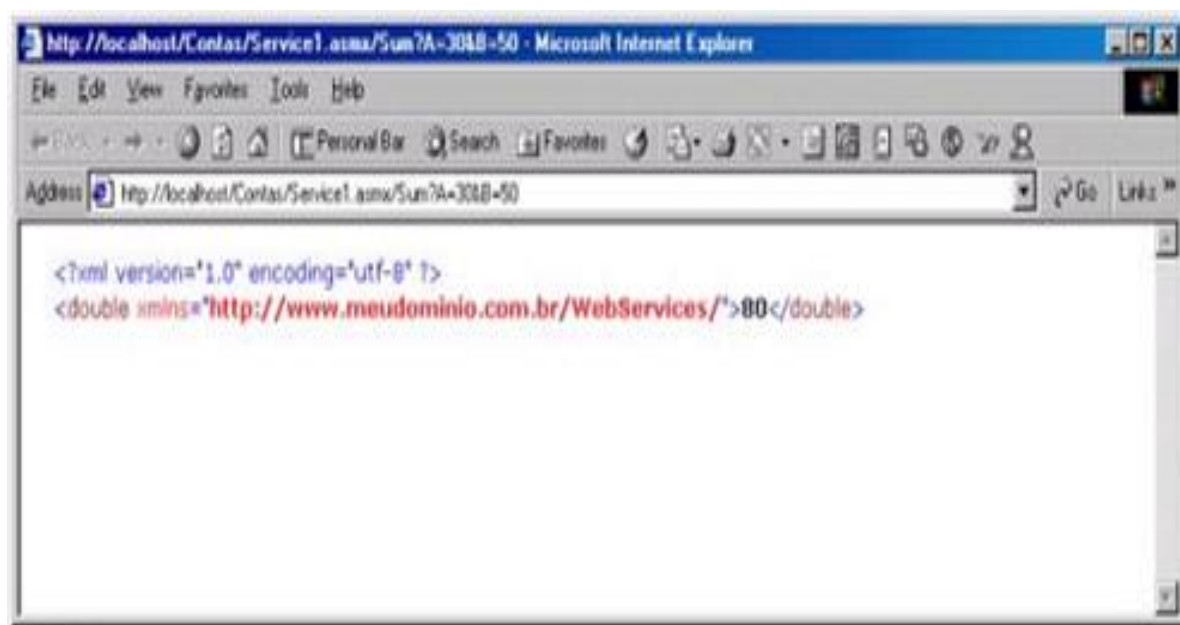
**Figura 3:** Página de teste do webservice

Selecione algum método, Sum, por exemplo:



**Figura 4:** Página de execução do método Sum

Preencha alguns valores e veja a saída do teste:



**Figura 5:** Resultado do método Sum

Podemos interrogar a descrição do serviço como XML através do padrão “WSDL”. Esta é a maneira que será usada pelo Visual Studio.NET para criar automaticamente uma classe “proxy” que chamará o Webservice:

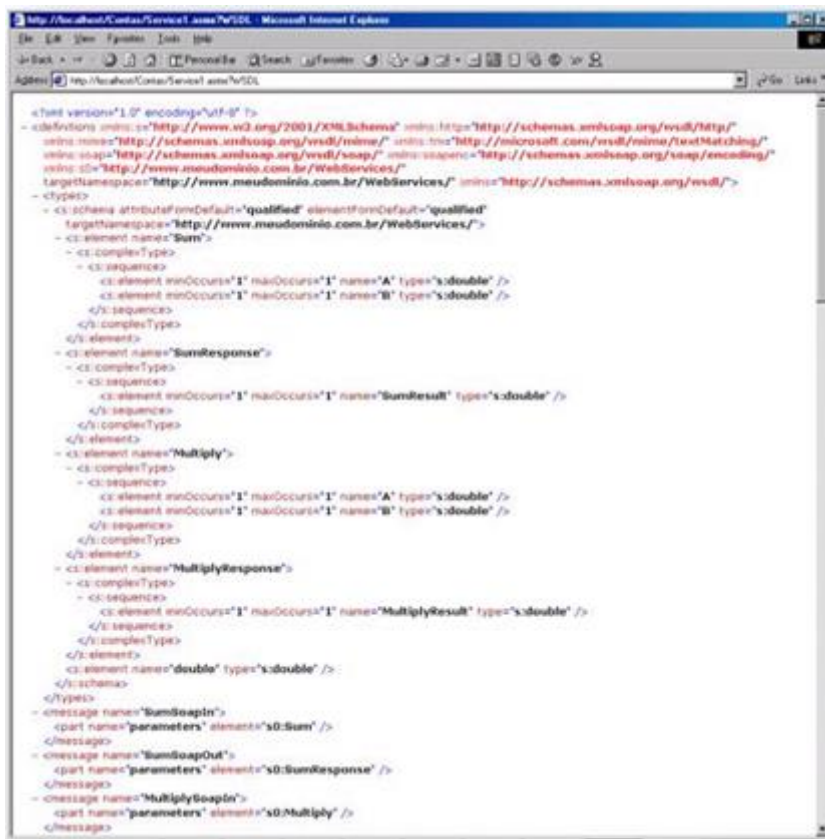


Figura 6: Descrição do serviço como XML

Observe que o VS.NET Beta 1 usava o padrão “SDL” correspondente ao “SOAP Toolkit 1.0”, baseado em uma especificação temporária do protocolo. O Beta 2 utiliza o padrão definitivo e ligeiramente diferente chamado “WSDL”, correspondente ao SOAP Toolkit 2.0.

### Consumindo um Web Service

Para consumir um Webservice, o Visual Studio.NET pode criar uma classe “proxy” a partir de informação obtida interrogando o SDL. Vamos criar um novo projeto “WinForms” para chamar o WebService:

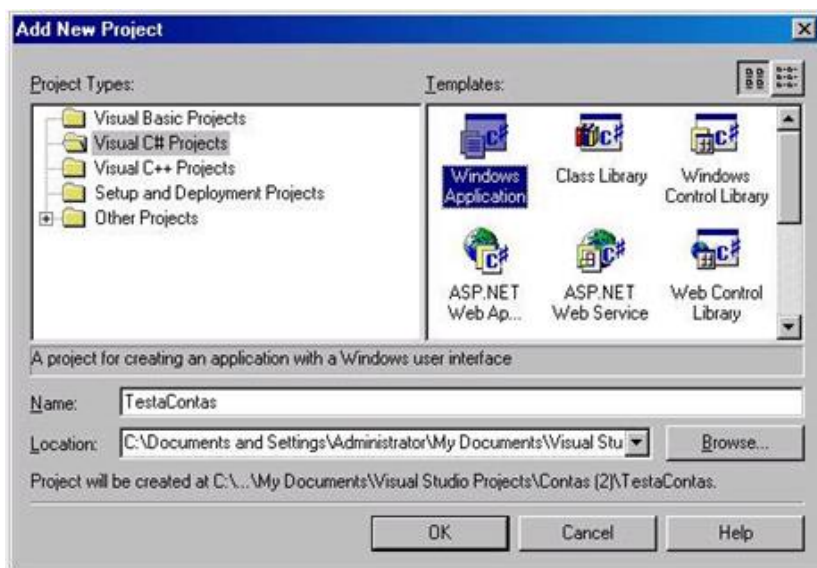
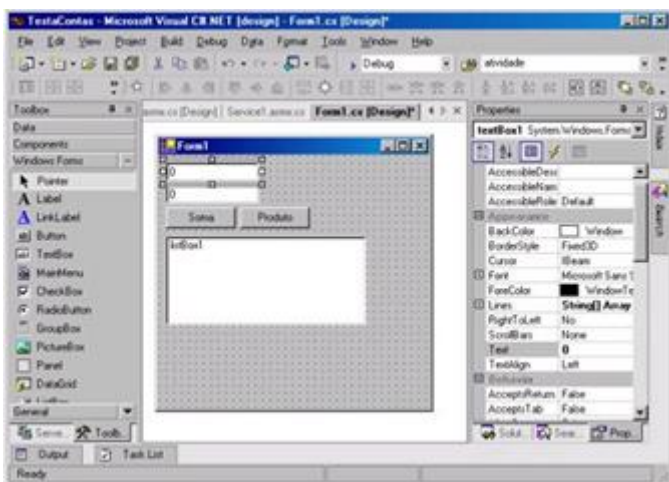


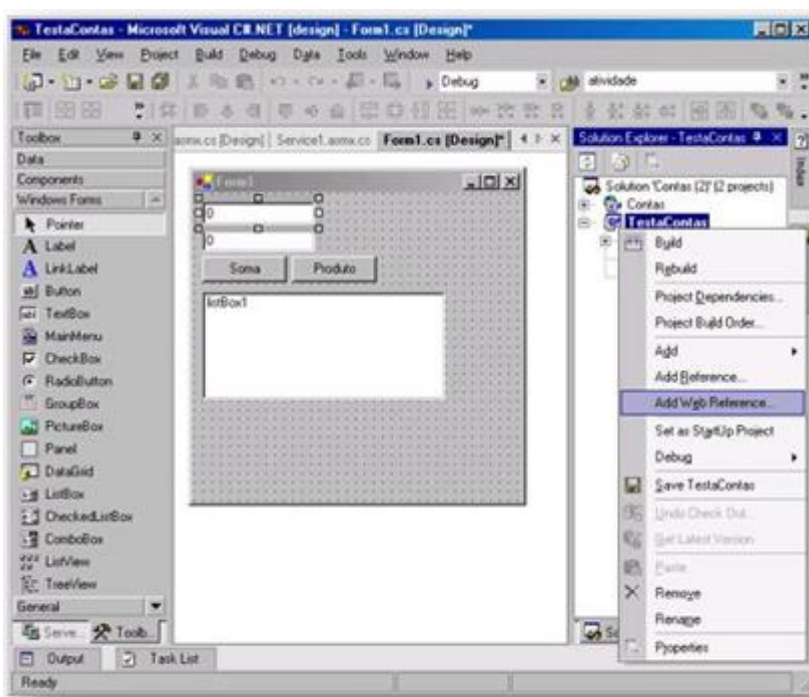
Figura 7: Criando nova aplicação windows

Adicione dois TextBox, um Button e um ListBox:



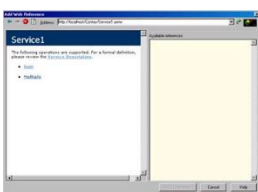
**Figura 8:** Layout do form

Clique com o botão direito sobre o projeto e peça “Add Web Reference...”:



**Figura 9:** Referenciando o web service

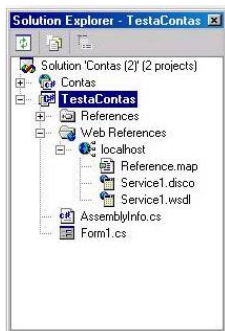
Entre com a URL do Webservice, <http://localhost/Contas/Service1.asmx>, no caso. A janela que aparece à esquerda é um navegador Web que pode ser utilizado normalmente para navegação e localização dos WebServices, listados à direita:



**Figura 10:** Informando o caminho do web service

Clique “Add Reference”. O Visual Studio.NET criará uma classe “proxy” no projeto. Esta classe tem a mesma sintaxe de uma classe .NET, mas na verdade está invocando um Webservice.

Veja o projeto com a “Web Reference”:



**Figura 11:** Estrutura do projeto após a adição da referência

Se você tiver curiosidade, esta é a classe criada:



**Figura 12:** Código da classe criada

O código que chama o Webservice é o seguinte:

**Listagem 2:** Código para utilizar o serviço

```

1 // Soma
2 private void button1_Click(object sender, System.EventArgs e) {
3     double N1 = Convert.ToDouble(textBox1.Text);
4     double N2 = Convert.ToDouble(textBox2.Text);
5     localhost.Service1 Contas = new localhost.Service1();
6     double R = Contas.Sum(N1, N2);
7     listBox1.Items.Add(R.ToString());

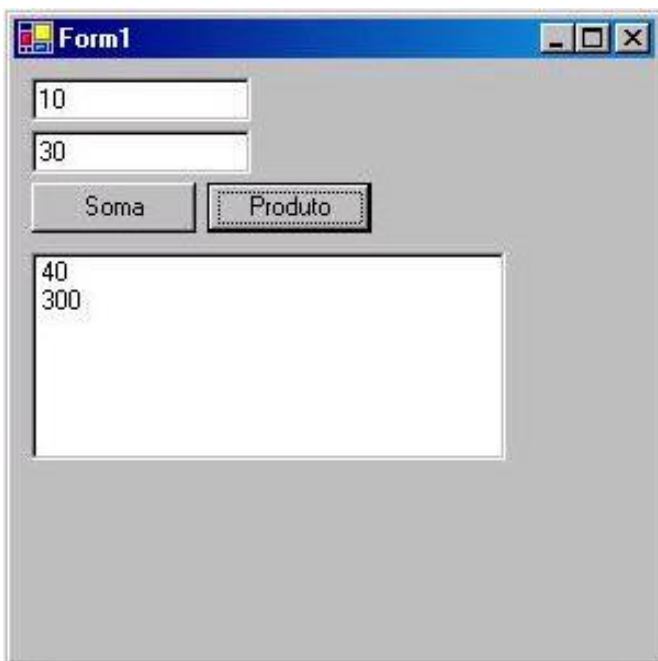
```

```

8 }
9 // Produto
10private void button2_Click(object sender, System.EventArgs e) {
11double N1 = Convert.ToDouble(textBox1.Text);
12double N2 = Convert.ToDouble(textBox2.Text);
13localhost.Service1 Contas = new localhost.Service1();
14double R = Contas.Multiply(N1, N2);
15listBox1.Items.Add(R.ToString());
16}

```

Veja o programa rodando e chamando o Webservice:



**Figura 13:** Programa em execução

Note que criamos um objeto da classe proxy e chamamos seus métodos para usá-la. O objeto Cotasusado para chamar o Webservice funciona de forma muito semelhante a um navegador Internet, mas sem interpretar documentos HTML. Ele pode armazenar cookies, fazer autenticação e encriptação, ter um nome como "user agent" e usar um servidor de proxy.

Na segunda parte, mostrarei o uso de WebServices "de verdade" na Internet e também como criar-WebServices mais complexos.

## REST

Origem: Wikipédia, a enciclopédia livre.

A **Representational State Transfer (REST)**, em português **Transferência de Estado Representacional**, é uma abstração da arquitetura da World Wide Web (Web), um estilo arquitetural que consiste de um conjunto coordenado de restrições arquiteturais aplicadas a componentes, conectores e elementos de dados dentro de um sistema de hipermídia distribuído.



O REST ignora os detalhes da implementação de componente e a sintaxe de protocolo com o objetivo de focar nos papéis dos componentes, nas restrições sobre sua interação com outros componentes e na sua interpretação de elementos de dados significantes.

O termo transferência de estado representacional foi apresentado e definido no ano de 2000 por Roy Fielding, um dos principais autores da especificação do protocolo HTTP que é utilizado por sites da Internet, em uma tese de doutorado (PHD) na UC Irvine.

A REST tem sido aplicada para descrever a arquitetura web desejada, identificar problemas existentes, comparar soluções alternativas e garantir que extensões de protocolo não violem as principais restrições que fazem da Web um sucesso. Fielding desenvolveu a REST em colaboração com seus colegas enquanto trabalhava no HTTP 1.1 e nos Identificadores de Recursos Uniformes.

O estilo arquitetural de REST também é aplicado no desenvolvimento de serviços Web. Pode-se caracterizar os web services como "RESTful" se eles estiverem em conformidade com as restrições descritas na seção restrições arquiteturais.

"A REST (Transferência do Estado Representativo) é pensada como uma imagem do design da aplicação se comportará: uma rede de sítios da Teia (um estado virtual), onde o utilizador progride com uma aplicação clicando em vínculos (transições do estado), tendo como resultado a página seguinte (que representa o estado seguinte da aplicação) que está sendo transferida ao utilizador e apresentada para seu uso."

O termo REST se referia, originalmente, a um conjunto de princípios de arquitetura (descritos mais abaixo), na atualidade se usa no sentido mais amplo para descrever qualquer interface web simples que utiliza XML (ou YAML, JSON, ou texto puro) e HTTP, sem as abstrações adicionais dos protocolos baseados em padrões de trocas de mensagem como o protocolo de serviços Web SOAP.

É possível projetar sistemas de serviços Web de acordo com o estilo arquitetural REST descrito por Fielding, e também é possível projetar interfaces XMLHTTP de acordo com o estilo de RPC mas sem utilizar SOAP. Estes usos diferentes do termo REST causam certa confusão em discussões técnicas, onde RPC não é um exemplo de REST.

Os sistemas que seguem os princípios REST são freqüentemente chamados de RESTful'.

### Princípios

REST afirma que a Web já desfrutou de escalabilidade como resultado de uma série de conceitos de projeto fundamentais:

Um protocolo cliente/servidor sem estado: cada mensagem HTTP contém toda a informação necessária para compreender o pedido. Como resultado, nem o cliente e nem o servidor necessitam gravar nenhum estado das comunicações entre mensagens.

Na prática, muitas aplicações baseadas em HTTP utilizam cookies e outros mecanismos para manter o estado da sessão (algumas destas práticas, como a reescrita de URLs, não são permitidas pela regra do REST).

Um conjunto de operações bem definidas que se aplicam a todos os recursos de informação: HTTP em si define um pequeno conjunto de operações, as mais importantes são **POST**, **GET**, **PUT** e **DELETE**.

Com frequência estas operações são combinadas com operações CRUD para a persistência de dados, onde POST não se encaixa exatamente neste esquema.

Uma sintaxe universal para identificar os recursos. No sistema REST, cada recurso é unicamente direcionado através da sua URI.

O uso de hipermídia, tanto para a informação da aplicação como para as transições de estado da aplicação: a representação deste estado em um sistema REST são tipicamente HTML ou XML.

Como resultado disto, é possível navegar com um recurso REST a muitos outros, simplesmente seguindo ligações sem requerer o uso de registros ou outra infraestrutura adicional.

### Recursos

Um conceito importante em REST é a existência de recursos (elementos de informação), que podem ser usados utilizando um identificador global (um Identificador Uniforme de Recurso) para manipular estes recursos, os componentes da rede (clientes e servidores) se comunicam através de uma interface padrão (HTTP) e trocam representações de recursos (os arquivos ou ficheiros são recebidos e enviados) – é uma questão polêmica e gera grande discussão, sem a distinção entre recursos e suas representações é demasiado utópico o seu uso prático na rede, onde é popular na comunidade RDF.

O pedido pode ser transmitido por qualquer número de conectores (por exemplo clientes, servidores, caches, etc) mas não poderá ver mais nada do seu próprio pedido (conhecido com separação de camadas, outra restrição do REST, que é um princípio comum com muitas outras partes da arquitetura de redes e da informação).

Assim, uma aplicação pode interagir com um recurso conhecendo o identificador do recurso e a ação requerida, não necessitando conhecer se existem caches, proxys, ou outra, entre ela e o servidor que guarda a informação.

A aplicação deve compreender o formato da informação de volta (a representação), que é geralmente um documento em formato HTML ou XML, onde também pode ser uma imagem ou qualquer outro conteúdo.

### REST e RPC

Uma aplicação web REST requer um enfoque de desenho diferente a uma aplicação baseada em RPCs. No RPC, dá-se ênfase à diversidade de operações do protocolo, ou verbos; por exemplo uma aplicação RPC poderia definir operações como:

```
getUser()
```

```
addUser()
```

```
removeUser()
```

```
updateUser()
```

```
getLocation()
```

```
addLocation()
```

```
removeLocation()
```

```
updateLocation()
```

```
listUsers()
```

```
listLocations()
```

```
findLocation()
```

```
findUser()
```

Em REST, ao contrário, a ênfase está na diversidade de recursos, nos nomes; por exemplo, uma aplicação REST poderia definir os seguintes tipos de recursos:

```
Usuario {}
```

Localizacao {}

Cada recurso teria seu próprio identificador, como [http://www.example.org/locations/us/ny/new\\_york\\_city](http://www.example.org/locations/us/ny/new_york_city).

Os clientes trabalhariam com estes recursos através das operações padrão de HTTP, como o GET para chamar uma cópia do recurso.

Observa-se como cada objeto tem sua própria URL e pode ser facilmente "cacheado", copiado e guardado como marcador. POST utiliza-se geralmente para ações com efeitos colaterais, como enviar uma ordem de compra.

Por exemplo, o registo para um Utilizador poderia ter o seguinte aspecto:

```
<usuario>
<nome>Maria Juana</nome>
<genero>feminino</genero>
<localizacao href="http://www.example.org/locations/us/ny/new_york_city"
>Nova York, NY, US</localizacao>
</usuario>
```

Para atualizar a localização do utilizador, um cliente REST poderia primeiro chamar um registo XML anterior usando GET. O cliente depois modificaria o arquivo para mudar a localização e submetê-la-ia para o servidor utilizando HTTP PUT.

Note-se que o HTTP não proporciona nenhum recurso padrão para descobrir recursos – não há nenhuma operação LIST ou FIND em HTTP, que se corresponda às operações list\*() e find\*() como por exemplo em RPC.

No seu lugar, as aplicações baseadas em dados REST resolvem o problema, tratando uma coleção de resultados de busca como outro tipo de recurso, o que requer que os engenheiros de software desenhem na aplicação colocando URLs adicionais para mostrar ou encontrar cada tipo de recurso.

Por exemplo, um pedido GET HTTP sobre a URL <http://www.example.org/locations/us/ny/> poderia devolver uma lista de arquivos em XML com todas as localizações possíveis em Nova Iorque, e outra, um pedido GET para URL <http://www.example.org/users?nome=Michaels> poderia devolver uma lista de todos os usuários com o apelido "Michaels".

REST proporciona algumas indicações sobre como realizar este tipo de ação como parte de sua restrição "hipermídia como o meio de estado da aplicação", o que sugere o uso de uma linguagem de marcação (como um formulário HTML) para especificar consultas parametrizadas.

A iniciativa OpenSearch da A9.com tenta padronizar as buscas usando REST estabelecendo especificações para descobrir recursos e um formato genérico para utilizar sistemas baseados em REST, incluindo o RDF, XTM, Atom, RSS (e suas várias formas) e XML com XLink para gerir as ligações.

### Implementações Públicas

Dado que a definição de REST é muito ampla, é possível afirmar que existe um enorme número de aplicações REST na rede (praticamente qualquer coisa acessível mediante um pedido HTTP GET). De forma mais restritiva, em contraposição aos serviços web e ao RPC, REST se pode encontrar em diferentes áreas da web:

Na blogosfera -o universo dos blogs- está, em sua maior parte, baseado em REST, dado que implica chamar arquivos/ficheros XML (em formato RSS ou Atom) que contém listas de ligações a outros recursos.



## Webhook

O webhook é uma técnica que vem ganhando seu espaço no mundo empresarial. Confira o nosso post e descubra: o que é, como funciona e por que sua equipe de tecnologia precisa experimentar. Já pensou se o usuário do seu software pudesse conseguir informações variadas em tempo real sem que ele precisasse interromper uma atividade, tendo de entrar em outras páginas e aplicativos para fazer buscas?

Seria demais, não?

Agora, imagine que ele seja capaz de poder conferir se o status das notas fiscais foram alterados direto do seu software, ajudaria bastante, não é mesmo?

Então... e seu eu te contar que esses são somente alguns diferenciais que a implementação do webhook pode proporcionar aos seus usuários?

Ficou interessado? Quer entender mais sobre o assunto? Então acompanhe o nosso guia para ficar por dentro a respeito desse recurso estratégico!

O que é Webhook?

O webhook, também conhecido como retorno de chamada web, é um método simples que facilita com que uma API forneça informações em tempo real sempre que um evento acontece.

Em outras palavras, trata-se de um modo de receber dados entre dois sistemas de forma passiva, ou seja, sem realizar nenhuma ação.

Portanto, um webhook é desenvolvido para reproduzir automações pré-definidas na ferramenta ou sistema de chegada, independentemente da origem da informação.

Dessa maneira, você poderá fazer com que uma ação X, do sistema 1, por exemplo, gere uma ação Y no sistema 2.

A ponta de “reação” das automações é fixa, mas cabe a você definir qual será o “gatilho” usado.

Assim, isso possibilita que programas simples e independentes sejam criados exclusivamente com o intuito de funcionar em cadeia conforme outros comandos são executados.

Qual a vantagem do webhook para os desenvolvedores?

Como vimos anteriormente, o webhook fornece automaticamente informações em tempo real, evitando que haja um trabalho de pesquisa a cada minuto em busca de uma resposta.

Por consequência, isso impede que aconteça uma sobrecarga do sistema e a perda de dados sempre que transcorrer algum erro.

Logo, um webhook facilita bem mais o serviço do desenvolvedor, além de deixar todo o processo mais simples e também mais barato.

Isso, pois, se formos analisar o que acontece em um sistema sem webhook, será possível identificar que fica impraticável, por exemplo, para uma integração descobrir onde enviar a mensagem.

Assim, a ferramenta fica obrigada a consultar periodicamente a API em busca de alguma mudança ou atualização.

Por fim, conclui-se que haveria um trabalho mais complexo e que exigiria muito mais tempo e empenho, além de encarecer o software.

Como funciona um webhook?

Webhooks permitem que sistemas externos recebam notificações de todos os eventos que ocorrem na API.

Quando um evento acontece, a API pode enviar uma requisição, sendo a mais comum a HTTP (POST), para a URL configurada no webhook com as informações relativas ao evento.

Ao receber a notificação, o sistema integrado à API pode executar diversas tarefas, dependendo do seu objetivo.

O webhook viabiliza que você receba notificações a cada alteração de status de um tipo de nota fiscal, por exemplo.

Desta forma, sempre que uma nota tiver seu status atualizado (aprovada, rejeitada, cancelada, etc) será enviado uma chamada via POST (PATCH, DELETE, PUT, etc), por exemplo, para uma URL pré-configurada conforme os parâmetros do seu conteúdo, de modo que você consiga realizar suas decisões comerciais.

Desse jeito, será possível criar gatilhos para determinadas ações, à medida que o status da nota fiscal for especificado.

Na prática, o que precisa ser entendido é que os Webhooks enviam dados em tempo real entre suas ferramentas ou sistemas para a URL do webhooks. E é dessa forma que você irá automatizar suas tarefas.

Diante disso, podem ser usados diversos formatos, sendo os mais utilizados, o JSON ou o XML. Veja só a diferença entre cada um deles:

- ⊕ O XML é uma sintaxe básica que pode ser usada para compartilhar informações entre aplicações, permitindo reutilizar o código com objetivos diferentes no outro sistema ou ferramenta.
  
- ⊕ O JSON é um formato mais leve de troca de informações entre sistemas. Suas siglas se referem a "JavaScript Object Notation", mas isso não significa que só pode ser usado com JavaScript.

Antes de criar um webhook para o seu sistema, é preciso que você observe alguns fatores de modo a evitar erros e problemas futuros.

Pensando nisso, montamos uma lista com os principais quesitos que você precisa se atentar antes de elaborar o seu webhook. Confira:

Tome cuidado com a perda de dados

Os Webhooks entregam dados ao seu aplicativo. Portanto, se a sua aplicação por algum motivo tiver um erro de dados, os mesmos podem ser perdidos.

Observe se há dados duplicados

Boa parte dos webhooks irão se basear nas respostas e solicitações de reenvio, sobretudo, se o seu aplicativo apresentar erros com frequência.

Se o seu pedido for processado e ainda houver esses erros, isso pode gerar dados duplicados em seu aplicativo.

Verifique se há erros de aplicação

Erros de aplicação são comuns, por isso, busque entender como o seu webhook provedor lida com as respostas de cada ação, assim você pode se preparar para contornar essa adversidade.

Explore a possibilidade da manipulação em escala

Um grande benefício dos webhooks, é que eles podem fazer vários pedidos ao mesmo tempo.

Portanto, o ideal é verificar se o seu aplicativo é capaz de manipular a escala esperada do seu webhook.

Opte por serviços adequados à sua demanda

A melhor maneira de compreender verdadeiramente um webhook é testar um serviço que se adequa à sua necessidade.

Felizmente, muitos serviços usam webhooks, assim, você pode facilmente aprender com eles e conhecer sobre todo o conteúdo referente ao sistema.

Webhook x API REST: quais são as diferenças?

Em uma API REST, definimos um recurso para receber dados, para então devolver alguma resposta.

Para enviar uma mensagem para um webhook, é feito uma requisição e também recebemos uma resposta (dependendo do caso).

Mas e aí?...Ambos fazem a mesma coisa? Qual é a diferença? Seria algo apenas conceitual ou de implementação?

Bom, vamos por partes:Um Webhook é um serviço, API é o conjunto de regras e métodos com que podemos comunicar e usar esse serviço.

Há webhooks que irão utilizar uma API do tipo REST, outros que usam API customizada, outras GraphQL, Thrift, etc.

Em outras palavras, um WebHook é um endpoint HTTP da qual um cliente deve implementar para receber notificações de eventos que ocorrem em um servidor.

Ambos fazem a mesma coisa?

Não. Em uma API REST, o servidor expõe um conjunto de serviços através de endpoints HTTP.

Já em um WebHook, o cliente expõe um endpoint HTTP e se inscreve no servidor para receber notificações de eventos.

Então qual é a diferença?

A diferença é que o cliente consome a API REST do servidor diretamente por meio de requisições HTTP, enquanto que os WebHooks são requisições HTTP das quais o cliente deve aguardar.

Como funciona o webhook do eNotas?

No caso do eNotas, seu sistema será notificado sempre que uma nota fiscal mudar para um dos status abaixo:

- Autorizada - Nota fiscal foi autorizada pelo órgão responsável;

- Negada - Nota fiscal foi negada pelo órgão responsável;
- Cancelada - O cancelamento da nota fiscal foi autorizado pelo órgão responsável;
- CancelamentoNegado - O cancelamento da nota fiscal foi rejeitado pelo órgão responsável;

Vale lembrar que, o valor em negrito representa o status que será retornado no parâmetro "nfeStatus".

#### Atenção: Parâmetro "nfeMotivoStatus"

Este parâmetro só retorna diferente de null nos seguintes status:

- "nfeStatus": "Negada";
- "nfeStatus": "CancelamentoNegado".

Já em relação aos dados do POST de notificação, estes serão variados conforme o tipo de nota fiscal.

No exemplo abaixo há o modelo de webhook para NFS-e:

Para cadastrar o webhook, será utilizado o painel do eNotas Gateway, acessando o menu Webhooks → Novo webhook:

Se seu sistema apresentar instabilidades ou estiver fora do ar, o Gateway tentará **lhe enviar o webhook de notificação até receber o status "HTTP 200"**, que indica que o webhook foi recebido e tratado com sucesso.

Serão realizadas **15 tentativas** com **intervalos de tempo progressivos** entre cada uma delas, **totalizando, em média, 4 horas**.



## Container Docker

A tecnologia está em constante evolução e a todo momento surgem novas ideias que aprimoram os métodos utilizados em TI. O container docker é uma delas e pode trazer vários benefícios para a sua organização.

As empresas optam por essa modalidade porque os containers trazem diferentes possibilidades. Por serem ambientes isolados e portáteis, os desenvolvedores podem empacotar aplicações com bibliotecas e links necessários. O resultado é mais eficiência no trabalho e a simplificação da implantação.

Porém, essa descrição ainda não diz todos os benefícios do uso dessa tecnologia. Por isso, criamos este post. Aqui, vamos explicar os principais detalhes sobre o docker, passando pelos seguintes aspectos:

conceito;

diferenças do sistema virtualização dessa forma;

por que se tornou uma tendência do mercado;

benefícios do uso.

Então, que tal entender melhor essa tecnologia? É só continuar lendo!

O que é um container docker?

Esse conceito precisa ser compreendido em partes. O container é um ambiente isolado. Já o docker é uma plataforma open source na linguagem de programação Go, que possui alto desempenho e é desenvolvida diretamente no Google.

Assim, o docker agrupa partes de softwares de um sistema de arquivo completo e que abrange todos os recursos necessários para a sua execução. Por isso, é uma plataforma de containers.

Isso significa que tudo pode ser instalado no servidor e é armazenado nos containers. Dessa forma, os mesmos softwares e as suas versões podem ter uma execução facilitada em qualquer ambiente de desenvolvimento.

O que ocorre na prática é que o docker destaca recursos e usa bibliotecas de kernel em comum. Os itens empacotados — ou até mesmo um ambiente inteiro — são dispostos no container e se tornam portáteis, o que torna o trabalho conjunto mais eficiente. Ao mesmo tempo, a implantação pode ser feita em ambientes não heterogêneos.

Assim, o docker é uma implementação de virtualização de containers que vem conquistando cada vez mais espaço devido à computação em nuvem. Anteriormente, o hypervisor dominava o ambiente, mas agora ele é excluído do processo.

Devido a todas as facilidades que oferece, o container é uma das principais tendências de TI. Ele simplifica a aplicação da metodologia DevOps e facilita o desenvolvimento ágil, tanto que o Google utiliza essa tecnologia há mais de 10 anos.

Já o docker, por ser open source, possibilita a execução de deploys e o escalonamento de aplicações com mais facilidade. Além disso, devido à virtualização por container, propicia um ambiente isolado e leve para rodar o programa.

Quais as diferenças do sistema virtualizado com docker?

O container exclui a virtualização pelo hypervisor e muda o processo para o docker. Essa é a principal modificação. No entanto, há vários detalhes intrincados que precisam ser compreendidos.

Para facilitar, vamos apresentar o modelo tradicional de virtualização e o novo, com o container:

### Sistema virtualizado por VM

O modelo mais comum conta com uma máquina virtual (VM, do inglês virtual machine), que trabalha com um sistema operacional (SO) completo, porém separado do equipamento. A execução do software é feita em cima de um servidor físico com a finalidade de emular determinado sistema de hardware.

Esse processo é possibilitado pelo hypervisor, software que cria e efetiva a VM. Basicamente, ele se localiza entre o hardware e o SO, sendo um elemento fundamental para a virtualização do servidor. Um exemplo de hypervisor é o VMWare, Hyper-V e o VirtualBox.

Uma mesma máquina pode contar com diferentes VMs. Cada uma opera um SO exclusivo e possui seu próprio kernel, binários, aplicativos e bibliotecas, o que significa que ocupam um espaço grande no servidor.

Quando foi criada, essa tecnologia trouxe diversos benefícios devido à capacidade de consolidação de aplicativos em um sistema único. Com isso, é possível alcançar a redução de custos pelo rápido provisionamento e obter o aprimoramento da possibilidade de recuperação de desastres.

Duas áreas muito fortalecidas e beneficiadas com essa medida foi a de QA e development, porque se tornou possível testar os programas e usar os servidores liberados para montar esses ambientes.

Assim, apesar de haver vários benefícios com a virtualização por VM, ainda era necessário criar uma alternativa mais simples e leve.

### Sistema Virtualizado Com Docker

Esse é um sistema desenvolvido na linguagem de programação Go. Com ele, os desenvolvedores conseguem criar e administrar diferentes ambientes isolados, fazendo com que os dockers sejam um sistema de virtualização diferente do tradicional.

Isso acontece porque esse novo sistema de virtualização usa o Linux Container (LXC) como backend. Ele não fornece uma VM, mas sim um ambiente virtual semelhante ao chroot, mas com um isolamento maior. Essa característica permite definir limitações de recursos por container, por exemplo, CPU, memória, I/O, entre outros.

Na prática, o SO convidado e o hypervisor são eliminados e o host entra em contato direto com as bibliotecas. Com essa ligação, os itens ficam portáteis para qualquer outro host que também possua o sistema de virtualização instalado. A consequência é a redução do tempo de deploy de uma aplicação ou infraestrutura.

Assim, é desnecessário ajustar o ambiente para que o serviço funcione corretamente. Ou seja, ele é sempre igual e é configurado uma vez. A partir disso, basta replicá-lo.

Outras possibilidades interessantes são criar os containers prontos para deploy (ou seja, imagens) a partir de dockerfiles, que são arquivos de definição.

É importante destacar que os containers se localizam em cima de um servidor físico e do SO hospedeiro. Cada um deles compartilha o kernel do SO host e costuma partilhar também as bibliotecas e binários.

A diferença é que os itens compartilhados servem somente para leitura, o que torna o container muito leve, especialmente se comparado à VM. Para ter uma ideia, o tamanho do primeiro é de Mb e, por isso, ele é iniciado em poucos segundos. Já a máquina virtual contém vários Gb e demora minutos para ser executada.

Perceba que esse novo modelo de virtualização prevê o fornecimento do básico para que uma aplicação seja executada em um SO hospedeiro.

Assim, escolher entre uma VM e um docker depende do contexto do projeto que se pretende realizar. A principal diferença é relativa ao consumo de recursos e espaço, porque a modalidade de virtualização mais recente possui diferentes camadas, que são reunidas com o UnionFS.

Essa característica traz mais agilidade ao processo, porque o rebuild é desnecessário para o update das imagens. De modo geral, os containers são mais interessantes devido a alguns fatores, como:

velocidade;

rapidez no boot;

economia de recursos;

entendimento dos processos do container como sendo realizados dentro do sistema host;

possibilidade de fazer o upload de vários containers simultaneamente, o que consome menos recursos do hardware virtual ou físico.

Assim, o container docker leva a um nível superior de virtualização, o que traz diferentes vantagens para a organização.

Como o docker funciona?

As médias e grandes empresas, principalmente, utilizam aplicações como ERPs e CRMs, ou seja, conjuntos de software que iniciam como projetos simples, mas que se tornam ineficientes com o tempo e impedem o progresso por contarem com um código-fonte monolítico.

O container surgiu para resolver esse problema. A solução passa por diferentes etapas. A primeira delas é a desagregação do aplicativo em componentes menores, os chamados microsserviços. A partir disso, os desenvolvedores conseguem adotar uma arquitetura que aumenta a eficiência operacional. Isso acontece porque o código-fonte é destinada para cada componente de aplicação. Assim, o software passa por vários estágios, por exemplo, vai para um ambiente de testes, para um virtual e, por último, de produção.

Em cada um desses locais, a aplicação deve ter uma performance consistente, que é garantida pelo container. Ele encapsula os componentes em um pacote único e leve, que permite executar os aplicativos com consistência independentemente do ambiente ser virtual ou físico.

Já o docker faz a comunicação entre cliente e servidor por meio de um API. Para realizar esse workflow, é necessário ter o serviço instalado em um local e apontar o cliente para esse servidor.

A plataforma em si usa alguns conjuntos de recursos para criar e administrar os containers, inclusive limitar os recursos. Dentre eles está a biblioteca libcontainer, que efetua a comunicação entre o Docker Daemon e o backend.

Vale a pena destacar que o docker está embasado no conceito de layers. O container é construído por meio de chroot, namespaces, cgroups e outras funcionalidades do kernel a fim de isolar a área para a sua aplicação.

Desse modo, no docker, o kernel monta o rootfs na modalidade somente leitura. Logo em seguida, um arquivo do sistema é criado como read-write sobre o rootf. Depois, o kernel sobrepõe a partição raiz como read-only e pega um novo file system para colocá-lo sobre o rootfs.

Observe que o container está pronto para executar a imagem (layer somente leitura) após o carregamento do rootfs. Ele também é uma camada read-write criada a partir de uma imagem somente leitura.

Por isso, ele é uma ferramenta excelente para DevOps. No caso dos administradores de sistemas há o benefício da flexibilidade, custos menores e menos áreas ocupadas. Para os desenvolvedores, é mais liberdade para que esses profissionais foquem a atividade principal.

Por que ele vem se tornando uma tendência no mercado?

O docker tem se popularizado devido a diferentes facilidades proporcionadas pela plataforma. Em comparação com a VM, a estrutura é menor, porque o hypervisor é substituído pelo docker engine e o SO hospedeiro é excluído.

Ambos os modelos permitem o isolamento de recursos e incluem as bibliotecas, aplicação e arquivos necessários. A diferença é que a VM precisa do SO e, com isso, exige espaço e possui custo de manutenção.

Essa especificação evidencia que o docker tem uma estrutura mais portátil e leve. É mais fácil mantê-la, porque os containers compartilham o mesmo SO da hospedagem. Apenas os processos são executados isoladamente e, portanto, o custo e a necessidade de espaço são menores.

A portabilidade é outra vantagem do docker, o que traz ainda mais benefícios para os ambientes de development. Assim, uma aplicação é mais facilmente executada no ambiente de homologação e/ou de produção.

Além disso, há uma garantia maior de que não haverá erros ou imprevistos durante o deploy. Por fim, gasta-se menos tempo para configurar os ambientes e se perde menos tempo analisando e identificando as diferenças existentes. O resultado é a integração das equipes de desenvolvimento e SysAdmin, que aumentam a produtividade.

Para entender melhor a importância, basta observar um exemplo prático. Imagine que a sua empresa realiza um projeto em WordPress com o PHP versão 5.\*, mas deseja atualizar a linguagem para a sétima.

Esse processo de update da versão se torna mais simples com o docker. Você pode realizar testes para definir a versão que permanecerá sendo utilizada e assinalar os ajustes necessários.

Devido a todas essas peculiaridades, centenas de empresas utilizam o docker em todo o mundo. Alguns exemplos são Uber, The Washington Post, PayPal, General Electric, eBay, Spotify etc. Um dos motivos para isso é o fato de diversas soluções de hospedagem adotarem a tecnologia.

Outra justificativa é que, no começo, ele pode representar uma demanda muito grande de recursos, mas a recompensa são as vantagens em agilidade, já que a velocidade de implantação de containers vai de milissegundos a alguns segundos.

Quais são as vantagens de utilizar?

O container docker é uma tecnologia que apresenta diferentes benefícios às empresas. As principais são:

### **Economia Significativa de Recursos**

Essa questão pode ser explicada pela contextualização da diferença existente entre imagens e containers. As primeiras são um ambiente read-only, enquanto os segundos são definidos como uma imagem em execução, na qual é gerada um layer extra que armazena os dados relativos à determinada operação.

É possível configurar a dependência de uma imagem em relação a outra. Nessa situação é criada uma pilha de imagens, sendo que cada uma delas é somente leitura. Assim, pode-se montar um amontoado para diferentes containers, o que ocasiona uma economia de recursos sobre o disco, que é compartilhado entre os ambientes.

Ainda é possível exemplificar essa situação por um ambiente composto por uma imagem Apache, Debian e módulo PHP. Se forem necessários 20 containers da pilha de imagens, não é necessário utilizar o recurso multiplicado por 20.

Isso porque a utilização do espaço em disco se refere somente aos logs e arquivos temporários de cada container. Desse modo, os dados são armazenados no layer extraindividual.

### **Disponibilidade Maior do Sistema**

Esse benefício é ocasionado pelo fato de o docker virtualizar o sistema de maneira diferente ao método da VM. Por compartilhar o SO e outros componentes, há mais espaço livre, o que deixa os processos mais ágeis e oferece uma disponibilidade maior.

Em termos bastante simples, a máquina fica menos “pesada” e fica com mais espaço para rodar outros programas e aplicações.

### **Possibilidade de Compartilhamento**

Os arquivos podem ser compartilhados entre o host e o container ou até mesmo um volume tem a possibilidade de ser distribuído para outros. A prática da segunda situação é mais indicada nos casos em que se deseja ter persistência de dados e quando não se atrela ao host que hospeda o container.

Outro aspecto que precisa ser considerado é que o container está em um nível de virtualização operacional, ou seja, é um processo em execução em um kernel compartilhado entre outros containers.

Por meio do namespace é feito o isolamento da memória RAM, disco, processamento e acesso à rede. Ou seja, em um contexto isolado há o compartilhamento do kernel, mas a impressão é de que o SO é dedicado.

Por fim, o compartilhamento pode ocorrer pela cloud. Essa tecnologia disponibiliza um repositório de imagens e ambientes prontos, que podem ser compartilhados. Com a utilização desse serviço, o docker chega a extrapolar o limite técnico e passa para questões de gerência, processo e update do ambiente.

Com isso, torna-se mais fácil compartilhar as modificações e oferecer uma gestão centralizada em relação às definições de ambiente. O espaço para testes também se torna mais leve, o que permite baixar uma solução durante uma reunião, por exemplo, ou fornecer um padrão de melhores práticas para todos os colaboradores.

### **Gerenciamento Facilitado**

Os containers são executados em máquinas físicas ou virtuais e o grupo delas é chamado de cluster. Esse item precisa ser monitorado constantemente. Por isso, foram criadas ferramentas para fazer o gerenciamento, por exemplo, o Kubernetes e o OpenShift.

Esses sistemas trabalham em conjunto com o docker e operam o equipamento que possibilita a execução dos containers. No entanto, os sistemas de arquivos também são gerenciados.

Essas ferramentas de monitoramento geram uma abstração, denominada pod, no nível de um componente da aplicação. Essa questão inclui um grupo de um ou mais containers, armazenamento compartilhado e alternativas de operação.

### **Similaridade Dos Ambientes**

A transformação da aplicação em uma imagem docker permite que ela seja instanciada como container em diferentes ambientes. Essa característica garante a sua utilização, por exemplo, tanto no notebook do desenvolvedor quando no servidor de produção.

Tenha em mente que a imagem aceita parâmetros na iniciação do container, situação que indica diferentes comportamentos conforme o ambiente. Por exemplo: ele pode se conectar ao banco de dados local para testes a partir da base de dados e credenciais, mas em produção acessará um database com infraestrutura robusta, que possui as suas próprias credenciais.

É importante destacar que ambientes semelhantes impactam a análise de erros e a confiabilidade do processo de entrega contínua de forma positiva. No segundo caso, a base é a criação de um artefato único que faz a migração, que pode ser a própria imagem do docker com as dependências necessárias para a execução do código dinâmico ou compilado.

### **Aplicação Como Pacote Completo**

As imagens do docker possibilitam o empacotamento da aplicação e as suas dependências, o que simplifica o processo de distribuição por não ser exigida ampla documentação sobre a configuração da infraestrutura com a finalidade de execução. Para isso, basta disponibilizar o repositório e permitir o acesso para o usuário.

A partir disso é possível fazer o download do pacote, que pode ser executado facilmente. O update também sofre impacto positivo, porque a estrutura de layers permite que, em caso de modificação, somente a alteração seja transferida.

Essa medida faz com que o ambiente possa ser mudado de maneira simples e rápida. Com apenas um comando existe a capacidade de fazer o update da imagem da aplicação, que é refletida no container em execução quando for desejado.

As imagens podem ser categorizadas com tags, o que facilita o armazenamento de diferentes versões de uma aplicação. Se houver algum problema na atualização, é possível retornar para a imagem com a tag anterior.

### **Padronização e Replicação**

As imagens do docker são construídas por meio de arquivos de definição, o que assegura o seguimento de um determinado padrão e eleva a confiança na replicação. Dessa maneira, fica muito mais viável escalar a estrutura.

Com a chegada de um novo membro para a equipe de TI, por exemplo, ele pode se integrar e receber o ambiente de trabalho rapidamente, com apenas alguns comandos. Ele ainda pode desenvolver os códigos de acordo com o padrão adotado para a equipe.

Por sua vez, na necessidade de testar uma versão com as imagens, é possível mudar um ou mais parâmetros de um arquivo de definição. Ou seja, é mais simples criar e mudar a infraestrutura.

### **Possibilidade de Acessar Comunidade**

O repositório de imagens docker pode ser acessado para conseguir modelos de infraestrutura de aplicações e serviços prontos para integrações complexas. Dois exemplos são o mysql como banco de dados e o nginx como proxy reverso.

No caso de a aplicação exigir esses recursos, você pode apenas usar as imagens do repositório e configurar os parâmetros para adequação ao seu ambiente. Essa é uma vantagem principalmente porque as imagens oficiais costumam ser condizentes com as boas práticas.

Como você pôde perceber, a tecnologia do docker é um pouco complexa, mas vale a pena apostar nela. A sua empresa consegue economizar recursos, configurar parâmetros e virtualizar em um nível muito mais alto que o permitido pelas VMs atualmente.

Docker se refere a muitas coisas. Isso inclui: um projeto da comunidade open source; as ferramentas resultantes desse projeto; a empresa Docker Inc., principal apoiadora do projeto; e as ferramentas compatíveis formalmente com a empresa. O fato de que as tecnologias e a empresa têm o mesmo nome pode causar uma certa confusão.

Veja uma simples explicação:

O software de TI "Docker" é uma tecnologia de containerização para criação e uso de containers Linux®.

A comunidade open source do Docker trabalha gratuitamente para melhorar essas tecnologias para todos os usuários.

A empresa Docker Inc. se baseia no trabalho realizado pela comunidade do Docker, tornando-o mais seguro, e compartilha os avanços com a comunidade em geral. Depois, ela oferece aos clientes corporativos o suporte necessário para as tecnologias que foram aprimoradas e fortalecidas.

Com o Docker, é possível lidar com os containers como se fossem máquinas virtuais modulares e extremamente leves. Além disso, os containers oferecem maior flexibilidade para você criar, implantar, copiar e migrar um container de um ambiente para outro. Isso otimiza as aplicações na cloud.

Certo. Mas o que são containers Linux?

### Como o Docker funciona?

A tecnologia Docker usa o kernel do Linux e recursos do kernel como Cgroups e namespaces para segregar processos. Assim, eles podem ser executados de maneira independente. O objetivo dos containers é criar essa independência: a habilidade de executar diversos processos e aplicações separadamente para utilizar melhor a infraestrutura e, ao mesmo tempo, manter a segurança que você teria em sistemas separados.

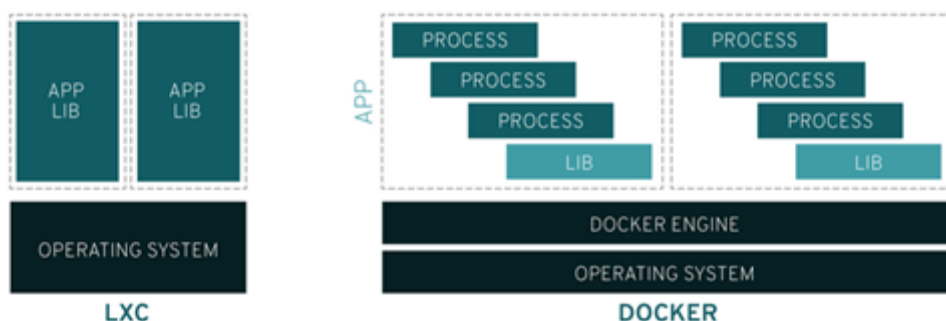
As ferramentas de container, incluindo o Docker, fornecem um modelo de implantação com base em imagem. Isso facilita o compartilhamento de uma aplicação ou conjunto de serviços, incluindo todas as dependências deles em vários ambientes. O Docker também automatiza a implantação da aplicação (ou de conjuntos de processos que constituem uma aplicação) dentro desse ambiente de container.

Essas ferramentas baseadas nos containers Linux (o que faz com que o Docker seja exclusivo e fácil de usar) oferecem aos usuários acesso sem precedentes a aplicações, além da habilidade de implantar com rapidez e de ter total controle sobre as versões e distribuição.

O Docker utiliza a mesma tecnologia que os containers Linux tradicionais?

Não, a tecnologia Docker foi desenvolvida inicialmente com base na tecnologia LXC, que a maioria das pessoas associa aos containers Linux "tradicionais". No entanto, desde então, essa tecnologia tornou-se independente. O LXC era útil como uma virtualização leve, mas não oferecia uma boa experiência para usuários e desenvolvedores. A tecnologia Docker oferece mais do que a habilidade de executar containers: ela também facilita o processo de criação e construção de containers, o envio e o controle de versão de imagens, dentre outras coisas.

### Traditional Linux containers vs. Docker



Os containers Linux tradicionais usam um sistema init capaz de gerenciar vários processos. Isso significa que aplicações inteiras são executadas como uma. A tecnologia Docker incentiva que as aplicações sejam segregadas em processos separados e oferece as ferramentas para fazer isso. Essa abordagem granular tem algumas vantagens.

### As Vantagens dos Containers Docker

#### Modularidade

A abordagem do Docker para a containerização se concentra na habilidade de desativar uma parte de uma aplicação, seja para reparo ou atualização, sem interrompê-la totalmente. Além dessa abordagem baseada em microsserviços, é possível compartilhar processos entre várias aplicações da mesma maneira como na arquitetura orientada a serviço (SOA).

#### Camadas e Controle de Versão de Imagens

Cada arquivo de imagem Docker é composto por uma série de camadas. Elas são combinadas em uma única imagem. Uma nova camada é criada quando há alteração na imagem. Toda vez que um usuário especifica um comando, como executar ou copiar, uma nova camada é criada.

O Docker reutiliza essas camadas para a construção de novos containers, o que torna o processo de criação muito mais rápido. As alterações intermediárias são compartilhadas entre imagens, o que melhora ainda mais a velocidade, o tamanho e a eficiência. O controle de versões é inerente ao uso de camadas. Sempre que é realizada uma nova alteração, é gerado um changelog integrado, o que fornece controle total sobre as imagens do container.

### Reversão

Talvez a melhor vantagem da criação de camadas seja a habilidade de reverter quando necessário. Toda imagem possui camadas. Não gostou da iteração atual de uma imagem? Simples, basta reverter para a versão anterior. Esse processo é compatível com uma abordagem de desenvolvimento ágil e possibilita as práticas de integração e implantação contínuas (CI/CD) em relação às ferramentas.

### Implantação Rápida

Antigamente, colocar novo hardware em funcionamento, provisionado e disponível, levava dias. E as despesas e esforço necessários para mantê-lo eram onerosos. Os containers baseados em docker podem reduzir o tempo de implantação de horas para segundos. Ao criar um container para cada processo, é possível compartilhar rapidamente esses processos similares com novos aplicativos. Como não é necessário inicializar um sistema operacional para adicionar ou mover um container, o tempo de implantação é substancialmente menor. Além disso, com a velocidade de implantação, é possível criar dados e destruir os criados pelos containers sem nenhuma preocupação e com facilidade e economia.

Em resumo, a tecnologia Docker é uma abordagem mais granular, controlável e baseada em micros-serviços que valoriza a eficiência.

Há limitações no uso do docker?

Por si só, o Docker é excelente para gerenciar containers únicos. No entanto, quando você começa a usar cada vez mais containers e aplicações em containers segregados em centenas de partes, o gerenciamento e a orquestração podem se tornar um grande desafio. Eventualmente, será necessário recuar e agrupar os containers para oferecer serviços como rede, segurança, telemetria etc. em todos eles. É aí que o Kubernetes entra em cena.

O Docker não fornece as mesmas funcionalidades parecidas com UNIX que os containers Linux tradicionais oferecem. Isso inclui a capacidade de usar processos como cron ou syslog dentro do container, junto à aplicação.

O Docker também tem algumas limitações em questões como a limpeza de processos netos (grand-child) após o encerramento dos processos filhos (child), algo que é processado de forma natural nos containers Linux tradicionais. Essas desvantagens podem ser mitigadas ao modificar o arquivo de configuração e configurar essas funcionalidade desde o início, algo que não está imediatamente óbvio em um primeiro momento.

Além disso, há outros subsistemas e dispositivos do Linux sem espaço de nomes. Incluindo os dispositivos SELinux, Cgroups e /dev/sd\*. Isso significa que, se um invasor adquirir controle sobre esses subsistemas, o host será comprometido. Para manter-se leve, o compartilhamento do kernel do host com os containers gera a possibilidade dessa vulnerabilidade na segurança. Isso é diferente nas máquinas virtuais, que são mais firmemente segregadas a partir do sistema host.

O daemon do Docker também pode representar uma vulnerabilidade à segurança. Para usar e executar os containers Docker, é provável que você use o daemon do Docker, um ambiente de execução persistente para containers.

O daemon do Docker requer privilégios de raiz. Portanto, é necessário ter um cuidado maior ao escolher as pessoas que terão acesso a esse processo e o local onde ele residirá. Por exemplo, um daemon local tem menos chances de sofrer um ataque do que um daemon em um local mais público, como um servidor web.

Docker é um projeto OpenSource que fornece uma plataforma para desenvolvedores e administradores de sistemas permitindo que se crie containers leves e prostrátil de diversas aplicações.



Sua funcionalidade permite adicionar e simplificar o uso, dos linux containers (LXC), que são, basicamente, uma forma de isolamento de processo e sistemas, quase como virtualização, porém mais leve e integrada ao host. O Docker permite criar aplicações e “containers” que isolam o S.O base e todo a pilha de dependências de seu app (libs, servidores e etc) de forma leve em espaço e performance.

O Docker também trabalha com um sistema de arquivos “empilháveis”, o aufs, que permite que os passos para configuração do seu container funcionem de forma incremental e “cacheable”, mais ou menos como “commits” do GIT.

Em poucas palavras, o Docker oferece a você um conjunto completo de ferramentas de alto nível para transportar tudo que constitui uma aplicação entre sistemas e máquinas – virtual ou física – e trás consigo grandes benefícios agregados.

O que é um Container?

O Container nada mais é que um chroot. Nele é possível definir recursos como memória, rede, sistema operacional, aplicação, serviço e etc. Em um Container Docker é possível fazer testes, desenvolvimentos, estudos, etc. Além disso, também é possível utilizá-lo em um ambiente de produção.

Qual é sua base?

O Docker tem como base o LXC Linux Container.

Linux Containers (LXC) é um método de virtualização em nível de sistema operacional para executar vários sistemas Linux isolados (contêineres) em um único host de controle (LXC host).

LXC não fornece uma máquina virtual, mas fornece um ambiente virtual que tem sua própria CPU, memória, bloco I/O, rede, espaço, entre outros. Este é fornecido por cgroups recursos no kernel do Linux no host LXC. É semelhante a um chroot, mas oferece muito mais isolamento.

Linux Containers (LXC).

Permite rodar um Linux dentro de outro Linux.

Chroot on Steroids.

Dentro do container, parece uma VM.

Não é Para-Virtualização.

Fora do Containers o Docker não passa de mais um processo para o sistema operacional.

Porque usar Containers?

Velocidade;

Boot em questão de segundos;

Economia de recursos;

Os processos rodando dentro de um container são vistos como um processo no sistema Host;

É possível subir vários containers ao mesmo tempo, consumindo o mínimo de recursos do hardware físico ou virtual.

Diferenças entre o Sistema Operacional comum/virtualizado com o Container Docker.

Sistema Operacional

Bootfs – Boot FileSystem;

Bootloader e Kernel;

Rootfs – Root Filesystem;

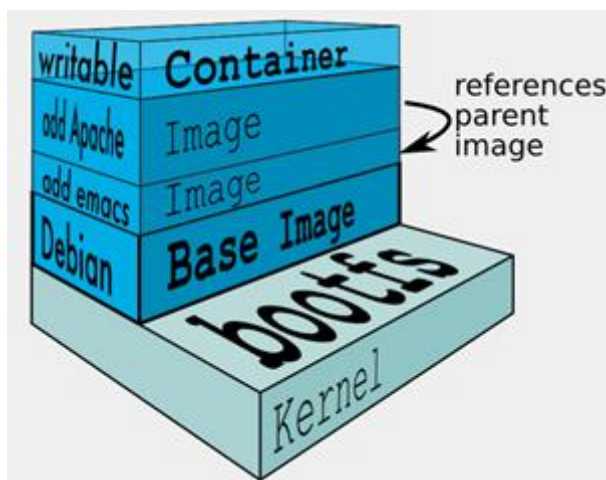
Restante dos arquivos do sistema.

Container Docker

Bootfs – (LXC, aufs/btrfs) – Mesmo Kernel da máquina.

Rootfs – Sistema de arquivos.

Com o carregamento do rootfs, o Container Docker ficará pronto para execução. O Docker está totalmente baseado no conceito de Layers.



Em um boot tradicional do Linux:

O Kernel monta o rootfs como read-only, checa sua integridade e faz a montagem como read-write.

No Docker:

O Kernel monta o rootfs como read-only, depois outro filesystem é montado como read-write em cima do rootf

Unio File System. ( O Kernel monta a partição raiz como somente leitura, só que ao invés de remontar a partição como read-write, ele se encarrega de pegar um outro FileSystem e em seguida o coloca acima do primeiro rootfs.

Observação:

No Docker após o carregando o rootfs o container estará pronto para executar a imagem;

Image – Layer somente leitura ( Imagem base de um sistema operacional – ‘snapshot’ );

O Container é uma layer read-write montada a partir de uma image que é read-only;

No Docker uma image é um layer read-only;

Uma image é uma herança de imagem;

Uma base image é quando uma imagem não possui um Pai (única).

Como é construído um Container?

O container é construído usando namespaces, cgroups, chroot entre outras funcionalidades do kernel para construir uma área isolada para sua aplicação.

O que é o Registry (Registro)?

O registro é um repositório provido pelo Docker. Ele se encontra na nuvem, e disponibiliza uma área para envio, download e compartilhamento de imagens (snapshots) de Containers.



## *Conselho Nacional de Justiça*

### **RESOLUÇÃO Nº 91, DE 29 DE SETEMBRO DE 2009.**

Institui o Modelo de Requisitos para Sistemas Informatizados de Gestão de Processos e Documentos do Poder Judiciário e disciplina a obrigatoriedade da sua utilização no desenvolvimento e manutenção de sistemas informatizados para as atividades judiciárias e administrativas no âmbito do Poder Judiciário.

**O PRESIDENTE DO CONSELHO NACIONAL DE JUSTIÇA**, no uso de suas atribuições legais, tendo em vista o decidido no Processo nº 2009.10.00.005334-8, em sessão realizada no dia 29 de setembro de 2009, e

**CONSIDERANDO** a necessidade de estabelecer requisitos mínimos para os sistemas informatizados do Poder Judiciário e garantir a confiabilidade, autenticidade e acessibilidade dos documentos geridos por esses sistemas;

**CONSIDERANDO** a necessidade de uniformizar regras mínimas de produção, tramitação, guarda, destinação, armazenamento, preservação, recuperação, arquivamento e recebimento de processos e outros documentos digitais, não-digitais ou híbridos geridos pelos sistemas informatizados do Poder Judiciário;

A blue ink signature, appearing to be a stylized 'C' or similar character, located in the bottom right corner of the page.

## RESOLVE:

Art. 1º Aprovar o Modelo de Requisitos para Sistemas Informatizados de Gestão de Processos e Documentos do Poder Judiciário – MoReq-Jus.

Art. 2º Os novos sistemas a serem desenvolvidos ou adquiridos para as atividades judiciárias e administrativas do Conselho e dos órgãos integrantes do Poder Judiciário deverão aderir integralmente aos requisitos do MoReq-Jus.

Parágrafo Único. Para os fins dispostos no presente artigo, as especificações para desenvolvimento ou aquisição de sistemas, bem como o documento de visão respectivos, deverão fazer menção expressa ao grau de adesão ao MoReq-Jus, em observância ao sistema de avaliação de conformidade a ser disciplinado pelo CNJ.

Art. 3.º Os sistemas legados que ora servem às atividades judiciárias e administrativas do Conselho e dos órgãos integrantes do Poder Judiciário deverão aderir ao MoReq-Jus, conforme o seguinte cronograma:

I – adesão aos requisitos de “organização dos documentos institucionais: plano de classificação e manutenção de documentos” (capítulo 2), “preservação” (capítulo 5) e “segurança” (capítulo 6) “avaliação e destinação” (capítulo 8), até dezembro de 2012;

II – adesão aos demais requisitos até dezembro de 2014.

Art. 4º O Departamento de Pesquisas Judiciárias e o Departamento de Tecnologia da Informação do CNJ serão responsáveis pela coordenação do programa de melhoria contínua do MoReq-Jus e pelo processo de acompanhamento e de validação do grau de aderência dos sistemas ao referido modelo.

Parágrafo único. O programa de melhoria contínua incluirá:

I - os metadados dos sistemas aplicativos das instituições do Poder Judiciário;



II - o sistema de acompanhamento e avaliação de conformidade dos novos sistemas e dos sistemas legados ao MoReq-Jus;

III – a permanente atualização do MoReq-Jus.

Art. 5º Esta Resolução entra em vigor na data de sua publicação.

**Ministro GILMAR MENDES**

A large, stylized handwritten signature in blue ink is written over the text 'Ministro GILMAR MENDES'. The signature consists of several loops and curves, starting from the top left, looping around the text, and ending with a long vertical stroke on the right side.

**Modelo de Requisitos  
para Sistemas Informatizados de Gestão de  
Processos e Documentos do Judiciário brasileiro.  
MoReq-Jus**

**Versão 1.0**

Brasília, agosto de 2009.

# Sumário

<b>1</b>	<b>Introdução</b> .....	<b>8</b>
1.1	Objetivos.....	10
1.2	Utilização.....	10
1.3	Organização do MoReq-Jus.....	11
1.4	Gestão de processos e documentos no Judiciário brasileiro.....	12
<b>2</b>	<b>Organização dos documentos institucionais: plano de classificação e manutenção dos documentos</b> .....	<b>22</b>
2.1	Configuração e administração do plano de classificação no GestãoDoc.....	22
2.2	Classificação e metadados dos processos/dossiês.....	25
2.3	Gerenciamento dos processos/dossiês.....	25
2.4	Processos.....	27
2.5	Volumes: abertura, encerramento e metadados.....	29
2.6	Manutenção de documentos institucionais não-digitais e híbridos.....	30
<b>3</b>	<b>Captura</b> .....	<b>31</b>
3.1	Captura: procedimentos gerais.....	35
3.2	Captura em lote.....	39
3.3	Captura de mensagens de sistema de comunicação eletrônica.....	39
3.4	Formato de arquivo e estrutura dos documentos a serem capturados.....	40
3.5	Estrutura dos procedimentos de gestão.....	41
<b>4</b>	<b>Armazenamento</b> .....	<b>42</b>
4.1	Durabilidade.....	44
4.2	Capacidade.....	45
4.3	Efetividade de armazenamento.....	47
<b>5</b>	<b>Preservação</b> .....	<b>47</b>
5.1	Aspectos físicos.....	49
5.2	Aspectos lógicos.....	50
5.3	Aspectos gerais.....	51
<b>6</b>	<b>Segurança</b> .....	<b>51</b>
6.1	Cópias de segurança.....	55

6.2	Controle de acesso .....	55
6.3	Classificação da informação quanto ao grau de sigilo e restrição de acesso à informação sensível .....	58
6.4	Trilha de auditoria .....	60
6.5	Assinaturas digitais .....	62
6.6	Criptografia .....	64
6.7	Marcas d'água digitais.....	67
6.8	Acompanhamento de transferência.....	67
6.9	Autoproteção.....	68
6.10	Alteração, ocultação e exclusão de documentos institucionais.....	69
<b>7</b>	<b>Tramitação e fluxo de trabalho.....</b>	<b>70</b>
7.1	Controle do fluxo de trabalho .....	71
7.2	Controle de versões e do <i>status</i> do documento.....	73
<b>8</b>	<b>Avaliação e destinação .....</b>	<b>74</b>
8.1	Configuração dos instrumentos de classificação, temporalidade e destinação de documentos .....	76
8.2	Aplicação dos instrumentos de classificação, temporalidade e destinação de documentos .....	78
8.3	Exportação de documentos.....	79
8.4	Eliminação .....	81
8.5	Avaliação e destinação de documentos institucionais não-digitais e híbridos.....	82
<b>9</b>	<b>Pesquisa, localização e apresentação de documentos .....</b>	<b>83</b>
9.1	Recuperação de informação .....	83
9.2	Pesquisa e localização.....	83
9.3	Apresentação: texto, imagem, som e vídeo.....	85
<b>10</b>	<b>Funções administrativas .....</b>	<b>87</b>
10.1	Monitoração do sistema .....	87
10.2	Manutenção e evolução .....	88
<b>11</b>	<b>Usabilidade .....</b>	<b>88</b>
<b>12</b>	<b>Interoperabilidade .....</b>	<b>92</b>



13	Disponibilidade .....	93
14	Desempenho e escalabilidade .....	94
15	Glossário.....	95
16	Modelos de referência, legislação, regulamentações, normas e referências bibliográficas .....	100
16.1	Modelos de requisitos para sistemas informatizados de gestão arquivística de documentos .....	100
16.2	Legislação federal .....	101
16.3	Resoluções do Conselho Nacional de Arquivos — Conarq .....	102
16.4	Normas	102
16.5	Referências bibliográficas .....	103
<b>Apêndice 1 - Metadados .....</b>		<b>105</b>
1.1	Elementos de metadados de segurança .....	108
1.2	Elementos de metadados de auditoria .....	125
1.3	Elementos de metadados de preservação .....	144

# Apresentação

O Modelo de Requisitos para Sistemas Informatizados de Gestão de Processos e Documentos do Judiciário brasileiro (MoReq-Jus) foi elaborado por um grupo de trabalho interdisciplinar integrado por especialistas (servidores e magistrados) das áreas de Ciência da Informação, Tecnologia da Informação e Direito, e decorreu da necessidade de se estabelecer requisitos mínimos para os sistemas informatizados do Poder Judiciário, de forma a garantir a confiabilidade, a autenticidade e a acessibilidade dos documentos e processos geridos por esses sistemas. Essa iniciativa teve como referência:

O Modelo de Requisitos para Sistemas Informatizados de Gestão de Processos e Documentos da Justiça Federal (MoReq-Jus) aprovado pela Resolução nº. 7, do Conselho da Justiça Federal, de 7 de abril de 2008, que disciplinou a obrigatoriedade de sua utilização no desenvolvimento de novos sistemas informatizados para as atividades judiciárias e administrativas, no âmbito do Conselho e da Justiça Federal de primeiro e segundo graus.

O Modelo de Requisitos para Sistemas Informatizados de Gestão Arquivística de Documentos (e-ARQ Brasil), elaborado pela Câmara Técnica de Documentos Eletrônicos do Conselho Nacional de Arquivos (Conarq), aprovado pela Resolução nº. 25, de 27 de abril de 2007;

O Modelo de Requisitos para Gestão de Arquivos Eletrônicos (MoReq), desenvolvido pelo Instituto dos Arquivos Nacionais/Torre do Tombo de Portugal. A versão do MoReq em Língua Portuguesa, por sua vez, teve como base o *Model Requirements for the Management of Electronic Records* (MoReq), elaborado pelo programa Intercâmbio de Dados entre Administrações (IDA) da Comissão Européia.

*Model Requirements for the Management of Electronic Records* (MoReq), elaborado pelo Cornwell Management Consultants para a Comissão Européia, e no *Model Requirements Specification for the Management of Electronic Records – MoReq2*;

PREMIS (*Preservation Metadata: Implementation Strategies*) Working Group, versão de Março de 2008.

Os modelos mencionados têm como objetivos comuns fornecer requisitos para a aquisição, desenvolvimento ou a avaliação de sistemas de gestão de processos e documentos:

Digitais — Os metadados e os próprios documentos são inseridos no sistema;

Não-digitais — O sistema registra apenas os metadados dos documentos e

Híbridos — Possibilita a gestão de documentos não-digitais e digitais.

A exemplo dos modelos mencionados, a existência de um programa de gestão de documentos institucionais (arquivísticos) na instituição é um dos requisitos para a utilização do MoReq-Jus, que se aplica aos sistemas informatizados relativos às atividades-meio e às atividades-fim de todo o Poder Judiciário.

Por se tratar de uma adaptação, na elaboração do MoReq-Jus não se mencionou a fonte ou referência do texto, em sua maioria extraído do e-ARQ, do MoReq-Jus (Justiça Federal) e do MoReq (Europeu). A adoção do MoReq-Jus garantirá a uniformização da

produção, da tramitação, da guarda, da destinação, do armazenamento, da preservação, da recuperação, do arquivamento e do recebimento de processos e de outros documentos digitais, não-digitais ou híbridos geridos pelos sistemas informatizados.

Além dos documentos mencionados, o grupo baseou seus trabalhos na Constituição Federal, em especial ao art. 216, § 2º, que estabelece como competência da Administração Pública a gestão da documentação governamental e as providências para franquear sua consulta a quantos dela necessitem; na Lei nº. 8.159, de 1991, que dispõe sobre a política nacional de arquivos públicos e privados, conceitua gestão de documentos (art. 3º) como o conjunto de procedimentos e operações técnicas que engloba a produção, a tramitação, o uso, a avaliação e o arquivamento em fase corrente e intermediária, visando a sua eliminação ou recolhimento para guarda permanente; e na Lei nº. 11.419, de 2006, que dispõe sobre a informatização do processo judicial; e, legislação processual vigente.

A gestão de documentos não é um problema exclusivamente das unidades de arquivo, ela visa garantir a produção, a guarda e o acesso aos documentos durante o seu ciclo de vida. Portanto, envolve os diversos atores e unidades da instituição e precisa atender às demandas de todos. Dessa forma, torna-se evidente que os sistemas informatizados de gestão de documentos administrativos e de acompanhamento processual são também sistemas de gestão de documentos e que a gestão do documento permeia todo o andamento processual: da distribuição do processo — fase da produção do documento, passando por toda a tramitação — até sua destinação final, depois da baixa definitiva.

Atualmente, a preservação digital constitui uma preocupação crescente dos profissionais ligados à gestão da informação. Na área Jurídica, em razão do valor probatório da informação, é preciso, cada vez mais, garantir o acesso continuado aos acervos digitais, em condições que assegurem a sua autenticidade a médio e longo prazo. Neste contexto, os requisitos do MoReq-Jus contribuirão para a aquisição, desenvolvimento ou avaliação de aplicativos que garantam a preservação dos documentos, a interoperabilidade e a segurança dos sistemas e a implementação de um plano de preservação digital para o Judiciário brasileiro.

Na sociedade atual são nítidos a redução de custos e o aumento de eficácia em decorrência da informação em suportes digitais. A importância dos meios e técnicas digitais de codificação, de armazenamento e de transporte da informação é amplamente reconhecida e, no entanto, não são elucidados os perigos associados ao caráter efêmero dessa informação e as dificuldades para a sua preservação.

A preservação da informação impressa é concentrada na preservação do papel, uma vez que nele está o texto, a estrutura e de certa forma também o contexto do documento.

Em meio digital, a preservação digital compreende a preservação física, lógica e intelectual dos documentos digitais.

A preservação física está focalizada nos conteúdos armazenados em suportes magnéticos (cassetes VHS, cassetes de música, etc.) e nos suportes ópticos (CD-ROM's, discos WORM, etc.) que levam à necessidade de definição de regras para a migração dos formatos em que os documentos estão registrados.

A preservação lógica procura na tecnologia formatos atualizados para a introdução dos dados (material audiovisual, correio eletrônico, etc.) e novas aplicações de hardware e

software que mantenham em atividade os seus bits, para conservar a sua capacidade de leitura.

Assim, a busca por estratégias de preservação digital requer não apenas procedimentos de manutenção e recuperação de dados, no caso de perdas acidentais, para resguardar a mídia e seu conteúdo, mas também estratégias e procedimentos para manter sua acessibilidade e autenticidade através do tempo, o que requer a aplicação de padrões de metadados e documentação.

A longo prazo, o fracasso na preservação dos documentos digitais acarretará na perda irreversível do registro, da prova, do testemunho, da memória. Assim, a questão da preservação pode ter um impacto negativo na memória coletiva, pública e privada da sociedade, com repercussão em questões legais, comerciais e organizacionais.

No Judiciário, com a crescente produção de documentos e processos em meio digital torna-se premente a definição de estratégias que garantam a preservação, a regulamentação de questões associadas ao valor probatório, a uniformização de procedimentos definidos normativamente e a definição de parâmetros para a certificação de qualidade.

As questões inerentes à preservação digital não se restringem somente à seleção do que deve ser preservado, mas garantir a confiabilidade, a autenticidade e a acessibilidade dos documentos e processos geridos pelos sistemas informatizados do Judiciário brasileiro. Esta é a grande iniciativa que o Conselho Nacional de Justiça dá efetividade às recomendações da *Carta da UNESCO para a Preservação do Patrimônio Arquivístico Digital* de forma a minimizar os efeitos da fragilidade e da obsolescência de *hardware*, *software* e formatos e que assegurem, ao longo do tempo o acesso contínuo e o uso pleno da informação a todos os segmentos da sociedade. Isso só será possível se houver uma ampla articulação entre os diversos setores comprometidos com a preservação do patrimônio arquivístico digital e cooperação com os organismos nacionais e internacionais<sup>1</sup>.

---

<sup>1</sup> UNESCO. **Carta sobre a Preservação do Patrimônio Digital**. Brasília. 2003. Acesso em 28 de junho de 2009. Disponível em <http://www.brasilia.unesco.org/publicacoes/livros/cartapatrimonioarquivistico>.

# 1 Introdução

O Modelo de Requisitos para Sistemas Informatizados de Gestão de Processos e Documentos do Judiciário brasileiro (MoReq-Jus) estabelece condições a serem cumpridas na produção, na tramitação, na guarda, no armazenamento, na preservação, no arquivamento ou no recebimento de documentos, pelos sistemas de gestão de processos e documentos digitais, não-digitais ou híbridos, a fim de garantir a sua confiabilidade e autenticidade, assim como o seu acesso.

O MoReq-Jus estabelece processos e requisitos mínimos para um Sistema Informatizado de Gestão de Processos e Documentos (GestãoDoc), independentemente da plataforma tecnológica em que for desenvolvido e implantado.

Um GestãoDoc deve ser capaz de gerenciar simultaneamente os documentos e processos digitais, não-digitais e híbridos. Para os documentos não-digitais, o sistema registra apenas as referências a esses documentos e as operações de produção, de tramitação, de guarda, de armazenamento, de preservação, de arquivamento e de recebimento. No caso dos sistemas de documentos digitais, este registra os documentos e as operações mencionadas.

A produção de documentos digitais levou à criação de sistemas de gerenciamento de documentos. Entretanto, para assegurar que documentos digitais sejam confiáveis e autênticos e que possam ser preservados com essas características, é fundamental que esses sistemas incorporem os conceitos arquivísticos e suas implicações no gerenciamento dos documentos digitais.

Para o bom entendimento desse Modelo de Requisitos, alguns conceitos que nortearam o trabalho foram relacionados a seguir. Outras definições operacionais foram incluídas no Capítulo 15, Glossário.

## **Sistema de Informação**

Conjunto organizado de políticas, procedimentos, pessoas, equipamentos e programas computacionais que produzem, processam, armazenam e provêem acesso à informação proveniente de fontes internas e externas para apoiar o desempenho das atividades do Judiciário brasileiro.

## **Gestão de Documentos**

Conjunto de procedimentos e operações técnicas referentes à produção, tramitação, uso, avaliação e arquivamento dos documentos em fase corrente e intermediária, visando sua eliminação ou seu recolhimento para a guarda permanente.

## **Sistema de Gestão de Documentos**

Conjunto de procedimentos e operações técnicas, cuja interação permite a eficiência e a eficácia da gestão de processos e documentos.

### **Gerenciamento Eletrônico de Documentos (GED)**

Conjunto de tecnologias utilizadas para organização da informação não-estruturada das instituições do Judiciário brasileiro, que pode ser dividido nas seguintes funcionalidades: captura, gerenciamento, armazenamento e distribuição. Entende-se por informação não-estruturada a mensagem de correio eletrônico, o arquivo de texto, imagem ou som, planilhas etc.

O GED engloba tecnologias de digitalização, automação de fluxos de trabalho (*workflow*), processamento de formulários, indexação, entre outras.

### **Sistema Informatizado de Gestão de Processos e Documentos (GestãoDoc)**

Sistema mais abrangente que o GED, desenvolvido para produzir, gerenciar a tramitação, receber, armazenar, dar acesso e destinar documentos em ambiente eletrônico. Pode compreender um *software* particular, um determinado número de *softwares* integrados — adquiridos ou desenvolvidos — ou uma combinação desses. Envolve um conjunto de procedimentos e operações técnicas característicos do sistema de gestão de processos e documentos, processado eletronicamente e aplicável em ambientes digitais ou em ambientes híbridos — documentos digitais e não-digitais ao mesmo tempo.

Um GestãoDoc inclui diversas operações, tais quais, produção do documento, controle de sua tramitação, aplicação do plano de classificação, controle de versões, controle sobre os prazos de guarda e destinação, armazenamento seguro e procedimentos que garantam o acesso e a preservação a médio e longo prazo de documentos digitais e não-digitais confiáveis, íntegros e autênticos.

No caso dos documentos digitais, um GestãoDoc deve abranger todos os tipos de documentos digitais das instituições do Judiciário brasileiro.

A partir dessas definições podemos fazer as seguintes considerações:

- Um sistema de informação pode abarcar todas as fontes de informação existentes nas instituições do Judiciário brasileiro, incluindo, principalmente, os sistemas de gestão de processos judiciais, administrativos e de documentos.
- O GestãoDoc mantém a organicidade dos documentos e a inter-relação destes com as atividades da instituição.
- A concepção de um GestãoDoc, por ser um sistema de gestão de processos e documentos, tem de dar-se concomitantemente com a adoção de uma política de gestão de documentos.
- O ciclo de vida dos documentos refere-se às sucessivas etapas pelas quais passam: produção, tramitação, uso, avaliação, arquivamento e destinação (guarda permanente, devolução às partes ou eliminação).

Requisitos que caracterizam um GestãoDoc:

- Captura, armazenamento, indexação e recuperação de todos os tipos de documentos institucionais e de todos os componentes digitais do documento institucional como uma unidade complexa<sup>1</sup>.
- Gestão dos documentos a partir de sua classificação para manter a relação orgânica entre eles.

- Implementação de metadados associados aos documentos para descrever o contexto em que se inserem (jurídico-administrativo, de proveniência, de procedimentos, documental e tecnológico).
- Integração entre documentos digitais e documentos não-digitais.
- Armazenamento seguro para garantir a integridade dos documentos.
- Avaliação e seleção dos documentos para recolhimento e preservação daqueles considerados de valor permanente.
- Aplicação de critérios de classificação e guarda.
- Exportação dos documentos para transferência e recolhimento.
- Instrumentos para gestão de estratégias de preservação dos documentos.

As especificações dos requisitos funcionais e não-funcionais de sistemas e dos metadados a serem implementados nos diferentes tipos de GestãoDoc existentes nas instituições do Judiciário brasileiro não estão incluídas neste documento, não obstante, devam ser detalhadas para cada sistema específico.

## 1.1 Objetivos

O MoReq-Jus tem por objetivo fornecer especificações técnicas e funcionais, para orientar a aquisição, o detalhamento e o desenvolvimento de sistemas de gestão de processos e documentos no âmbito do Judiciário brasileiro. Também tem por objetivo estabelecer critérios para certificação do grau de aderência ao modelo.

## 1.2 Utilização

O MoReq-Jus é especialmente dirigido a:

- Potenciais usuários de um GestãoDoc — Na elaboração de um edital de licitação para a apresentação de propostas de fornecimento de *software*.
- Usuários de um GestãoDoc — Como base para auditoria ou inspeção do GestãoDoc existente.
- Fornecedores e desenvolvedores de sistemas — Como guia no desenvolvimento de um GestãoDoc em conformidade com os requisitos exigidos.
- Profissionais e provedores de serviços de gestão de documentos — Com vistas a orientar a execução desses serviços a partir de uma abordagem arquivística.
- Potenciais usuários de serviços externos de gestão de documentos — Guia para a especificação dos serviços a serem adquiridos.

## 1.3 Organização do MoReq-Jus

O MoReq-Jus descreve o modelo de requisitos necessários para o desenvolvimento de um GestãoDoc.

O capítulo inicial apresenta a gestão de processos e documentos em instituições judiciárias, além de questões relativas à política arquivística e os instrumentos que devem ser utilizados na gestão de processos e documentos. Os demais capítulos apresentam os aspectos da funcionalidade, o glossário e as referências normativas e bibliográficas.

Os aspectos da funcionalidade estão divididos em:

- Organização dos documentos (plano de classificação);
- Captura;
- Armazenamento;
- Preservação;
- Segurança;
- Tramitação;
- Destinação;
- Pesquisa, localização e apresentação das informações;
- Funções administrativas;
- Usabilidade;
- Interoperabilidade
- Disponibilidade e
- Desempenho.

Cada capítulo compreende um texto preliminar que apresenta o assunto e a relação dos requisitos correspondentes ao capítulo. Os requisitos são apresentados em quadros numerados com o enunciado correspondente e a classificação dos níveis de obrigatoriedade.

### Níveis dos requisitos

Os requisitos foram classificados em obrigatórios e desejáveis — de acordo com o grau de exigência — para que o GestãoDoc desempenhe suas funções.

Cada requisito numerado é classificado como:

- O (Obrigatório) — O requisito é imprescindível.
- D (Desejável) — Podem existir razões válidas em circunstâncias particulares para se ignorar um determinado item, mas a totalidade das implicações deve ser cuidadosamente examinada antes da escolha de uma proposta diferente.

Com relação aos requisitos considerados desejáveis, deve ser observado que um sistema que não inclui determinado item desejável deve estar preparado para



interoperar com outro que o inclui, mesmo com o não-atendimento pleno da funcionalidade. De forma inversa, um sistema que inclui um item desejável deve estar preparado para interoperar com outro que não o inclui.

## **1.4 Gestão de processos e documentos no Judiciário brasileiro**

O processo de informatização das instituições do Judiciário iniciou-se na década de 80.

Essas mudanças, aceleradas com a aprovação da Lei nº. 11.419, de 2006, que dispôs sobre a informatização do processo judicial e procedeu às alterações necessárias para esse fim no Código de Processo Civil, Lei nº. 5.869, de 11 de janeiro de 1973, decorreram das características do documento digital, que deixa de ser apenas entidade física e torna-se também entidade lógica e conceitual, trazendo a necessidade de adequação nas políticas de segurança e de preservação de documentos da instituição.

Os documentos digitais e as alterações na legislação processual trouxeram uma série de vantagens na produção, na transmissão, no armazenamento e no acesso aos documentos, mas, por sua vez, provocaram novos desafios. A facilidade de acesso pode acarretar intervenções não-autorizadas, adulteração ou perda dos documentos.

Também os sistemas de gerenciamento passam a ser utilizados para os documentos não-digitais e digitais.

Os documentos produzidos e recebidos no decorrer das atividades do Judiciário, independentemente do suporte em que se apresentam, registram suas políticas, funções, procedimentos e decisões.

Para conferir essa capacidade, os documentos precisam ser confiáveis, autênticos, acessíveis, compreensíveis e preserváveis, o que só é possível com a implantação de um programa de gestão de processos e documentos.

Os documentos institucionais, segundo a Teoria das Três Idades, quanto ao seu ciclo de vida, são classificados em correntes, intermediários e permanentes.

As operações técnicas cujos requisitos estão relacionados no MoReq-Jus destinam-se à gestão dos documentos em todas as fases de seu ciclo de vida, visando a eficácia administrativa com relação à recuperação da informação disponível, à tomada de decisões e ao cumprimento da missão institucional do Judiciário.

### 1.4.1 Definição do programa de gestão documental

Os órgãos e entidades integrantes do Poder Judiciário deverão implementar os seus Programas de Gestão Documental que tenham por objeto as ações previstas no art. 20 da Lei nº 8.159, de 1991.

Estas ações devem ter por objetivo produzir, tramitar, utilizar, manter e preservar documentos confiáveis, autênticos, acessíveis e compreensíveis a longo prazo.

As linhas gerais do programa de gestão, bem como os procedimentos necessários para que essas intenções sejam alcançadas devem também ser comunicadas e implementadas em todos os níveis dos órgãos e entidades.

Assim, para efetivação do programa e imprescindível a formação de um grupo de trabalho ligado aos níveis mais altos da hierarquia do órgão ou entidade, com a designação de responsáveis pelo cumprimento do programa e implementação das respectivas ações.

O programa de gestão de documentos deve ser formulado com base na análise do perfil institucional, isto é, seu contexto jurídico-administrativo, estrutura organizacional, missão, competências, funções e atividades, de forma que os documentos produzidos sejam os mais adequados, completos e necessários.

Esse Programa deverá definir um conjunto de procedimentos e operações técnicas que compreendem a gestão de documentos na instituição, que deverá observar as seguintes características:

- As unidades arquivísticas devem atuar como responsáveis pela avaliação e destinação documental, organização do acervo arquivístico da instituição e pelo acesso aos documentos sob sua guarda, entre outras atribuições.
- Prever a existência de instrumentos de classificação, destinação e temporalidade.
- Promover a constituição de comissões e grupos permanentes de gestão documental.
- Estabelecer que a guarda do documento, independentemente do suporte físico (papel ou eletrônico), deve garantir sua autoria, integridade e tempestividade.
- Determinar a observância aos critérios de preservação ambiental, preferencialmente por meio da reciclagem dos documentos a serem descartados.

### 1.4.2 Instrumentos utilizados na gestão de processos e documentos

Os instrumentos utilizados na gestão de processos e documentos sistematizam a gestão dos documentos produzidos e recebidos pelo Judiciário no exercício de suas atividades, com vistas a uniformizar o tratamento da documentação, agilizando a recuperação da informação.

Para o programa de gestão documental alguns instrumentos devem ser desenvolvidos, tais como:

- plano de classificação, tabela de temporalidade e destinação de documentos baseado nas funções e atividades do órgão ou entidade;
- tabela unificada de assuntos;

- manual de gestão de documentos institucionais (arquivísticos);
- vocabulário controlado e tesauro;
- outros instrumentos aplicáveis.

### 1.4.3 Designação de responsabilidades

A designação de responsabilidades é um dos fatores que garantem o êxito da gestão de processos e documentos. Nesse sentido, as autoridades responsáveis terão o dever de assegurar o cumprimento das normas e dos procedimentos previstos no programa de gestão.

As responsabilidades devem ser distribuídas a todos os magistrados e servidores de acordo com a função e a hierarquia de cada um, além disso, devem envolver as seguintes categorias:

- Presidentes dos Tribunais, corregedores, diretores de foro e magistrados — reais responsáveis pela viabilidade do programa e normas aplicáveis. Caberá a eles apoiar integralmente a implantação dos requisitos estabelecidos neste documento, alocando recursos humanos, materiais e financeiros e promovendo o envolvimento de todos no programa de gestão de processos e documentos.
- Comitês e grupos de gestão documental — têm a competência de acompanhar a aplicação dos procedimentos do Programa de Gestão Documental; atuar de forma ativa no processo de avaliação dos documentos e processos administrativos e judiciais definidos nos instrumentos de avaliação como passíveis de eliminação, com vistas a selecionar aqueles que, pela sua peculiaridade, devem ser preservados permanentemente; e analisar propostas de guarda definitiva de documentos realizadas por magistrados, bem como pronunciar-se acerca do seu acolhimento.
- Profissionais de arquivo — responsáveis pela implantação do programa de gestão documental e pela avaliação e controle dos trabalhos executados no âmbito de suas instituições. Além disso, o profissional de arquivo é responsável também pela disseminação das técnicas e cultura arquivísticas.
- Gerentes de unidades organizacionais ou grupos de trabalho – responsáveis por garantir que os membros de sua equipe produzam e mantenham documentos como parte de suas tarefas, de acordo com o programa de gestão de processos e documentos.
- Usuários — responsáveis, em todos os níveis, pela produção e uso dos documentos institucionais em suas atividades rotineiras, conforme estabelecido pelo programa de gestão.
- Gestores dos sistemas de informação e de tecnologia da informação – responsáveis pelo projeto, desenvolvimento e manutenção de sistemas de informação nos quais os

documentos digitais e não-digitais são gerados e usados, e pela operacionalização dos sistemas informatizados.

#### 1.4.4 Exigências a serem cumpridas pelo programa de gestão de processos e documentos

O programa de gestão de processos e documentos terá de atender a uma série de exigências, tanto em relação ao documento institucional como ao seu próprio funcionamento.

O documento institucional deve:

- Refletir corretamente o que foi comunicado, decidido ou implementado;
- Conter os metadados necessários para documentar a ação;
- Servir de suporte às atividades;
- Revelar as atividades realizadas.

O programa de gestão de processos e documentos deve:

- Contemplar o ciclo de vida dos documentos;
- Garantir a acessibilidade aos documentos;
- Manter os documentos em ambiente seguro;
- Reter os documentos somente pelo período estabelecido nos instrumentos de classificação, temporalidade e destinação da política de gestão documental do Judiciário brasileiro;
- Implementar estratégias de preservação dos documentos desde sua produção e pelo tempo que houver sido definido e
- Garantir as qualidades de um documento institucional: organicidade, unicidade, confiabilidade, integridade, autenticidade, não-repúdio, tempestividade e confidencialidade.

A cada uma das mencionadas qualidades do documento institucional, corresponde novo conjunto de exigências a serem cumpridas pelo programa de gestão, conforme especificado a seguir:

- **Organicidade** — O documento institucional caracteriza-se por sua contextualização, que reflete suas funções e atividades. Os documentos institucionais apresentam um conjunto de relações que devem ser mantidas, com o registro da seqüência das atividades realizadas por meio da aplicação dos critérios de classificação.
- **Unicidade** — O documento é único no conjunto documental ao qual pertence; podem existir cópias em um ou mais grupos de documentos, mas cada cópia é única em seu lugar, porque o conjunto de suas relações com os demais documentos do grupo é sempre único. A fim de evitar duplicação

dos documentos, permite-se a utilização de referências lógicas para a individualização dos documentos digitais. Nessa hipótese, deve ser garantida a localização única do documento.

- **Confiabilidade** — O documento é dotado de todos os elementos exigidos pela organização e pelo sistema jurídico-administrativo a que pertence, de forma a produzir conseqüências no mundo administrativo e jurídico. É criado por usuário autorizado, e todos os seus procedimentos de criação foram bem controlados. Pode-se garantir de forma indubitável a autoria do documento e que este não foi alterado. Os documentos digitais deverão ser assinados eletronicamente, conforme legislação vigente.
- **Integridade** — O documento institucional deve ter a garantia de que se encontra completo e que não sofreu nenhum tipo de corrupção ou alteração não-autorizada nem documentada. O programa de gestão documental deve definir estratégias de armazenamento e preservação e regras para a transmissão dos documentos.
- **Autenticidade** — O documento institucional autêntico é aquele que é o que diz ser, independentemente de se tratar de original ou cópia. O documento autêntico deve manter a mesma forma do momento de sua produção e ter a garantia de sua autoria. O programa de gestão documental deve implementar políticas e procedimentos que controlem a transmissão, a manutenção, a avaliação, a destinação e a preservação dos documentos, impedindo-os de sofrerem qualquer alteração, exclusão ou ocultação indevidas.
- **Não-repúdio** — O documento institucional deve ter garantida a sua autoria, evitando-se que haja qualquer dúvida quanto ao produtor do documento. O programa de gestão documental deve garantir a identificação do autor do documento, que deverá ser realizada por meio de identificação única e unívoca do autor.
- **Tempestividade** — O documento institucional deve ter garantida a hora legal do momento de sua produção, alteração e registros dos eventos de sua tramitação. O programa de gestão documental deve possuir um mecanismo de protocolo para os documentos. No caso do documento digital, deverá ser prevista a utilização de uma Autoridade de Tempo com data e hora sincronizada com o Observatório Nacional e periódica auditoria pelo mesmo, conforme legislação vigente.
- **Confidencialidade** — O documento institucional só poderá ser acessado e manipulado por pessoas ou unidades previamente autorizadas. O programa de gestão documental deve definir estratégias de armazenamento e preservação, bem como regras para a transmissão dos documentos. Os mecanismos de assinatura e preservação do documento deverão respeitar a legislação vigente, de modo a ter garantida sua identificação e permitir-lhe o acesso.

### 1.4.5 Metodologia do programa de gestão

A metodologia de planejamento e implantação de um programa de gestão de processos e documentos estabelece oito passos, não necessariamente sequenciais, podendo ser desenvolvidos em diferentes estágios, interativa, parcial ou gradualmente, de acordo com as necessidades da instituição. A metodologia prevê ainda ciclos de aplicação, e as tarefas previstas do passo “c” ao passo “h” devem ser realizadas periodicamente.

É importante destacar que este programa de gestão não se restringe a documentos digitais. É necessário prever a manutenção de documentos em outros suportes, como papel, fitas de vídeo ou de áudio etc. Há de se considerar, portanto, o caráter híbrido dos documentos a serem geridos e a necessidade de preservação da integridade e da usabilidade dos documentos digitais e não-digitais.

Os oito passos referidos são:

#### a. Levantamento preliminar

Consiste em identificar e registrar atos normativos, legislação, regimento e regulamentos.

O objetivo deste primeiro passo é gerar o conhecimento necessário sobre a missão, a estrutura organizacional e o contexto jurídico-administrativo no qual a instituição opera, de forma a identificar as exigências para produzir e manter documentos.

Esta etapa de levantamento é fundamental para a definição de quais documentos devem ser produzidos e capturados, de acordo com as normas estabelecidas no programa de gestão documental.

#### b. Análise das funções, das atividades desenvolvidas e dos documentos produzidos

Consiste em identificar, documentar e classificar cada função e atividade, bem como identificar e documentar os fluxos de trabalho e os documentos produzidos.

O objetivo é desenvolver um modelo conceitual sobre o que a instituição faz e como faz, demonstrando como os processos e documentos se relacionam com a missão e as atividades.

O levantamento da produção documental deve subsidiar a definição dos procedimentos de produção, captura controle, armazenamento, acesso e destinação dos documentos.

Com a ampliação da produção de documentos digitais, é desejável que o levantamento seja permanentemente atualizado. Essa definição é particularmente importante em ambientes digitais, em que os documentos só poderão ser capturados e mantidos se o sistema tiver sido projetado para tal.

Os produtos resultantes deste passo devem incluir:

- Esquema de classificação das funções e;
- Mapa dos fluxos de trabalho que mostre quando e quais documentos são produzidos ou recebidos como resultado das atividades desenvolvidas.

A análise das funções e atividades fornece a base para desenvolver ferramentas de gestão de documentos, que devem incluir:

- Critérios de classificação para contextualizar os documentos produzidos e recebidos.

- Instrumentos de classificação, temporalidade e destinação para estabelecer os prazos de guarda e as ações de destinação dos documentos.
- Tesouro e vocabulário controlado para identificar e indexar documentos de uma atividade específica.

### **c. Identificação das exigências a serem cumpridas para a produção de documentos**

Consiste em identificar que documentos devem ser produzidos, determinar a forma documental que melhor satisfaça cada atividade desempenhada e definir quem está autorizado a produzir cada documento. Essas exigências devem tomar por base a legislação vigente, as normas internas, a necessidade de se manter documentos em suporte digital e não-digital concomitantemente e, ainda, os riscos decorrentes da falta de registro de uma atividade em um tipo de documento.

O objetivo desse passo é assegurar que somente os documentos realmente necessários sejam produzidos, que sua produção seja obrigatória e que o seja de forma completa e correta.

Os produtos resultantes deste passo podem incluir:

- Lista das exigências a serem cumpridas para a produção e a manutenção de documentos.
- Relatório de avaliação dos riscos decorrentes da falta de registro de uma atividade em documento.
- Documento formal, regulamentando as exigências a serem cumpridas para a produção e a manutenção de documentos, especificando que documentos devem ser produzidos, a forma documental que devem apresentar e a relação dos níveis de permissão de acesso.

### **d. Avaliação dos sistemas existentes**

Consiste em identificar e avaliar o sistema de gestão de processos e documentos, bem como outros sistemas de informação e comunicação existentes.

O objetivo deste passo é identificar as lacunas entre as exigências para a produção e manutenção de processos e documentos e o desempenho do sistema de gestão de processos e documentos, bem como dos sistemas de informação e comunicação existentes. Isso fornecerá a base para o desenvolvimento de novos sistemas ou alterações nos sistemas vigentes de forma a atender às exigências, identificadas e acordadas nos passos anteriores.

Os produtos resultantes deste passo podem ser:

- Inventário do sistema de gestão de processos e documentos, bem como dos demais sistemas de informação e comunicação existentes;
- Relatório sobre o sistema de gestão de processos e documentos e sistemas de informação existentes, avaliando até que ponto atendem às exigências a serem cumpridas para a produção e manutenção de documentos;
- Relatório sobre a avaliação da integração entre os documentos digitais e os não-digitais constantes de um dossiê híbrido.

#### **e. Identificação das estratégias para satisfazer as exigências a serem cumpridas para a produção de documentos**

Consiste em determinar as estratégias (padrões, procedimentos, práticas e ferramentas) que levem ao cumprimento das exigências para a produção de documentos. O objetivo deste passo é avaliar o potencial de cada estratégia para alcançar o resultado desejado e o risco, em caso de falha.

A escolha das estratégias deve considerar:

- a natureza da instituição, incluindo sua missão e história;
- os tipos de atividades desenvolvidas;
- a forma como as atividades são conduzidas;
- o ambiente tecnológico existente;
- as tendências tecnológicas;
- a cultura institucional e
- a inclusão das funcionalidades do sistema anterior.

Os produtos resultantes deste passo podem incluir:

- lista das estratégias selecionadas para satisfazer as exigências para a produção de documentos.
- documento a ser encaminhado à administração, recomendando a elaboração de um projeto de gestão de documentos e relacionando as estratégias a serem adotadas, com as devidas justificativas.

#### **f. Projeto e implementação do sistema de gestão de processos e documentos**

Consiste em projetar um sistema de gestão que incorpore as estratégias selecionadas no passo anterior, que atenda às exigências identificadas e documentadas no passo “c” e que corrija quaisquer deficiências identificadas no passo “d”, redesenhando os procedimentos e os sistemas de informação e comunicação existentes, implementando-os e integrando-os ao sistema de gestão de processos e documentos.

O projeto de um sistema de gestão de processos e documentos visa:

- Projetar mudanças ou adaptações para sistemas, nos processos e práticas correntes;
- Determinar como incorporar essas mudanças ou adaptações para melhorar a gestão dos processos e documentos na instituição;
- Adaptar ou adotar soluções tecnológicas, considerando, o quanto possível, um plano estratégico de evolução para minimizar os efeitos da obsolescência tecnológica;

Para alcançar esses objetivos, o projeto e a implementação de um sistema de gestão de processos e documentos devem ter como base uma metodologia de desenvolvimento de sistemas que inclua:

- Organização dos processos envolvidos.
- Especificações detalhadas dos componentes tecnológicos, como *software* e *hardware*, considerando que o sistema deve ser modular, evolutivo e expansível. Uma metodologia de especificação de requisitos do *software* deve ser adotada para representar as diferentes



visões do sistema, como, por exemplo, casos de uso.

- Metodologia de gestão de projetos, envolvendo planejamento:
- das atividades (escopo), incluindo as responsabilidades (recursos humanos) e o cronograma (tempo);
- dos custos;
- das aquisições;
- dos riscos;
- da integração;
- da qualidade;
- da comunicação;
- Plano de segurança da informação (física e lógica) e de ;
- Metodologia e procedimentos de auditoria;
- *Design do software*, com diagramas representando a arquitetura e os componentes do sistema, a integração e a interoperabilidade entre os sistemas;
- Implementação dos componentes de *software*, pela construção e aquisição;
- Documentação técnica do sistema voltada a usuários;
- Testes do sistema;
- Plano de implantação do sistema, inclusive com previsão de treinamento de pessoal e
- Detalhamento das revisões periódicas do projeto, em conformidade com o plano estratégico de evolução e com as mudanças na tecnologia e no mercado.

#### **g. Implantação do sistema de gestão de processos e documentos**

Consiste em colocar em produção o objeto do projeto e da implementação por meio de:

- procedimentos de carga de dados, conversão de dados e migração de sistemas;
- homologação do sistema, mediante sua validação por parte dos usuários e da área de tecnologia de informação (TI), para autorizar a operação do sistema no ambiente de produção;
- projeto piloto — entrada em produção do sistema em uma área de abrangência menor;
- dimensionamento de ambiente computacional (*hardware*, *software* e comunicação de dados) para dar suporte ao sistema de gestão de processos e documentos;
- instalação dos componentes;
- parametrização do sistema para adaptá-lo a necessidades específicas;
- treinamento de pessoal,
- operação assistida;
- integração do sistema com os procedimentos e os demais sistemas de informação e comunicação existentes e
- suporte e manutenção do sistema.

A implantação de um sistema de gestão de processos e documentos é um

empreendimento complexo. Ela deve ser planejada de modo a minimizar a necessidade de interrupções das atividades na instituição. O processo de implantação deverá contar com documentação prévia, detalhando os passos previstos, assim como gerar relatórios das ações realizadas e de problemas encontrados.

Os produtos resultantes deste passo podem incluir:

- Regulamentação das políticas, diretrizes e procedimentos por meio de normas e manuais.
- Material de treinamento.
- Documentação dos processos de conversão e migração dos sistemas.
- Relatórios sobre avaliação de desempenho do sistema.

#### **h. Monitoramento e ajustes**

Consiste em recolher, de forma sistemática, informação sobre o desempenho do sistema de gestão de processos e documentos.

O desempenho é medido ao avaliar se os documentos são produzidos e organizados de acordo com as necessidades da instituição e se estão relacionados apropriadamente aos processos dos quais fazem parte.

O objetivo deste passo é medir o desempenho do sistema, detectar possíveis deficiências e fazer os ajustes necessários.

Este passo envolve:

- Planejamento e aplicação de testes de avaliação de desempenho;
- Entrevistas com a administração, equipe e outros parceiros;
- Aplicação de questionários para medir o desempenho do sistema e
- Observação, análise, avaliação da correção e auditoria das informações e dos procedimentos implementados.

A análise de informações quantitativas providas pelo monitoramento tem por objetivo a avaliação concreta dos benefícios da automação trazida pelo sistema. Outro objetivo importante do monitoramento constante é a minimização de riscos nas atividades do programa de gestão.

Constatações de insuficiência de desempenho do sistema, em face da demanda das informações gerenciadas, poderão indicar a necessidade de incremento no *hardware* (*upgrades*), reconfigurações no ambiente do sistema e evolução do *software*, na forma de versões otimizadas.

Os produtos resultantes deste passo podem incluir:

- Desenvolvimento e aplicação de uma metodologia para avaliar objetivamente o sistema de gestão de processos e documentos;
- Documentação do desempenho desse sistema e
- Relatório para a administração com conclusões e recomendações.

#### 1.4.6 Suspensão ou extinção do GestãoDoc

Quando um GestãoDoc é suspenso ou extinto, ele deve se tornar acessível para consulta, e novos documentos não devem ser incluídos. Quanto aos documentos já inseridos, poderão ser removidos de acordo com as diretrizes de destinação ou transferidos para outros sistemas.

O processo de suspensão ou extinção do GestãoDoc deve ser documentado, incluindo planos de conversão ou mapeamento dos dados, pois essas informações detalhadas serão necessárias à verificação de autenticidade, integridade e manutenção da acessibilidade dos documentos contidos no sistema suspenso ou extinto.

## 2 Organização dos documentos institucionais: plano de classificação e manutenção dos documentos

Os documentos institucionais podem ser agregados em processos/dossiês, de forma estruturada. Essa estrutura reflete as funções e atividades da organização, representadas no plano de classificação.

O Judiciário comumente organiza seus documentos institucionais em documentos e processos: judiciais e administrativos.

### 2.1 Configuração e administração do plano de classificação no GestãoDoc

Os requisitos desta seção referem-se às funcionalidades que deverão ser desenvolvidas no sistema para apoiar a configuração dos instrumentos utilizados na classificação dos documentos e processos. Esses requisitos especificam como desenhar um plano de classificação dentro de GestãoDoc.

REF.	REQUISITO	OBRIG.
RPC2.1.1	Incluir os instrumentos de classificação de documentos e processos judiciais e administrativos de acordo com as normas vigentes compatíveis com o programa de gestão documental.	O

<b>REF.</b>	<b>REQUISITO</b>	<b>OBRIG.</b>
RPC2.1.2	Garantir a criação de classes, subclasses, grupos e subgrupos nos níveis do plano de classificação e de outros instrumentos de classificação de documentos e processos de acordo com o método de codificação adotado.	<input type="radio"/>
RPC2.1.3	Permitir a usuários autorizados o acréscimo de novas classes de acordo com as alterações dos instrumentos de classificação de documentos e processos.	<input type="radio"/>
RPC2.1.4	Registrar as datas de abertura de uma nova classe, reclassificação, movimentação e modificação da classe no respectivo metadado.	<input type="radio"/>
RPC2.1.5	Registrar a mudança de nome de uma classe já existente no respectivo metadado.	<input type="radio"/>
RPC2.1.6	Permitir o deslocamento de uma classe inteira, incluindo as subclasses, grupos, subgrupos e os documentos ali classificados, para outra localização no plano de classificação. Nesse caso, é necessário fazer o registro do deslocamento nos metadados desses instrumentos.	<input type="radio"/>
RPC2.1.7	Permitir que o gestor do sistema torne inativa e inacessível aos demais usuários uma classe em que não serão mais classificados documentos.	<input type="radio"/>
RPC2.1.8	Impedir a eliminação de uma classe ativa ou inativa.	<input type="radio"/>
RPC2.1.9	Permitir a associação de metadados às classes e restringir a inclusão e alteração desses mesmos metadados somente a usuários autorizados.	<input type="radio"/>
RPC2.1.10	Disponibilizar pelo menos dois mecanismos de atribuição de identificadores às classes do plano de classificação, prevendo a possibilidade de se utilizar ambos, separadamente ou em conjunto, na mesma aplicação: <ul style="list-style-type: none"><li>▪ Atribuição de um código numérico ou alfanumérico.</li><li>▪ Atribuição de um termo que identifique cada classe.</li></ul>	<input type="radio"/>

REF.	REQUISITO	OBRIG.
RPC2.1.11	<p>Prever atributos associados aos instrumentos de classificação e indexação por assuntos, de modo a não permitir classificações genéricas, nos casos em que a classificação por assuntos não atender ao caso específico.</p> <p>Com exceção do usuário autorizado, em algumas classes não é permitido incluir documentos. Nesses casos os documentos devem ser classificados apenas nos níveis subordinados.</p> <p>Ex.1 – Na Tabela Unificada de Assuntos do Judiciário brasileiro os processos devem ser classificados no último nível disponível para cada assunto e quando se tratar de um novo assunto, os usuários autorizados podem classificar o processo, apenas provisoriamente, no último nível pertinente, até que se crie o novo assunto solicitado.</p>	O
RPC2.1.12	<p>Utilizar o termo completo para identificar uma classe.</p> <p>Entende-se por termo completo toda a hierarquia referente àquela classe.</p> <p>Ex: Tabela Unificada de Assuntos do Judiciário</p> <p>Termo de Adesão da LC 110/2001 – Atualização de Conta FGTS/Fundo de Garantia por Tempo de Serviço – Entidades Administrativas /Administração Pública – Direito Administrativo e outras matérias do Direito Público.</p>	O
RPC2.1.13	<p>Assegurar que os termos completos, que identificam cada classe, sejam únicos em um instrumento de classificação de processos ou documentos.</p>	O
RPC2.1.14	<p>Importar e exportar total ou parcialmente um instrumento de classificação de processos ou documentos.</p>	D
RPC2.1.15	<p>Prover funcionalidades com vistas à elaboração de relatórios para apoiar a gestão de um instrumento de classificação de processos ou documentos, incluindo a capacidade de gerar relatório:</p> <ul style="list-style-type: none"><li>▪ Completo de todo o instrumento de classificação de processos ou documentos.</li><li>▪ Parcial do instrumento de classificação de processos ou documentos a partir de um ponto determinado na hierarquia.</li><li>▪ Dos documentos ou processos/dossiês classificados em uma ou mais classes do plano de classificação.</li><li>▪ De documentos classificados por unidade administrativa.</li></ul>	O

## 2.2 Classificação e metadados dos processos/dossiês

Os requisitos desta seção referem-se à formação e classificação de processos/dossiês e à associação de metadados.

REF.	REQUISITO	OBRIG.
RPC2.2.1	Permitir a classificação dos processos/dossiês somente nas classes autorizadas. Ver RPC2.1.11 e RPC2.3.7	O
RPC2.2.2	Permitir a classificação de um número ilimitado de processos/dossiês dentro de uma classe.	O
RPC2.2.3	Utilizar o termo completo da classe para identificar um processo/dossiê, tal como especificado em RPC2.1.12.	O
RPC2.2.4	Permitir a associação de metadados aos processos/dossiês e restringir a inclusão e alteração desses mesmos metadados somente a usuários autorizados.	O
RPC2.2.5	Associar os metadados dos processos/dossiês conforme estabelecido nos elementos de metadados.	O
RPC2.2.6	Permitir que um novo processo/dossiê herde, da classe na qual foi classificado, determinados metadados predefinidos. Exemplos desta herança são: temporalidade prevista e restrição de acesso.	O
RPC2.2.7	Relacionar os metadados herdados de forma que uma alteração no metadado de uma classe seja automaticamente incorporada ao processo/dossiê que herdou esse metadado.	D
RPC2.2.8	Permitir a alteração conjunta de um determinado metadado em um grupo de processos/dossiês previamente selecionado.	O

## 2.3 Gerenciamento dos processos/dossiês

Os requisitos desta seção referem-se ao gerenciamento dos documentos institucionais no que diz respeito a controles de abertura e encerramento de processos/dossiês e seus respectivos volumes e inclusão de novos documentos nesses processos/dossiês e respectivos volumes, bem como procedimentos de reclassificação.

REF.	REQUISITO	OBRIG.
------	-----------	--------

REF.	REQUISITO	OBRIG.
RPC2.3.1	Registrar automaticamente a data de abertura e de encerramento ou baixa do processo/dossiê.  Essas datas são parâmetros para aplicação dos prazos de guarda e destinação do processo/dossiê.	O
RPC2.3.2	Permitir que um processo/dossiê seja encerrado, reaberto ou baixado mediante procedimentos regulamentares.	O
RPC2.3.3	Permitir que um processo/dossiê e seus respectivos volumes e documentos sejam reclassificados por um usuário autorizado e que todos os documentos já inseridos permaneçam nos processos/dossiês e volumes que estão sendo reclassificados, de modo a conservar a relação entre os documentos, volumes e processos/dossiês.	O
RPC2.3.4	Manter o registro de suas posições anteriores à reclassificação, quando um processo/dossiê é reclassificado, de forma a obter-se um histórico.	O
RPC2.3.5	Permitir que o usuário autorizado introduza as razões para a reclassificação, quando um processo/dossiê ou documento é reclassificado.	O
RPC2.3.6	Permitir a geração de referências cruzadas para processos/dossiês afins.	O
RPC2.3.7	Registrar múltiplas entradas para um documento digital em mais de um processo/dossiê.	O
RPC2.3.8	Impedir a eliminação de um processo/dossiê digital ou de qualquer parte de seu conteúdo em qualquer momento, exceto quando se tratar de eliminação definitiva, consoante os critérios de classificação e guarda.  A eliminação será devidamente registrada em trilha de auditoria.	O
RPC2.3.9	Impedir o acréscimo de novos documentos a processos/dossiês já encerrados.  Para receber novos documentos, os processos/dossiês encerrados deverão ser reabertos.	O
RPC2.3.10	Permitir a consulta aos processos/dossiês já encerrados.	O

REF.	REQUISITO	OBRIG.
RPC2.3.11	Garantir a integridade da relação hierárquica entre classe, processo/dossiê, volume e documento em todos os momentos, independentemente de atividades de manutenção, ações do usuário ou falha de componentes do sistema.  Em hipótese alguma poderá ocorrer uma situação em que qualquer ação do usuário ou falha do sistema dê origem a uma inconsistência na base de dados do GestãoDoc.	O
RPC2.3.12	Oferecer ferramentas para a realização de operações em lote, tais como: abertura e encerramento de processos/dossiês e seus respectivos volumes, reclassificação, citação/intimação, sentença/decisão/despacho, etc.	O

## 2.4 Processos

A formação e manutenção de processos no Judiciário obedecem a regras específicas que os diferenciam dos dossiês.

O dossiê é entendido como um conjunto de documentos relacionados entre si, tratados como uma unidade, e agregados por se reportarem a um mesmo assunto (ex.: dossiê de evento de capacitação). O processo diferencia-se do dossiê, basicamente, por ser constituído de documentos oficialmente reunidos no decurso de uma ação administrativa ou judicial.

O detalhamento dessas regras está previsto em legislação, provimentos e em normas específicas, que deverão ser respeitadas pelos órgãos, de acordo com seu âmbito de atuação.

REF.	REQUISITO	OBRIG.
RPC2.4.1	Prever a formação/autuação de processos conforme estabelecido nas leis e regulamentações vigentes.	O
RPC2.4.2	Prever que as peças integrantes do processo recebam paginação contextualizada de acordo com cada processo relacionado.  Uma mesma contestação juntada a vários processos tem numeração diferente de acordo com a numeração de cada processo.	O



<b>REF.</b>	<b>REQUISITO</b>	<b>OBRIG.</b>
RPC2.4.3	<p>Prever procedimentos para reunião de processos por apensação.</p> <p>Nos processos judiciais, a apensação ocorre por determinação legal ou judicial e nos administrativos, por determinação da autoridade competente.</p> <p>Esse procedimento deverá ser registrado nos metadados do processo. Quando se tratar de processo judicial, deve-se lançar o evento correspondente da Tabela Unificada de Movimentação Processual do Judiciário Brasileiro (apensamento) e seu complemento obrigatório (número do processo).</p>	O
RPC2.4.4	<p>Prever procedimentos para desapensação.</p> <p>Nos processos judiciais, via de regra, a desapensação ocorre por decisão judicial e nos administrativos, por determinação da autoridade competente.</p> <p>Esse procedimento deverá ser registrado nos metadados do processo. Quando se tratar de processo judicial, deve-se lançar o evento correspondente da Tabela Unificada de Movimentação Processual do Judiciário Brasileiro (desapensamento) e seu complemento obrigatório (número do processo).</p>	O
RPC2.4.5	<p>Prever procedimentos para desentranhamento de peças dos processos judiciais em atenção à decisão judicial ou segundo a legislação específica, e dos administrativos, por determinação da autoridade competente.</p> <p>Esse procedimento deverá ser registrado nos metadados do processo. Quando se tratar de processo judicial, deve-se lançar o evento correspondente da Tabela Unificada de Movimentação Processual do Judiciário Brasileiro (cancelamento de juntada — desentranhamento) e seu complemento obrigatório (especificar documento).</p>	O
RPC2.4.6	<p>Prever procedimentos para desmembramento de um processo judicial ou administrativo em dois ou mais processos.</p> <p>Nos processos judiciais, sempre ocorre por decisão judicial, como no exemplo do art. 46, parágrafo único, do CPC ou do art. 80, CPP.</p> <p>Esse procedimento deverá ser registrado nos metadados do processo.</p>	O

REF.	REQUISITO	OBRIG.
RPC2.4.7	Prever procedimentos para a anexação de documentos organizados em volumes próprios a um determinado processo. Esse procedimento deverá ser registrado nos metadados do processo.	O

## 2.5 Volumes: abertura, encerramento e metadados

Em alguns casos os processos/dossiês são compartimentados em volumes ou partes, de acordo com convenções predeterminadas. Essa divisão não está baseada no conteúdo intelectual dos processos/dossiês, mas em outros critérios, como a dimensão, o número de documentos, períodos de tempo etc. Essa prática tem como objetivo facilitar o gerenciamento físico dos processos/dossiês.

Os requisitos desta seção referem-se à utilização de volumes para subdividir processos/dossiês em processos físicos.

REF.	REQUISITO	OBRIG.
RPC2.5.1	Gerenciar volumes para subdividir processos/dossiês, distinguindo entre processos/dossiês e volumes.	O
RPC2.5.2	Permitir a associação de metadados aos volumes e restringir a inclusão e a alteração desses mesmos metadados somente a usuários autorizados.	O
RPC2.5.3	Permitir que um volume herde automaticamente do processo/dossiê ao qual pertence determinados metadados predefinidos. Por exemplo: volume juntado em processo sigiloso também é sigiloso.	O
RPC2.5.4	Permitir a abertura de volumes a qualquer processo/dossiê que não esteja encerrado.	O
RPC2.5.5	Assegurar que, ao abrir um novo volume, o volume precedente seja automaticamente encerrado, registrando a data de encerramento. Apenas o volume produzido mais recentemente pode estar aberto; todos os outros volumes existentes nesse processo/dossiê têm de estar fechados.	O
RPC2.5.6	Impedir a reabertura de um volume já encerrado para acréscimo de documentos.	O

REF.	REQUISITO	OBRIG.
RPC2.5.7	Assegurar que um volume somente conterá documentos. Não é permitido que contenha outro volume ou um outro processo/dossiê.	<input type="radio"/>
RPC2.5.8	Permitir que um volume seja encerrado por meio de procedimentos regulamentares.	<input type="radio"/>

## 2.6 Manutenção de documentos institucionais não-digitais e híbridos

O Judiciário possui documentos e processos digitais e não-digitais. Esses últimos podem estar registrados em papel ou outros suportes, tais como fitas de vídeo, de áudio etc. Um GestãoDoc deve registrar os documentos ou processos/dossiês não-digitais e digitais utilizando o mesmo plano de classificação e deve ainda possibilitar a gestão de documentos ou processos/dossiês híbridos, formados por uma parte digital e uma parte não-digital.

REF.	REQUISITO	OBRIG.
RPC2.6.1	Capturar documentos ou processos/dossiês não-digitais e gerenciá-los como os digitais. Para conceito de captura veja capítulo 3.	<input type="radio"/>
RPC2.6.2	Gerenciar as partes dos documentos ou processos/dossiês híbridos, associando-as ao mesmo número identificador, atribuído pelo sistema, e título, além de indicar que se trata de um documento institucional híbrido.	<input type="radio"/>
RPC2.6.3	Permitir que um conjunto específico de metadados seja configurado para os documentos ou processos/dossiês não-digitais e incluir informações sobre o local onde se encontram.	<input type="radio"/>
RPC2.6.4	Possuir mecanismos para acompanhar a movimentação do documento, processo/dossiê não-digital, de forma que se evidencie ao usuário a localização atual.	<input type="radio"/>
RPC2.6.5	Oferecer ao usuário funcionalidades para solicitar vista, carga ou desarquivamento de um documento e/ou processo não-digital.	<input type="radio"/>

REF.	REQUISITO	OBRIG.
RPC2.6.6	Incluir mecanismos de impressão e reconhecimento de códigos de barra para automatizar a introdução de dados e acompanhar as movimentações de documentos ou processos/dossiês não-digitais.	O
RPC2.6.7	Assegurar que a recuperação de um documento ou processo/dossiê híbrido permita igualmente a recuperação dos metadados tanto da parte digital como da parte não-digital.	O
RPC2.6.8	Sempre que os documentos ou processos/dossiês híbridos estiverem classificados quanto ao grau de sigilo, garantir que o grau de sigilo seja estendido ao todo ou à parte, independentemente do suporte.	O
RPC2.6.9	Registrar na trilha de auditoria todas as alterações efetuadas nos metadados dos documentos ou processos/dossiês não-digitais ou híbridos.	O

### 3 Captura

A captura é a incorporação de um documento/processo ao GestãoDoc, quando passará a seguir as rotinas de tramitação. Uma vez capturado, o documento será incluído no fluxo de trabalho.

Tradicionalmente, nos sistemas de gestão de processos e documentos em papel, a captura é feita no momento em que o documento é registrado, classificado e identificado. Para o processo judicial é o momento da autuação.

Em um GestãoDoc, o documento tanto pode ser produzido diretamente dentro do sistema e então capturado automaticamente no momento do registro, como pode ser produzido fora do sistema, capturado e registrado posteriormente.

A política de gestão de processos e documentos do Judiciário brasileiro é única para documentos não-digitais, digitais e híbridos, assim, os GestãoDocs terão de capturar todos os documentos pertinentes independentemente do suporte.

Além do código de classificação, descritores, número de protocolo e número de registro, a captura pode prever a introdução de outros metadados, a saber: data e hora da criação, da transmissão e do recebimento do documento; nome do autor, do originador, do digitador e do destinatário, entre outros. Esses metadados podem ser registrados em vários níveis de detalhe, dependendo das necessidades geradas pelos procedimentos normativos e do seu contexto jurídico-administrativo.

Os metadados são essenciais para identificar o documento institucional de modo inequívoco e mostrar sua relação com os outros documentos.

A captura tem como pré-requisito definir:

- Que documentos (produzidos e recebidos) serão capturados pelo sistema de gestão de processos e documentos.
- Quem deve ter acesso a esses documentos e em quais níveis.
- A destinação final do documento: guarda permanente ou passível de eliminação.

A captura consiste nas ações de:

- Protocolo
- Autuação
- Classificação
- Indexação
- Atribuição de restrição de acesso
- Arquivamento

### **Protocolo**

As atividades de protocolo são constituídas pelo conjunto de operações que visam ao controle de entrada dos documentos produzidos e recebidos que tramitam nas instituições do Judiciário brasileiro, assegurando sua localização, recuperação e acesso.

Após o recebimento dos documentos, o serviço de protocolo identifica se o documento é processual ou administrativo/forense, atribuindo-lhe número e data de entrada, anotando o código de classificação e, de acordo com essa identificação, seleciona o sistema onde tramitará ou será autuado.

### **Autuação**

- Registro do processo judicial

Consiste em formalizar a captura do documento judicial dentro do sistema de gestão de processos e documentos, por meio da atribuição de um número identificador e de uma descrição informativa.

O registro tem por objetivo demonstrar que o documento foi produzido ou recebido e capturado pelo sistema de gestão de processos e documentos, bem como facilitar sua recuperação.

Os documentos judiciais são de dois tipos:

- Vinculados diretamente ao processo (petição inicial, certidão de trânsito em julgado, recurso etc.).
- Relacionados a atividades administrativas forenses (diário eletrônico, pauta de julgamentos, estatísticas etc.).

Os documentos vinculados a processos (judiciais) serão numerados pelo sistema de Numeração Única de Processos, conforme requisitos estabelecidos pela Resolução CNJ n. 65, de 16 de dezembro de 2008, que estabeleceu a uniformização do número dos processos nos órgãos do Poder Judiciário.

Além de numerados, os documentos serão classificados de acordo com as Tabelas Processuais Unificadas do Judiciário brasileiro.

- Registro de documentos e processos administrativos

As atividades dos protocolos administrativos, independentemente de serem centralizados ou descentralizados, visam ao registro do documento institucional por meio da captura dos dados informacionais dos documentos recebidos, tais como: o número original, o assunto, a data da produção, a data do registro, o conteúdo etc., com a finalidade de informar, de forma rápida e precisa, a sua situação e localização.

Os documentos produzidos no âmbito de cada uma das instituições do Judiciário (memorandos, pareceres, requisições etc.) serão numerados, classificados e encaminhados com base no Plano de Classificação e Temporalidade da Documentação Administrativa interno da instituição até que o assunto seja regulamentado pelo CNJ.. Os documentos pertinentes às atividades administrativas forenses devem receber um número atribuído pelo sistema e classificação de acordo com a classe do instrumento mencionado.

### **Classificação**

É o ato ou efeito de analisar e identificar o conteúdo dos documentos e processos judiciais e de selecionar a classe (tipologia documental) à qual pertencem para fins de arranjo e de recuperação da informação. Essa classificação é feita a partir de planos ou esquemas de classificação aprovados pelo Conselho Nacional de Justiça, como as tabelas unificadas de classes e de assuntos do Judiciário brasileiro.

A classificação deve refletir a atividade que gerou o documento e determinar o uso da informação nele contida. Ela também define a organização física dos documentos não-digitais, constituindo-se em referencial básico para sua recuperação.

Objetivos da classificação:

- Estabelecer a relação orgânica dos documentos institucionais.
- Assegurar que os documentos sejam identificados de forma consistente ao longo do tempo.
- Auxiliar a recuperação de todos os documentos institucionais relacionados a uma determinada função ou atividade.
- Possibilitar a avaliação de um grupo de documentos de forma que os documentos associados sejam transferidos, recolhidos ou eliminados em conjunto.

A classificação deve se basear no plano de classificação e envolve os seguintes passos:

- Identificar a ação que o documento registra.
- Localizar a ação ou atividade no plano de classificação.
- Comparar a atividade com a estrutura organizacional para verificar se é apropriada à unidade que gerou o documento.
- Aplicar a classificação ao documento.

No Judiciário brasileiro, os documentos e processos administrativos serão

classificados para fins de armazenamento e atribuição de prazos de guarda e de destinação com base em planos de classificação e tabela de temporalidade aprovados pelos seus tribunais ou no instrumento com essa finalidade a ser elaborado pelo Comitê Gestor do PRONAME e aprovado pelo CNJ.

Os processos judiciais são classificados pela Tabela Unificada de Classes Processuais do Judiciário brasileiro, que lhes assegura a reunião pela tipologia documental.

### **Indexação**

A indexação de assuntos envolve duas etapas principais:

- Análise conceitual — atividade de definição dos assuntos tratados no documento.
- Tradução — atividade de conversão dos conceitos identificados na análise para uma linguagem de indexação (vocabulário controlado e/ou lista de descritores, tesouro e o próprio plano de classificação).
- O principal objetivo da indexação é assegurar a recuperação de qualquer documento em um sistema de informações.
- Os processos judiciais são indexados apenas com base na Tabela Unificada de Assuntos Processuais do Judiciário brasileiro.

### **Atribuição de restrição de acesso**

Os documentos também devem ser analisados com relação às precauções de segurança — se são considerados ostensivos ou sigilosos. Para os documentos sigilosos, a legislação estabelece diferentes graus a serem atribuídos a cada documento.

Os documentos que dizem respeito à segurança da sociedade e do Estado, bem como aqueles necessários à utilidade do processo e ao resguardo da inviolabilidade da intimidade, da vida privada, da honra e da imagem das pessoas estarão sujeitos às restrições de acesso, conforme legislação em vigor.

A atribuição de restrições pode ser feita em qualquer fase da tramitação, com base no esquema de classificação de segurança e sigilo elaborado pelas instituições do Judiciário brasileiro e envolve os seguintes passos:

- Identificar a ação ou atividade que o documento registra.
- Identificar a unidade administrativa à qual o documento pertence.
- Verificar a precaução de segurança e o grau de sigilo.
- Atribuir o grau de sigilo e as restrições de acesso ao documento.
- Registrar o grau de sigilo e as restrições de acesso no sistema de gestão de processos e documentos.

Os requisitos de segurança são apresentados no Capítulo 6.

### **Arquivamento**

O arquivamento de documentos e processos em fase corrente é a guarda dos documentos não-digitais ou mídias digitais no local estabelecido, de acordo com a sua

classificação.

Os métodos de arquivamento devem ser associados ao Plano de Classificação, de acordo com a especificidade de cada tipo documental, visando facilitar o arquivamento, a busca e a recuperação do documento.

A escolha de um método ideal de arquivamento deve ser determinada pela natureza dos documentos e pela estrutura da organização. Esses métodos podem ser:

- Direto (alfabético, geográfico) — Busca o documento no local arquivado, sem intervenção de instrumentos de pesquisas.
- Indireto (por assuntos, numérico) — Necessita de consultas a guias ou instrumentos de pesquisa para a localização do documento.

A operação de arquivamento dos documentos digitais se diferencia do arquivamento dos documentos não-digitais porque nesses últimos a operação é ao mesmo tempo lógica e física, como, por exemplo, arquivar um relatório na pasta.

### 3.1 Captura: procedimentos gerais

REF.	REQUISITO	OBRIG.
RCA3.1.1	<p>A captura deve garantir a execução das funções relacionadas a seguir, de acordo com o sistema de classificação do Judiciário brasileiro:</p> <ul style="list-style-type: none"> <li>▪ Registrar e gerenciar todos os documentos não-digitais.</li> <li>▪ Registrar e gerenciar todos os documentos digitais, independentemente do contexto tecnológico.</li> <li>▪ Classificar todos os documentos de acordo com o plano de classificação ou critérios de guarda.</li> <li>▪ Controlar e validar a introdução de metadados.</li> </ul>	O
RCA3.1.2	<p>Capturar documentos digitais das seguintes formas:</p> <ul style="list-style-type: none"> <li>▪ Documento individual produzido em arquivo digital fora do GestãoDoc.</li> <li>▪ Documento individual produzido em <i>workflow</i> integrado ao GestãoDoc.</li> <li>▪ Documentos em lote.</li> </ul>	O
RCA3.1.3	<p>Automatizar a produção de documentos por meio da exibição de formulários e modelos predefinidos pelo programa de gestão de processos e documentos.</p>	D
RCA3.1.4	<p>Automatizar a produção de petições eletrônicas por meio da exibição de formulários e modelos predefinidos.</p>	O



REF.	REQUISITO	OBRIG.
RCA3.1.5	Aceitar o conteúdo do documento, bem como as informações que definem sua apresentação.	<input type="radio"/>
RCA3.1.6	Os documentos associados a vários objetos digitais devem ser desassociados e capturados individualmente.	<input type="radio"/>
RCA3.1.7	<p>Permitir a inserção de todos os metadados, obrigatórios e optativos, definidos em sua configuração e garantir que se mantenham associados ao documento.</p> <p>São exemplos de metadados:</p> <ul style="list-style-type: none"> <li>▪ Nome do arquivo digital.</li> <li>▪ Número identificador atribuído pelo sistema.</li> <li>▪ Data e hora de produção e captura.</li> <li>▪ Data e hora de transmissão e recebimento.</li> <li>▪ Título ou descrição abreviada.</li> <li>▪ Formato (gênero / espécie / tipo).</li> <li>▪ Usuário cadastrador.</li> <li>▪ Unidade responsável pela execução da ação.</li> <li>▪ Indicação de anotação.</li> <li>▪ Restrição de acesso.</li> <li>▪ Registro das migrações e data em que ocorreram.</li> <li>▪ Números das páginas inicial e final do documento.</li> <li>▪ Tamanho do documento.</li> <li>▪ Checksum.</li> <li>▪ <i>Software</i> (nome e versão) sob o qual o documento foi produzido ou no qual foi capturado.</li> <li>▪ Máscaras de formatação (<i>template</i>) necessárias para interpretar a estrutura do documento.</li> <li>▪ Classificação de acordo com os instrumentos de classificação, temporalidade e destinação da política de gestão documental do Judiciário brasileiro.</li> </ul>	<input type="radio"/>
RCA3.1.8	Permitir a inserção dos metadados obrigatórios, previstos em legislação específica do Judiciário brasileiro no momento da captura de documentos e processos/dossiês.	<input type="radio"/>
RCA3.1.9	Atribuir um número identificador a cada processo/dossiê e documento capturado, que serve para identificá-lo desde o momento da captura até sua destinação final dentro do GestãoDoc a fim de manter a integridade.	<input type="radio"/>

REF.	REQUISITO	OBRIG.
RCA3.1.10	<p>O formato do número identificador atribuído pelo sistema deve ser definido no momento da configuração do GestãoDoc.</p> <p>O identificador pode ser numérico ou alfanumérico, ou pode incluir os identificadores encadeados das entidades superiores no ramo apropriado da hierarquia.</p>	O
RCA3.1.11	<p>O identificador atribuído pelo sistema deve:</p> <ul style="list-style-type: none"> <li>▪ Ser único e gerado automaticamente, vedada sua introdução manual e alteração posterior.</li> </ul> <p>ou</p> <ul style="list-style-type: none"> <li>▪ Ser atribuído pelo usuário e validado pelo sistema antes de ser aceito.</li> </ul>	O
RCA3.1.12	Utilizar o sistema de classificação e indexação de assuntos do Judiciário brasileiro para apoiar a atribuição do metadado assunto/descritor.	O
RCA3.1.13	Utilizar um tesauro ou vocabulário controlado para apoiar a atribuição do metadado assunto/descritor.	D
RCA3.1.14	Garantir que os metadados associados a um documento sejam alterados somente por usuários autorizados.	O
RCA3.1.15	<p>Inserir automaticamente os metadados previstos no sistema para o maior número possível de documentos.</p> <p>Por exemplo, para diminuir as tarefas do usuário do sistema e garantir maior exatidão e eficiência na inserção dos metadados, no caso de documentos com forma padronizada (formulários, modelos de requerimentos, memorandos etc.) alguns metadados podem ser inseridos automaticamente, tais como: número identificador, título, prazo de guarda.</p>	D
RCA3.1.16	<p>Garantir a visualização do registro de entrada do documento dentro do sistema com todos os metadados que possam ser inseridos automaticamente e os demais a serem atribuídos pelo usuário.</p> <p>Por exemplo, o sistema pode atribuir automaticamente o número identificador, a data de captura, o título, o originador e requerer que o usuário preencha os demais metadados.</p>	O
RCA3.1.17	<p>Garantir a inserção de outros metadados após a captura.</p> <p>Por exemplo, data e hora de alteração e mudança de suporte.</p>	O
RCA3.1.18	Permitir ao usuário o registro de todas as versões de documento enquanto não for dada publicidade.	D

REF.	REQUISITO	OBRIG.
RCA3.1.19	Registrar a versão final do documento institucional após ter sido dada publicidade ou assinado digitalmente.	O
RCA3.1.20	<p>Prestar assistência aos usuários no que diz respeito à classificação dos documentos, por meio de algumas ou de todas as ações que se seguem:</p> <ul style="list-style-type: none"> <li>▪ Tornar acessível ao usuário somente o subconjunto do plano de classificação que diz respeito à sua atividade.</li> <li>▪ Indicar as últimas classificações feitas pelo usuário.</li> <li>▪ Indicar dossiês que contenham documentos institucionais relacionados.</li> <li>▪ Indicar classificações possíveis a partir dos metadados já inseridos, como, por exemplo, classe de apelação criminal quando se tratar de processo criminal e com apelação recebida ou pauta de julgamento.</li> <li>▪ Indicar classificações possíveis a partir do conteúdo do documento.</li> </ul>	D
RCA3.1.21	Permitir que um usuário disponibilize documentos a outro usuário para complementar o processo de captura, no caso dos procedimentos dessa captura serem executados por vários usuários.	O
RCA3.1.22	<p>No caso de documentos ou processos/dossiês constituídos por mais de um objeto digital:</p> <ul style="list-style-type: none"> <li>▪ Tratar o documento como uma unidade indivisível, assegurando a relação entre os objetos digitais.</li> <li>▪ Preservar a integridade do documento, mantendo a relação entre os objetos digitais.</li> <li>▪ Garantir a integridade do documento quando da recuperação, visualização e gestão posteriores.</li> <li>▪ Gerenciar a destinação de todos os objetos digitais que compõem o documento como uma unidade indivisível.</li> </ul>	O
RCA3.1.23	<p>Emitir um aviso caso o usuário tente registrar um documento que já tenha sido registrado no mesmo processo/dossiê.</p> <p>Para realizar a comparação devem ser observados os metadados dos documentos.</p>	O
RCA3.1.24	Impedir a reinserção de documentos digitais que forem detectados como idênticos.	O
RCA3.1.25	Permitir a captura dos documentos não-digitais e/ou híbridos.	O

REF.	REQUISITO	OBRIG.
RCA3.1.26	Acrescentar aos metadados dos documentos não-digitais informações sobre sua localização.	O

### 3.2 Captura em lote

REF.	REQUISITO	OBRIG.
RCA3.2.1	<p>Proporcionar a captura em lote de documentos gerados por outros sistemas. Esse procedimento tem que:</p> <ul style="list-style-type: none"> <li>▪ Permitir importação de transações predefinidas de arquivos em lote.</li> <li>▪ Registrar automaticamente cada um dos documentos importados contidos no lote.</li> <li>▪ Permitir e controlar a edição do registro dos documentos importados.</li> <li>▪ Validar a integridade dos metadados.</li> </ul> <p>Exemplos de lote de documento podem ser: mensagens do sistema de comunicação eletrônica, correspondência digitalizada por meio de escâner, documentos provenientes de uma unidade administrativa/órgão, de um grupo ou indivíduo, transações de aplicações de um computador ou ainda documentos oriundos de um sistema de gestão de processos e documentos.</p>	O

### 3.3 Captura de mensagens de sistema de comunicação eletrônica

O sistema de comunicação eletrônica é utilizado para criar, transmitir e receber mensagens eletrônicas e outros documentos digitais por meio de redes de computadores. As características do sistema de comunicação eletrônica podem dificultar seu gerenciamento. Assim, um GestãoDoc deve permitir controles de gestão para dotar os usuários da capacidade de capturar apenas mensagens e anexos previamente selecionados.

Esse procedimento requer que os usuários avaliem a pertinência e a importância dos itens, bem como o risco de não os capturar.

REF.	REQUISITO	OBRIG.
RCA3.3.1	Permitir a captura de mensagens de sistema de comunicação eletrônica após a seleção de quais serão objeto de registro.	O

REF.	REQUISITO	OBRIG.
RCA3.3.2	Assegurar a captura dos metadados referentes à mensagem de sistema de comunicação eletrônica, de tal forma que a confiabilidade e a autenticidade estejam garantidas.	O

### 3.4 Formato de arquivo e estrutura dos documentos a serem capturados

Os órgãos do Judiciário brasileiro precisarão realizar a captura de uma gama diversificada de documentos com formatos de arquivo e estruturas diferentes. Os requisitos técnicos para a captura variarão de acordo com a complexidade dos documentos. Em alguns ambientes não é possível identificar antecipadamente todas os formatos de arquivo e estruturas possíveis dos documentos, já que alguns são recebidos de fontes externas.

#### Documentos automodificáveis

Alguns documentos aparentam ter seus conteúdos alterados sem intervenção do usuário. Um exemplo é um modelo de mandado de citação ou intimação cuja data é colocada automaticamente pelo sistema e armazenada como um “campo” ou “código”. Nesse caso, cada vez que o documento é exibido, a data apresentada é atualizada. Entretanto o documento lógico não se modifica, apenas sua exibição (documento conceitual) sofre alterações conforme o *software* utilizado para visualizá-lo.

Outros documentos podem conter um código que os modifique realmente. É o caso de uma folha de cálculo com uma “macro” sofisticada que a altera (mediante *software* de aplicações utilizado para visualização) e, em seguida, guarda-a automaticamente.

Os documentos automodificáveis devem ser evitados. Caso isso não seja possível, os documentos devem ser armazenados em formatos que desativem o código automodificador ou visualizados por meio de *software* que não desencadeie a alteração. Por exemplo, uma planilha de cálculo que contenha “macros” deve ser convertida para um formato estável, como o PDF, antes de ser capturada no GestãoDoc.

Quando não for possível converter os documentos automodificáveis para formato estável ou visualizá-los por meio de *software* que não desencadeie a alteração, a captura desses documentos no GestãoDoc deve ser acompanhada do registro das informações relativas às funções automodificadoras nos metadados.

REF.	REQUISITO	OBRIG.
RCA3.4.1	Possuir a capacidade de capturar documentos nos formatos previamente definidos como padrão.	O

REF.	REQUISITO	OBRIG.
RCA3.4.2	Capturar, entre outros, os seguintes documentos: <ul style="list-style-type: none"> <li>▪ Informações de outros aplicativos: contabilidade, folha de pagamento, desenho assistido por computador (CAD).</li> <li>▪ Documentos em papel digitalizados por meio de escâner.</li> <li>▪ Documentos sonoros.</li> <li>▪ Videoclipes.</li> <li>▪ Diagramas e mapas digitais.</li> <li>▪ Dados estruturados.</li> </ul>	D
RCA3.4.3	Fornecer recursos que possibilitem o reconhecimento de caracteres a partir de documentos digitalizados, áudio vídeo e outras mídias.	O
RCA3.4.4	Capturar documentos que se apresentam com as seguintes estruturas: <ul style="list-style-type: none"> <li>▪ Simples: texto, imagens, mensagens sistema de comunicação eletrônica, slides digitais, som e vídeo.</li> <li>▪ Composta: mensagens de sistema de comunicação eletrônica com anexos, páginas Web e publicações eletrônicas.</li> </ul>	O
RCA3.4.5	Permitir que um documento composto seja capturado de qualquer uma das duas formas seguintes: <ul style="list-style-type: none"> <li>▪ Único documento de arquivo composto.</li> <li>▪ Série de documentos de arquivo simples relacionados, um para cada componente do documento composto.</li> </ul>	O
RCA3.4.6	Incluir novos formatos e arquivos à medida que forem adotados pelo Judiciário brasileiro.	O
RCA3.4.7	Armazenar em formato que desative o código automodificador quando da captura de documento automodificável.	O

### 3.5 Estrutura dos procedimentos de gestão

A gestão de processos e documentos digitais prevê o estabelecimento de três domínios dentro do ambiente eletrônico.

- Espaço individual — Designado a cada usuário autorizado.
- Espaço do grupo — Designado a cada grupo de trabalho, equipe, comitê etc.
- Espaço geral — Espaço no qual o documento não pode mais ser alterado e onde ocorre a publicidade dos documentos.

As regras estabelecidas pelo sistema de gestão de processos e documentos definem

em que espaços os documentos podem ser:

- Produzidos, recebidos, alterados, capturados (registrados, classificados, indexados e arquivados ou encaminhados), armazenados e eliminados.
- O espaço no qual os metadados são incluídos.
- Os direitos de acesso em cada espaço e a maneira pela qual os documentos tramitam dentro e fora do Judiciário brasileiro.

Uma vez capturados, os documentos e seus metadados devem ser mantidos em versão definitiva e protegidos contra alterações deliberadas ou acidentais.

REF.	REQUISITO	OBRIG.
RCA3.5.1	Reconhecer três domínios para o controle dos procedimentos de gestão: espaço individual, espaço de grupo e espaço geral.	O
RCA3.5.2	Operacionalizar as regras estabelecidas pelo sistema de gestão de processos e documentos nos três espaços.	O
RCA3.5.3	Impedir que o conteúdo de um documento seja alterado por usuários, gestores e administradores, exceto nos casos em que a alteração fizer parte do processo documental, conforme tratado na seção 6.10 — Alteração e exclusão de documentos institucionais.	O
RCA3.5.4	Emitir um aviso, ao se tentar capturar um documento cujos dados estruturados estejam incompletos e impedir quando estiverem inconsistentes.	O

## 4 Armazenamento

As considerações e as ações relativas ao armazenamento dos documentos institucionais não-digitais e digitais permeiam todo seu ciclo de vida. Esse armazenamento deve garantir a integridade e o acesso aos documentos.

Os documentos, independentemente do formato, requerem um armazenamento criterioso desde o momento de sua criação, para garantir sua preservação de longo prazo.

Num cenário híbrido que envolve ao mesmo tempo documentos institucionais não-digitais e digitais, deve-se atentar para requisitos de armazenamento que atendam igualmente às necessidades desses dois tipos de documentos.

As condições de armazenamento devem considerar o volume e as propriedades físicas dos documentos. Devem ser projetadas também em razão da proteção contra o acesso não-autorizado e perdas por destruição, furto e sinistro.

No caso dos documentos institucionais digitais, o Judiciário brasileiro deve dispor de políticas e diretrizes para conversão, migração ou acesso a esses documentos de maneira a

garantir sua autenticidade, acessibilidade e utilização. Os procedimentos de conversão e migração devem detalhar as mudanças ocorridas nos sistemas e nos formatos dos documentos. (Ver Capítulo 5, especificamente voltado à preservação)

Os fatores importantes na seleção das opções de armazenamento são:

- Volume e estimativa de crescimento dos documentos — Fator que deve ser considerado para se avaliar a capacidade de armazenamento — capacidade dos dispositivos de armazenamento e no caso de documentos não-digitais, áreas de depósito, tipos e quantidade de estantes.
- Segurança dos documentos — As instalações de armazenamento (depósitos, arquivos, computadores) devem prever limitação de acesso aos documentos, como, por exemplo, controle das áreas de armazenamento e sistemas de detecção de entradas não-autorizadas. O depósito deve estar localizado em área que não seja de risco. No caso de documentos digitais, devem ser previstos procedimentos que previnam a perda de documentos por falha do GestãoDoc. (Ver Capítulo 6, Segurança)
- Características físicas do suporte e do ambiente — Fatores como tipo de suporte, peso, grau de contaminação do documento e do ambiente, temperatura e umidade influenciarão na adequação das condições de armazenamento. Nesse sentido, devem ser adotados procedimentos — como o controle e verificação do tempo de vida útil e da estabilidade dos suportes — para prevenir quaisquer danos aos documentos. É importante que os meios de acondicionamento sejam robustos e adequados ao formato e à quantidade de documentos. As áreas de depósito devem ter amplitude adequada, estabilidade de temperatura e de níveis de umidade, proteção contra sinistro, contaminação (tais como isótopos radioativos, toxinas e mofo) e infestação de insetos e microorganismos. Os documentos digitais devem passar periodicamente pela troca de suporte — transferência de informações contidas num suporte para outro. Essa técnica é conhecida por rejuvenescimento (*refreshing*).
- Frequência de uso — O uso mais ou menos freqüente dos documentos deve ser considerado na seleção das opções de armazenamento. No caso dos documentos não-digitais, as opções envolverão acondicionamento (pastas suspensas, caixas entre outros) e localização dos depósitos (próximos ou distantes da área de trabalho). Já em relação aos documentos digitais, as opções podem envolver armazenamento *on-line* (acesso imediato) ou *off-line*, nas chamadas “mídias removíveis” de armazenamento (disco óptico, fita magnética e outros) em diferentes graus de disponibilidade e velocidade.
- Custo relativo das opções de armazenamento dos documentos — Além do custo dos dispositivos de armazenamento, deve ser considerado o dos equipamentos para sua manipulação e de *software* de controle.

Os documentos digitais são armazenados em dispositivos de armazenamento eletrônicos, magnéticos e ópticos. Do ponto de vista tecnológico, distinguem-se três tipos de memória, em ordem decrescente de preço e velocidade de acesso: primária, secundária e



terciária.

As memórias secundária e terciária são adequadas para armazenamento.

Os equipamentos devem adequar-se às características das operações — *on-line* ou *off-line*. Operações *on-line* só podem ser realizadas por meio GestãoDoc, ao passo que operações *off-line* podem ser realizadas em outros sistemas computacionais, desvinculadas do funcionamento do GestãoDoc.

Seja qual for a tecnologia empregada, um GestãoDoc deve garantir a integridade dos documentos institucionais.

Os itens seguintes enumeram requisitos de armazenamento organizados segundo os critérios de durabilidade, capacidade e viabilidade técnica.

## 4.1 Durabilidade

Os dispositivos de armazenamento de um GestãoDoc e os documentos nele armazenados devem estar sujeitos a ações de preservação que garantam sua conservação de longo prazo.

REF.	REQUISITO	OBRIG.
RAR4.1.1	Utilizar dispositivos e padrões estáveis no mercado.  Utilizar preferencialmente padrões abertos de armazenamento (como exemplo: ISO 9660:1988 — definição do formato de sistema de arquivos para CD-Rom).  A escolha dos dispositivos de armazenamento deve contemplar padrões estáveis de mercado e fornecedores consolidados.	O
RAR4.1.2	Avaliar periodicamente a escolha de dispositivos sempre que a evolução tecnológica indicar mudanças importantes.	O
RAR4.1.3	Efetuar migrações preventivas sempre que se tornar patente ou previsível a obsolescência do padrão corrente.	O
RAR4.1.4	Manter o registro de MTBF ( <i>Mean Time Between Failures</i> ) <sup>2</sup> , MTRR ( <i>Mean Time To Repair</i> ) <sup>3</sup> e MTBSI ( <i>Mean Time Between Service Incidents</i> ) <sup>4</sup> para as memórias secundárias, bem como as datas de sua aquisição.	O

<sup>2</sup> *Mean Time Between Failures* (MTBF) — Tempo médio entre falhas. É um valor relativo ao período médio entre falhas de um

REF.	REQUISITO	OBRIG.
RAR4.1.5	<p>Realizar o gerenciamento das mídias por meio do registro de durabilidade prevista, data de aquisição e histórico de utilização das memórias secundária e terciária.</p> <p>Informações técnicas sobre previsibilidade de duração de mídias referidas em RAR4.1.4 devem ser obtidas preferencialmente a partir de órgãos independentes. Quando isso não for possível, podem ser utilizadas informações de fornecedores.</p> <p>A origem da informação deve ficar registrada em ambos os casos.</p>	O
RAR4.1.6	<p>Manter estatísticas da durabilidade efetivamente observada das memórias secundária e terciária.</p>	O
RAR4.1.7	<p>Utilizar preferencialmente as redes de dados para o acesso às informações armazenadas em memória terciária.</p> <p>O objetivo é minimizar o acesso físico às mídias, visando à diminuição do desgaste. A manipulação direta das mídias deverá ser realizada preferencialmente por meio de sistemas automáticos de manipulação de mídias.</p>	D
RAR4.1.8	<p>Quando se proceder à eliminação de documentos, as memórias de suporte devem ser devidamente "sanitizadas", isto é, ter suas informações efetivamente indisponibilizadas.</p> <p>A eliminação de um documento não implica a eliminação de seus metadados.</p> <p>Esse requisito aplica-se principalmente às memórias secundária e terciária, pela sua característica não-volátil. As informações devem ser eliminadas de forma irreversível, incluindo, no caso de memória terciária, a possibilidade de destruição física das mídias.</p>	O

## 4.2 Capacidade

Um GestãoDoc deve garantir a escalabilidade no armazenamento, permitindo a expansão ilimitada de seus dispositivos.

sistema ou dispositivo e que permite a avaliação de sua confiabilidade ou vida útil.

<sup>3</sup> Mean Time To Repair (MTTR) - Tempo médio de recuperação. É um valor relativo ao período médio para colocar um componente ou sistema defeituoso em funcionamento novamente.

<sup>4</sup> MTBSI ( Mean Time Between Service Incidents) É o tempo médio entre falhas. MTBSI = MTBF+MTTR.

REF.	REQUISITO	OBRIG.
RAR4.2.1	Possuir capacidade de armazenamento suficiente para a acomodação de todos os documentos, metadados e suas cópias de segurança.	O
RAR4.2.2	<p>Prever a possibilidade de expansão da estrutura de armazenamento.</p> <p>A quantidade de memória primária deve ser dimensionada adequadamente no momento da aquisição, a fim de minimizar as indisponibilidades do GestãoDoc nas situações de expansão desse tipo de memória.</p> <p>Quando da aquisição de memória secundária e terciária as possibilidades de expansão dos equipamentos de controle devem ser consideradas.</p>	O
RAR4.2.3	Permitir ao administrador a configuração dos limites de capacidade de armazenamento dos diversos dispositivos.	D
RAR4.2.4	<p>Oferecer ao administrador facilidades para a monitoração da capacidade de armazenamento.</p> <p>Esse controle indica, por exemplo, capacidade utilizada, capacidade disponível e taxa de ocupação. Tais informações são úteis para subsidiar ações de expansão em tempo hábil.</p>	O
RAR4.2.5	Informar automaticamente ao administrador quando os dispositivos de armazenamento <i>on-line</i> atingirem níveis de alerta e níveis críticos de ocupação.	O
RAR4.2.6	<p>Manter estatísticas de taxa de crescimento de utilização de memória secundária e terciária para fornecer ao administrador previsões de exaustão de recursos.</p> <p>Esse tipo de estimativa possibilita ao administrador antecipar ações de expansão antes que a utilização atinja níveis críticos.</p>	O
RAR4.2.7	Permitir a definição de outras estatísticas referentes à capacidade de armazenamento de acordo com as necessidades específicas do Judiciário brasileiro.	D

### 4.3 Efetividade de armazenamento

REF.	REQUISITO	OBRIG.
RAR4.3.1	Os dispositivos de armazenamento devem suportar métodos de detecção de erros para leitura e escrita de dados e prover mecanismos automáticos de aviso ao administrador do sistema.	O
RAR4.3.2	Utilizar técnicas de restauração de dados em caso de falhas.	O
RAR4.3.3	Utilizar mecanismos de proteção que previnam alterações indevidas e mantenham a integridade dos dados armazenados.	O
RAR4.3.4	Prever a utilização de técnicas para garantir maior confiabilidade e desempenho.  As técnicas recomendadas incluem redundância e paralelismo.	D
RAR4.3.5	A integridade dos dispositivos de armazenamento deve ser periodicamente verificada.	O

## 5 Preservação

Os documentos institucionais devem ser mantidos acessíveis e utilizáveis por todo o tempo que se fizer necessário, com vistas a garantir sua longevidade, funcionalidade e disponibilidade. Deverão ser asseguradas as características dos documentos — tais como: autenticidade, integridade e acessibilidade — pela adoção de estratégias institucionais e técnicas proativas de criação e de preservação, que garantam sua perenidade. Essas estratégias são estabelecidas por uma política de preservação. As razões para preservação de um determinado documento normalmente estão associadas a seu valor probatório e informativo.

Tradicionalmente, a preservação de documentos institucionais se concentra na obtenção da estabilidade do suporte da informação. Nos documentos não-digitais, o conteúdo e o suporte estão intrinsecamente ligados, assim, a manutenção do suporte garante a preservação do documento. De forma distinta, nos documentos digitais o foco da preservação é a manutenção do acesso, que pode exigir mudança de suporte e formatos, bem como a atualização do ambiente tecnológico. A fragilidade do suporte digital e a obsolescência tecnológica de *hardware*, *software* e formato exigem essas intervenções periódicas.

As estratégias de preservação de documentos institucionais podem ser divididas em gerais e específicas da tecnologia digital.

As primeiras incluem monitoramento e controle ambiental das instalações, restrições de acesso, cuidados no manuseio de equipamentos e material de acervo.

Já as estratégias voltadas para tecnologia digital compreendem cuidados quanto à durabilidade das mídias e à atualização da base tecnológica.

O problema da durabilidade das mídias pode ser contornado mediante cópias periódicas para novas mídias (*refreshing*).

Por outro lado, cuidados com a atualização da base tecnológica exigem planejamento e estratégias mais complexas, envolvendo mudanças de *software* e de *hardware*. O desafio a ser enfrentado é vencer a obsolescência tecnológica.

As mudanças em *software* — incluindo sistemas operacionais, sistemas de gerenciamento de banco de dados e aplicativos como editores de texto, planilhas eletrônicas, editores de imagem, entre outros — costumam ser bastante freqüentes. Os *softwares* podem ser simplesmente descontinuados, substituídos por outros equivalentes ou superiores, ou ainda ter sua versão atualizada para correção de *bugs* ou acréscimo de novas funcionalidades. É importante notar que os fornecedores de *software* deixam de prestar suporte a versões mais antigas de seus produtos.

Os formatos também sofrem alterações, muitas vezes em função de mudanças ocorridas nos programas aos quais estão associados. Novos programas podem ser compatíveis com os formatos antigos, mas também podem apresentar incorreções durante operações de leitura e escrita de dados nesses formatos.

Algumas técnicas comumente utilizadas para evitar os riscos provenientes da obsolescência tecnológica são:

- **Preservação da tecnologia** — Evita a necessidade imediata de implementação de novos sistemas. Porém, as necessidades de manutenção e integração com outros sistemas podem apresentar problemas ao longo do tempo. A preservação do *hardware*, em especial, é uma alternativa cara, mesmo nas situações em que o *hardware* é compartilhado entre mais de um usuário. Além disso, essa alternativa não é exequível a longo prazo, uma vez que o *hardware* pode ser danificado de forma irreversível, ficando completamente indisponível.
- **Emulação** — É a simulação de um determinado *hardware* ou *software* por meio de *software*. Permite que um computador moderno, possivelmente mais barato e de fácil manutenção, possa executar programas antigos desenvolvidos originalmente para outra plataforma. Para evitar possíveis perdas de informação e funcionalidades, deve ser realizada com bastante rigor. A probabilidade de ocorrência de perdas de informações e funcionalidades aumenta à medida que são utilizadas diversas camadas de emulação, como resultado da aplicação dessa técnica repetidas vezes.
- **Conversão de dados** — É empregada quando os formatos se tornam obsoletos. Os dados em formatos antigos são convertidos para novos formatos, apoiados em *hardware* e *software* mais atuais. Esse processo não está isento de problemas, podendo resultar em perdas de informações e funcionalidades. A conversão de dados também pode ser utilizada para reduzir a quantidade de formatos utilizados e, conseqüentemente, de sistemas a serem mantidos e gerenciados, de modo a facilitar as ações de preservação.
- **Migração** — A migração para novos sistemas é realizada no caso de obsolescência de elementos — *hardware*, *software* e formatos. Envolve, inclusive, a conversão de dados.

Pode abranger uma grande quantidade desses elementos e, dessa forma, apresentar uma maior complexidade para ser planejada e executada. Apesar disso, mostra-se como uma alternativa interessante para o acompanhamento das mudanças decorrentes da evolução tecnológica. A migração, assim como a emulação e a conversão de dados, apresenta riscos quanto à integridade e à funcionalidade dos documentos institucionais digitais, por isso, deve ser realizada de modo criterioso e sistemático.

Embora os problemas de degradação dos suportes e obsolescência tecnológica possam ser contornados com conhecimento técnico e utilização de técnicas de preservação, sua execução pode ser muito dispendiosa. Por isso, as preocupações com preservação devem existir desde a concepção do GestãoDoc e a escolha de sua base tecnológica. De uma forma geral, recomenda-se a utilização de suportes de alta qualidade e que tenham uma vida útil prevista adequada para os propósitos de preservação, o monitoramento contínuo dos avanços tecnológicos e da degradação do suporte, a adoção de formatos abertos e a busca por soluções independentes de *hardware*, *software* e fornecedor.

As estratégias e os procedimentos de preservação devem ser bem definidos, documentados e periodicamente revisados. É importante destacar que as ações de preservação são contínuas e devem ser implementadas desde a produção dos documentos até sua destinação final.

Qualquer que seja a estratégia de preservação adotada, há de se documentar os procedimentos e as estruturas de metadados.

Nesta seção, não se pretende apresentar procedimentos de preservação preestabelecidos ou argumentar em favor de uma técnica específica. Os requisitos foram organizados em aspectos físicos, lógicos e gerais.

É atribuição do Conselho Nacional de Justiça a definição e o acompanhamento da implementação da política de preservação de documentos institucionais do Judiciário brasileiro.

## 5.1 Aspectos físicos

REF.	REQUISITO	OBRIG.
RPR5.1.1	Os suportes de armazenamento devem ser acondicionados, manipulados e utilizados em condições ambientais compatíveis com sua vida útil prevista e/ou pretendida, dentro das especificações técnicas de seu fabricante e de entidades isentas e com base em estatísticas de utilização.	O
RPR5.1.2	Permitir ao administrador especificar a vida útil prevista/preendida dos suportes.	O
RPR5.1.3	Permitir o controle da vida útil dos suportes para auxiliar no processo de rejuvenescimento.	O

REF.	REQUISITO	OBRIG.
RPR5.1.4	Informar automaticamente quais são os suportes que se encontram próximos do fim de sua vida útil.	O

## 5.2 Aspectos lógicos

Um GestãoDoc deve garantir escalabilidade no armazenamento, permitindo expansão ilimitada dos dispositivos de armazenamento.

REF.	REQUISITO	OBRIG.
RPR5.2.1	Manter cópias de segurança que devem ser guardadas em ambientes adequados segundo a política de segurança da informação. O armazenamento das cópias de segurança deve ser realizado em local diferente de onde se encontra a informação original.  As informações mantidas em mídia terciária devem ser duplicadas e armazenadas em locais diferentes.	O
RPR5.2.2	Possuir funcionalidades para a verificação periódica dos dados armazenados, visando detecção, reparação e informação de possíveis erros.  Nesse caso, recomenda-se a utilização de um checksum robusto, que permita a constatação da integridade dos dados e seja seguro quanto a fraudes.	O
RPR5.2.3	Permitir ao administrador a reparação dos dados armazenados que apresentarem erros.	O
RPR5.2.4	Manter um histórico dos resultados da verificação periódica dos dados armazenados.	O
RPR5.2.5	Efetivar ações de preservação sempre que verificada obsolescência tecnológica ou quando favoreça a padronização da plataforma tecnológica do Judiciário brasileiro.	O
RPR5.2.6	Suportar a transferência em bloco de documentos e metadados para outros sistemas.	O

## 5.3 Aspectos gerais

REF.	REQUISITO	OBRIG.
RPR5.3.1	Registrar as operações de preservação realizadas em trilhas de auditoria.	O
RPR5.3.2	Utilizar suportes de armazenamento, recursos de <i>hardware</i> e de <i>software</i> que sejam estáveis no mercado e amplamente disponíveis e que contribuam para a padronização e uniformização da plataforma tecnológica do Judiciário brasileiro.	D
RPR5.3.3	As modificações em um GestãoDoc e em sua base tecnológica devem ser verificadas em um ambiente exclusivo para essa finalidade, de modo a garantir que, após a implantação das alterações, os dados continuem sendo acessados sem alteração de conteúdo.	O
RPR5.3.4	Utilizar normas amplamente aceitas, descritas em especificações abertas e disponíveis publicamente, no que refere a estruturas para codificação, armazenamento e banco de dados.	D
RPR5.3.5	Evitar a utilização de estruturas proprietárias, para codificação, armazenamento ou banco de dados.	D
RPR5.3.6	Nos caso em que se utilize estruturas proprietárias, para codificação, armazenamento ou banco de dados, elas devem estar plenamente documentadas (incluindo o motivo para a utilização dessas estruturas proprietárias) e essa documentação, disponível para o administrador.	O
RPR5.3.7	Gerir metadados relativos à preservação dos documentos e seus respectivos componentes.	O

## 6 Segurança

O sistema de gestão de documentos deve prever controles de acesso e procedimentos de segurança que garantam a confidencialidade, a integridade, a disponibilidade e a autenticidade dos documentos. Dentre esses procedimentos, pode-se destacar a utilização de controles técnicos e programáticos, diferenciando tipos de documentos, perfis de usuários e característica de acesso aos dados, manutenção de trilhas de auditoria e de rotinas de cópias de segurança.

Além disso, também devem ser consideradas exigências e procedimentos de segurança da infra-estrutura das instalações.

Problemas de segurança não são resolvidos apenas com tecnologia, já que envolvem



características do comportamento humano. Por isso, o GestãoDoc deve ser projetado, desenvolvido e mantido em consonância com uma Política de Segurança de Informação,

### **Controle de acesso**

Um GestãoDoc deve limitar ou autorizar o acesso a documentos, por usuário e/ou papéis. Nessa acepção papéis representam conjuntos de usuários com mesmos perfis de atividade do ponto de vista do GestãoDoc, tendo os mesmos direitos de acesso.

O controle de acesso deve garantir, no mínimo, as seguintes funções:

- Restrição de acesso aos documentos.
- Exibição dos documentos, criptografados ou não, e dos metadados somente aos usuários autorizados.
- Uso e intervenção nos documentos somente pelos usuários autorizados.

Os documentos também devem ser analisados em relação à confidencialidade: ostensivos, reservados, sigilosos etc. Regras, normas e legislação<sup>5</sup> estabelecem diferentes razões para o sigilo e também diferentes graus a serem atribuídos a cada documento e as autoridades competentes para fazê-lo. (Ver Capítulo 3, Captura — Atribuição de restrição de acesso)

Um sistema de gestão de processos e documentos deve garantir que apenas usuários autorizados tenham acesso à informação sigilosa. O acesso aos metadados dos documentos sigilosos deve ser estabelecido com base nas diretrizes para o tratamento de processos e investigações sigilosas ou que tramitam em segredo de justiça, no que diz respeito à autuação, processamento, transporte, inserção de dados no sistema eletrônico de informações processuais, consulta e arquivamento, conforme estabelecido nos atos normativos das instituições do Judiciário.

O monitoramento e mapeamento das permissões de acesso são um processo contínuo em todos os sistemas de gestão de documentos.

### **Uso e rastreamento**

O uso dos documentos pelos usuários deve ser registrado pelo sistema nos seus respectivos metadados. A gestão desse uso inclui:

- Identificação da permissão de acesso dos usuários, isto é, o que ele pode acessar.
- Identificação dos níveis de segurança e da categoria de sigilo dos documentos.
- Garantia de que somente os indivíduos autorizados tenham acesso aos

---

<sup>5</sup> Lei nº 8.159, de 8 de janeiro de 1991; Lei nº 11.111, de 5 de maio de 2005; Decreto nº 4.553 de 27 de dezembro de 2002 e Decreto nº 5.301, de 9 de dezembro de 2004.

originalmente sigilosos ou aos posteriormente classificados.

- Registro de todos os acessos, tentativas de acesso e usos dos documentos (visualização, impressão, transmissão e cópia para a área de transferência) com identificação de usuário, data, hora e, se possível, a estação de trabalho.
- Revisão periódica das classificações de acesso a fim de garantir sua atualização.

O rastreamento dos documentos em trilhas de auditoria é uma medida de segurança que tem por objetivo verificar a ocorrência de acesso aos documentos e seu uso indevido. O grau de controle de acesso e o detalhamento do registro na trilha de auditoria dependem da natureza do órgão, dos documentos produzidos e deverá refletir o nível de preocupação da política de segurança da informação das instituições do Judiciário brasileiro.

### **Trilha de auditoria**

A trilha de auditoria é o conjunto de informações registradas que permite o rastreamento de intervenções ou tentativas de intervenções feitas no documento institucional digital ou no GestãoDoc.

A trilha de auditoria deve registrar o movimento e a utilização dos documentos institucionais dentro de um GestãoDoc (captura, registro, classificação, indexação, arquivamento, armazenamento, recuperação da informação, acesso e utilização, preservação e destinação), informando quem operou, a data, a hora e as ações tomadas. A trilha de auditoria tem o objetivo de fornecer informações sobre o cumprimento das políticas e regras da gestão de documentos das instituições do Judiciário e serve para:

- Identificar os autores de cada operação sofrida pelos documentos.
- Prevenir a perda de documentos.
- Monitorar todas as operações realizadas no GestãoDoc.
- Garantir a segurança e a integridade do GestãoDoc.

No caso de procedimentos que tenham prazos a serem cumpridos pelas instituições do Judiciário, devem-se implementar ações de rastreamento de forma a:

- Determinar os passos a serem dados em resposta às atividades ou ações registradas em um documento.
- Atribuir responsabilidade por uma ação a uma pessoa.

### **Cópias de segurança**

O GestãoDoc deve prever controles para proporcionar a salvaguarda regular dos documentos institucionais e dos seus metadados. Deve também poder recuperá-los rapidamente em caso de perda devido a sinistros, falhas no sistema ou de segurança ou degradação do suporte. Esses mecanismos devem seguir a política de segurança da informação das instituições do Judiciário.

No caso dos sistemas de gestão de documentos não-digitais pode-se prever a reprodução de documentos para outros suportes como medida de segurança, como, por exemplo, mediante processo de microfilmagem ou digitalização.

No caso dos sistemas de gestão de processos e documentos digitais, é aconselhável que o GestãoDoc contenha meios de monitoramento e acompanhamento da realização das cópias de segurança (*backup*). Esse processo consiste na realização de cópias periódicas das informações para restauração posterior das mesmas, em caso de perda devido a falhas de software, *hardware* ou mesmo acidente. O processo reverso ao *backup* é o de restauração (*restore*), que consiste em recuperar as informações para o ambiente de produção do GestãoDoc para um estado consistente.

Como o objetivo é restaurar o sistema em caso de falhas, as informações serão armazenadas conforme definido na política de segurança da informação das instituições do Judiciário. O procedimento de cópias de segurança não pode ser confundido com uma estratégia de preservação de longo prazo.

### **Segurança da infraestrutura**

A natureza das medidas de segurança da infra-estrutura de instalações do acervo digital diz respeito a requisitos operacionais e não é muito diferente daquela do acervo não-digital. Essas medidas devem considerar os seguintes aspectos:

- As salas reservadas a computadores servidores, equipamentos de rede e ao armazenamento dos documentos digitais devem ter temperatura ambiente e umidade relativa do ar controladas, fornecimento estável de energia elétrica e aterramento. Deve haver controle contínuo para verificar se essas condições são atendidas.
- Equipamentos contra incêndio devem ser providos em toda área de instalação e estar de acordo com as normas de segurança estabelecidas.
- A substituição dos equipamentos contra incêndio deve seguir uma rotina de verificação e ocorrer antes do final da vida útil a eles prevista.
- Instalações adequadas de pára-raios, com procedimentos de manutenção periódica, seguindo a legislação e normas técnicas já estabelecidas.
- A área reservada à instalação do GestãoDoc deverá ser restrita e com controle de acesso físico, com o objetivo de controlar o acesso às informações.
- As salas de infra-estrutura de tecnologia da informação são de utilização exclusiva de pessoal autorizado e devem possuir controle eletrônico de acesso.

Os requisitos de identificação, autenticação de usuário e trilhas de auditoria devem integrar qualquer GestãoDoc. Políticas de segurança da informação específicas poderão definir o rigor, maior ou menor, do tratamento dos demais requisitos. Atenção especial deve ser dada, nesse aspecto, às normas internacionais ISO 17799 e 27001.

No que diz respeito ao controle de acesso, esta especificação contempla três tipos de requisitos:

- Controle de acesso baseado em papéis de usuário.
- Controle de acesso por grupos.
- Classificação quanto ao grau de sigilo.

Os três tipos de controle de acesso podem ser combinados, e os requisitos de administração de controle de acesso devem ser adaptados a cada um dos tipos referidos anteriormente ou a combinação deles, de acordo com as normas institucionais.

Quanto à utilização da tecnologia de criptografia, tanto para sigilo quanto para autenticação, o rigor dos requisitos está sujeito às normas do ICP-Brasil e à política de segurança da informação das instituições do Judiciário. A criptografia pode ser utilizada como mecanismo de garantia de sigilo na transmissão de documentos, seja na cifragem da conexão estabelecendo canais seguros, seja na cifragem do documento transmitido ou capturado. Os requisitos de assinatura digital são necessários para as instituições que recebem documentos digitais assinados e onde são necessárias verificações de integridade e autenticidade. Nesses casos, o não-repúdio é garantido pela MP 2.200-2, de 2001, utilizando certificados digitais emitidos no âmbito da ICP-Brasil.

Esses requisitos não esgotam o tema segurança da informação, pois a segurança integral é sistêmica e abrange não somente a tecnologia, mas também pessoas, processos, ambiente e legislação.

## 6.1 Cópias de segurança

As cópias de segurança têm por objetivo prevenir a perda de informações, e garantir a disponibilidade do sistema. Os procedimentos de *backup* devem ser feitos regularmente e, pelo menos uma cópia deve ser armazenada remotamente (*off-site*).

Podem-se distinguir vários tipos de informações necessários ao funcionamento de um GestãoDoc. Essas informações compreendem os documentos digitais, metadados e informações de controle associadas às camadas de *software* relacionadas ao GestãoDoc (sistema operacional, gerenciador de bancos de dados, *software* aplicativo). Todas essas informações devem ser incluídas nos procedimentos de cópias de segurança.

REF.	REQUISITO	OBRIG.
RSE6.1.1	Cumprir a política de segurança da informação das instituições do Judiciário.	O
RSE6.1.2	Possibilitar o acompanhamento das ações efetivadas de <i>backup</i> e <i>restore</i> .	D

## 6.2 Controle de acesso

Esta seção trata dos requisitos de identificação e autenticação de usuários, controle de acesso baseado em papéis de usuários, bem como dos requisitos comuns a qualquer tipo de controle de acesso.

### Identificação e autenticação de usuários

Os requisitos a seguir tratam do mapeamento da identidade do usuário e das permissões concedidas a ele, imediatamente após sua autenticação.

Usuários acessam informações e funcionalidades por meio da interface do programa. A associação entre identidade do usuário e as autorizações de acesso é feita durante a fase de identificação e autenticação do usuário por meio da interface do programa, com base nas credenciais de autenticação.

REF.	REQUISITO	OBRIG.
RSE6.2.1	Implementar o controle de acesso, mantendo pelo menos os seguintes atributos dos usuários, de acordo com a política de segurança da informação das instituições do Judiciário: <ul style="list-style-type: none"> <li>▪ Identificador do usuário.</li> <li>▪ Autorizações de acesso.</li> <li>▪ Credenciais de autenticação.</li> </ul> Senha, chave criptográfica e biometria são exemplos de credenciais de autenticação.	O
RSE6.2.2	Utilizar, para efeito de autenticação, um sistema de gerenciamento de identidade externo .	D
RSE6.2.3	Exigir que o usuário esteja devidamente identificado e autenticado antes que este inicie qualquer operação no sistema.	O
RSE6.2.4	Garantir que os valores dos atributos de segurança e controle de acesso, associados ao usuário estejam dentro de conjuntos de valores válidos.	O
RSE6.2.5	Garantir que as tecnologias de credenciais de autenticação só possam ser modificadas pelo administrador, em conformidade com a política de segurança da informação das instituições do Judiciário.	O
RSE6.2.6	Permitir avaliação periódica dos direitos de acesso dos usuários do sistema.	O

### Aspectos gerais de controle de acesso

Os requisitos desta seção são aplicáveis independentemente do modelo de controle de acesso adotado, de acordo com a política de segurança da informação.

REF.	REQUISITO	OBRIG.
RSE6.2.7	Permitir o acesso a funções administrativas do sistema somente a usuários autorizados e sob controle rigoroso do gestor.	O

REF.	REQUISITO	OBRIG.
RSE6.2.8	<p>Fornecer uma das seguintes respostas (estabelecidas durante a configuração) se o usuário solicitar o acesso ou pesquisa em um documento institucional, volume ou processo/dossiê específicos aos quais não tenha o direito de acesso:</p> <ul style="list-style-type: none"> <li>▪ Mostrar determinados dados cadastrais do documento ou processo.</li> <li>▪ Demonstrar a existência do processo/dossiê ou documento mas não seu conteúdo (exemplo: informações pessoais).</li> <li>▪ Não mostrar qualquer informação constante do documento, nem indicar sua existência (exemplo: quebra de sigilo telefônico).</li> </ul>	O
RSE6.2.9	Garantir que somente o gestor seja capaz de criar, alterar, remover ou revogar as permissões associadas a perfis de usuários, grupos de usuários ou usuários individuais.	O
RSE6.2.10	Implementar imediatamente alterações ou revogações dos atributos de segurança de usuários e de documentos digitais.	D
RSE6.2.11	Oferecer ferramentas de aumento de produtividade ao gestor, tais como: realização de operações sobre lotes ou grupos de usuários e lotes de documentos digitais, agenda de tarefas, análises de trilhas e geração de alarmes.	O
RSE6.2.12	Quando um GestãoDoc controlar o acesso por grupos de usuários, perfis de usuários e usuários individuais obedecer a uma hierarquia de permissões preestabelecida na política de segurança da informação das instituições do Judiciário.	O

### **Controle de acesso por papéis de usuários**

Papéis são funções ou cargos com responsabilidades e autoridades bem definidas.

Operações são tarefas executadas sobre os documentos e os processos/dossiês.

Atribuições de usuários são as associações entre usuários e papéis. Um usuário pode estar associado a um ou mais papéis e vice-versa.

Permissões são garantias aprovadas para realização de operações sobre documentos institucionais.

REF.	REQUISITO	OBRIG.
RSE6.2.13	Utilizar os seguintes atributos do usuário ao implementar a política de controle de acesso por perfis de usuários sobre documentos: <ul style="list-style-type: none"><li>▪ Identificação do usuário.</li><li>▪ Perfis associados ao usuário.</li></ul>	O
RSE6.2.14	Utilizar os seguintes atributos dos documentos ao implementar a política de controle de acesso por perfis: <ul style="list-style-type: none"><li>▪ Identificação do documento.</li><li>▪ Operações permitidas para os vários perfis de usuários, sobre as unidades a que pertence o documento.</li></ul>	O
RSE6.2.15	Conceder acesso a documentos, processos/dossiês somente se a permissão requerida para a operação estiver presente em pelo menos um dos perfis associados ao usuário.	O
RSE6.2.16	Impedir que um usuário assuma perfis com direitos conflitantes. Em caso de conflito, prevalece o perfil mais restritivo.	O
RSE6.2.17	Permitir a criação de hierarquias de perfis e o conceito de herança de permissões entre eles.	O

### 6.3 Classificação da informação quanto ao grau de sigilo e restrição de acesso à informação sensível

Os requisitos descritos nesta seção referem-se ao acesso aos processos/dossiês e documentos com base na classificação do grau de sigilo bem como restrição de acesso à informação sensível.

Nos processos judiciais a informação sensível é comumente tratada de duas formas:

- Considera-se em segredo de justiça a investigação, o processo, os dados e as informações determinadas pela autoridade judicial competente para o feito, nos termos da legislação aplicável à matéria.
- Considera-se sigilosa, quando determinada pela autoridade judicial competente, toda a informação, documento, elemento ou feito que, por sua natureza ou quando a preservação de direitos individuais e o interesse público o exigirem deva ser de conhecimento restrito e, portanto, requeira medidas especiais para segurança de seu conteúdo. O caráter sigiloso poderá ser atribuído ao processo ou às partes. Quando atribuído ao processo, a consulta ao sistema será restrita a pessoas autorizadas, a critério da autoridade judicial.
-

REF.	REQUISITO	OBRIG.
RSE6.3.1	Aceitar a definição de graus de sigilo e de perfis de usuários de acordo com as necessidades das instituições do Judiciário.	O
RSE6.3.2	Implementar a classificação de grau de sigilo baseando-se nos seguintes atributos de segurança para documentos e para usuários: <ul style="list-style-type: none"> <li>▪ Grau de sigilo do documento.</li> <li>▪ Credencial de segurança do usuário.</li> </ul>	O
RSE6.3.3	Recusar o acesso de usuários a documentos que possuam um grau de sigilo superior à sua credencial de segurança.	O
RSE6.3.4	Garantir que os documentos sem atribuição de grau de sigilo, importados a partir de fontes externas ao GestãoDoc, estejam sujeitos às políticas de controle de acesso e de sigilo.	O
RSE6.3.5	Manter a marcação de sigilo original durante a importação de documentos marcados com graus de sigilo, a partir de fontes externas ao GestãoDoc.	O
RSE6.3.6	Garantir a não-ambigüidade na associação entre as marcações de grau de sigilo e os outros atributos de segurança (permissões) do documento importado.	D
RSE6.3.7	Garantir que nos casos em que grau de sigilo e atributos de segurança incidam sobre um mesmo documento, o critério de acesso seja o de maior restrição.	O
RSE6.3.8	Permitir que o usuário autorizado seja capaz de alterar o grau de sigilo de todos os documentos institucionais de um processo/dossiê ou documento, em caso de erro, reavaliação, reclassificação ou redução do grau de sigilo.	O
RSE6.3.9	Garantir que o grau de sigilo de um documento importado esteja associado a um usuário autorizado com a credencial de segurança pertinente para receber o documento.	O
RSE6.3.10	Permitir somente ao gestor a possibilidade de alterar a configuração dos valores predefinidos ( <i>default</i> ) para os atributos de segurança e marcações de graus de sigilo, quando necessário e apropriado.	O
RSE6.3.11	Permitir somente aos usuários autorizados realizar as ações: criar, alterar, conceder ou revogar credenciais de segurança aos usuários.	O



REF.	REQUISITO	OBRIG.
RSE6.3.12	Prover mecanismos de proteção que permitam cópias de segurança de documentos confidenciais, preservando a inviolabilidade da informação.  Tais cópias poderão migrar para sites remotos, fora do controle do GestãoDoc. Por isso, os mecanismos citados deverão lançar mão de técnicas de criptografia.	O
RSE6.3.13	Impedir que um documento sigiloso seja eliminado, antes de se tornar ostensivo e ser submetido ao processo de avaliação para definição de sua destinação final, salvo as exceções legais.	O

## 6.4 Trilha de auditoria

A trilha de auditoria consiste num histórico de todas as intervenções, ou tentativas de intervenções, feitas no documento e no próprio GestãoDoc. Nesse sentido, é também um metadado sobre os documentos digitais e informa sobre a sua autenticidade.

REF.	REQUISITO	OBRIG.
RSE 6.4.1	Assegurar que as informações da trilha de auditoria estejam disponíveis para inspeção a fim de que uma ocorrência específica possa ser identificada e que todas as respectivas informações sejam claras e compreensíveis.	O

<b>REF.</b>	<b>REQUISITO</b>	<b>OBRIG.</b>
RSE6.4.2	Registrar na trilha de auditoria as informações: <ul style="list-style-type: none"> <li>▪ Data e hora da captura de todos os documentos.</li> <li>▪ Responsável pela captura.</li> <li>▪ Alteração do grau de sigilo de um documento ou de um processo/dossiê, registrando as modificações efetuadas.</li> <li>▪ Qualquer alteração nos instrumentos de classificação, temporalidade e destinação da política de gestão documental do Judiciário brasileiro.</li> <li>▪ Qualquer ação de reavaliação de documentos.</li> <li>▪ Qualquer alteração nos metadados associados a processos/dossiês ou documentos.</li> <li>▪ Data e hora de produção, aditamento e exclusão de metadados.</li> <li>▪ Usuário, data e hora de acesso ou tentativa de acesso a documentos e ao GestãoDoc.</li> <li>▪ Tentativas de acesso negado a qualquer documento.</li> <li>▪ Ações de exclusão de qualquer documento e seus metadados.</li> <li>▪ Todas as ações administrativas sobre os atributos de segurança (papéis, grupos, permissões etc.).</li> <li>▪ Todas as ações administrativas sobre dados de usuários (cadastro, ativação, bloqueio, atualização de dados e permissões, troca de senha etc.).</li> <li>▪ Todos os eventos de administração de manutenção das trilhas de auditoria (alarmes, cópias, configuração de parâmetros etc.).</li> </ul>	O
RSE6.4.3	Registrar, em cada evento auditado, informações sobre a identidade do usuário.	O
RSE6.4.4	Permitir a leitura das trilhas de auditoria apenas ao administrador e ao auditor.	O
RSE6.4.5	Possuir mecanismos para a realização de buscas nos eventos das trilhas de auditoria.	O
RSE6.4.6	Impedir qualquer modificação de conteúdo da trilha de auditoria.	O
RSE6.4.7	Permitir somente aos administradores a exportação e a transferência das trilhas de um suporte de armazenamento para outro, garantindo que em tais casos as informações não sejam comprometidas.	O

REF.	REQUISITO	OBRIG.
RSE6.4.8	Gerar um alarme, para os administradores, se o tamanho da trilha de auditoria exceder um limite preestabelecido.	O
RSE6.4.9	Aplicar um conjunto de regras na monitoração de eventos auditados e, com base nessas regras indicar a possível violação da segurança, como, por exemplo: <ul style="list-style-type: none"> <li>▪ Acumulação de um número predeterminado de tentativas consecutivas de <i>login</i> com erro (autenticação mal sucedida), conforme especificado pela política de segurança das instituições do Judiciário.</li> <li>▪ Ocorrência de vários <i>logins</i> simultâneos do mesmo usuário em locais (computadores) diferentes.</li> <li>▪ <i>Login</i> do usuário fora do horário autorizado, após <i>logout</i> no período normal.</li> </ul>	D
RSE6.4.10	Fornecer relatórios, em ordem cronológica, sobre as ações que afetam processos/dossiês e documentos.	O
RSE6.4.11	Garantir que somente administradores sejam capazes de configurar o conjunto de eventos auditáveis e seus atributos.	O
RSE6.4.12	Documentar em trilha de auditoria as configurações do GestãoDoc que redefinem o conjunto de eventos auditáveis.	O

## 6.5 Assinaturas digitais

A assinatura digital é um mecanismo para dar garantia de integridade e autenticidade a arquivos eletrônicos. A assinatura digital prova que a mensagem ou arquivo não foi alterado, e que foi assinado pela entidade ou pessoa que possui a chave privada e o certificado digital correspondente, utilizados na assinatura.

### Descrição do processo de assinatura digital

Para assinar digitalmente um arquivo, aplica-se inicialmente uma função matemática ao conteúdo do arquivo, obtendo-se um resumo criptográfico (*hash*) desse arquivo. A função *hash* garante a integridade de um documento na medida em que qualquer alteração no conteúdo desse documento altera o resultado da função *hash* aplicada sobre o mesmo.

O *hash* é então criptografado com a chave privada do signatário.

A criptografia de chave pública ou assimétrica permite verificar a autoria de um documento assinado digitalmente, uma vez que só é possível decifrar as informações, cifradas com determinada chave privada, utilizando-se a chave pública correspondente. Os pares de chaves são únicos. A chave privada é de posse e responsabilidade exclusiva de seu proprietário.

Os certificados digitais são documentos digitais que certificam a posse de um

determinado par de chaves por um indivíduo ou instituição.

O signatário de um documento, ao calcular o *hash* gera uma espécie de “impressão digital” do conteúdo do documento.

Ao criptografar o *hash* com sua chave privada, junta-se a sua própria “impressão digital” ao “pacote” composto nesse momento de original mais assinatura digital (*hash* criptografado).

Finalmente, o certificado digital do signatário é agregado ao pacote. Agregar o certificado ao pacote possibilita a divulgação da chave pública do signatário e a imediata verificação da validade do certificado e da integridade do documento. Assim, um arquivo assinado digitalmente geralmente compõe-se de:

- Original.
- Assinatura digital (*hash* criptografado — assinado).
- Certificado do signatário.

O receptor do “pacote”, inicialmente desempacota o certificado e utiliza as funções de PKI (*Public Key Infrastructure*) para fazer a verificação da validade do certificado e da cadeia de certificação. Validado o certificado, extrai-se a chave pública deste, aplicando-a à assinatura.

Só será possível decifrar a assinatura se a chave pública for correspondente à chave privada utilizada para a assinatura. Uma vez que a operação criptográfica se concretize estará estabelecida a autoria da assinatura e obtém-se o *hash* do documento.

Em seguida, aplica-se a função *hash* ao original e compara-se com o *hash* assinado. Dessa forma, é estabelecida a integridade do documento.

Toda operação descrita acima é feita de forma automática e transparente para o usuário, pelos assinadores digitais (que também fazem a verificação). Esses aplicativos emitem avisos caso ocorra falha na validação do documento ou certificado.

### **Não-repúdio e validade legal**

O não-repúdio é determinado pela relação do titular de um certificado e a autoridade certificadora, responsável por garantir:

- A identidade do titular do certificado.
- Que o titular do certificado gerou seu próprio par de chaves.
- Que o titular se compromete pela segurança e inviolabilidade da chave privada.

No Brasil, com o advento da ICP-Brasil e da MP nº 2.200-2, de 2001, foi estabelecida a validade legal de documentos assinados digitalmente, utilizando-se certificados digitais emitidos dentro da cadeia de certificação da ICP-Brasil.

A ICP-Brasil fiscaliza e audita o processo de emissão digital das autoridades certificadoras integrantes a fim de garantir a total confiabilidade do processo de certificação. Assim, dá respaldo à determinação legal de integridade, autenticidade e não-repúdio dos arquivos assinados digitalmente.

REF.	REQUISITO	OBRIG.
RSE6.5.1	Garantir a origem e a integridade dos documentos com assinatura digital.	O
RSE6.5.2	Utilizar o padrão ICP-Brasil quando houver necessidade de emprego de assinatura digital.	O
RSE6.5.3	Verificar a validade da assinatura digital no momento da captura do documento, e caso não esteja válida, recusar a captura.	O
RSE6.5.4	No processo de verificação da assinatura digital, registrar nos metadados do documento: <ul style="list-style-type: none"> <li>▪ Validade da assinatura verificada.</li> <li>▪ Autoridade certificadora do certificado digital.</li> <li>▪ Data e hora em que a verificação ocorreu.</li> </ul>	O
RSE6.5.5	Armazenar juntamente com o documento as informações de certificação: <ul style="list-style-type: none"> <li>▪ Assinatura digital.</li> <li>▪ Certificado digital (cadeia de certificação) usado na verificação da assinatura.</li> </ul>	O
RSE6.5.6	Receber atualizações tecnológicas quanto à plataforma criptográfica e padrões de assinatura digital.	O
RSE6.5.7	Acessar relógios e carimbador de tempo oficiais para o seu próprio uso.	O

## 6.6 Criptografia

Criptografia é a ciência do ocultamento de informações. Criptografar ou cifrar um arquivo significa alterar os *bits* de tal modo que a informação representada subsiste, mas de forma ininteligível. O processo inverso é decriptografar ou decifrar: recuperar a informação original, passível de interpretação e modificação.

A criptografia objetiva controlar o acesso à informação e não aos arquivos, de modo que o furto do arquivo não implica furto da informação.

### Criptografia simétrica e assimétrica

A criptografia baseia-se em chaves ou senhas, informações externas ao arquivo. Quando uma mesma chave serve para criptografar ("fechar") e para decriptografar ("abrir") o arquivo, tem-se **criptografia simétrica**.

**Criptografia assimétrica**, por sua vez, conta com um par de chaves: uma para fechar e outra para abrir. Curioso (e importante) é que são chaves alternáveis: um arquivo fechado com uma pode ser aberto com a outra chave do par e vice-versa. Porém, não se pode fechar e depois abrir com a mesma chave.

### Aplicabilidade

Criptografia simétrica é especialmente indicada quando um mesmo agente (usuário ou sistema) cripta e decripta a informação. É muito usada em bancos de dados. A ênfase é segurança em armazenamento e recuperação da informação.

Criptografia assimétrica prevê um agente emissor e um agente receptor, geralmente em sistemas distintos. A ênfase é a transmissão segura da informação. É base para as assinaturas e certificações digitais citadas no item anterior.

Na criptografia assimétrica, as chaves constituintes do par são designadas por chave privada (resguardada pelo usuário) e chave pública (de conhecimento público). O mecanismo de criptografia de chaves públicas, complementado pela infra-estrutura administrativa da ICP-Brasil, equaciona a questão de transmitir dados, de modo seguro, por meio inseguro (internet).

### Utilização de criptografia num GestãoDoc

Num GestãoDoc, o armazenamento e a recuperação de informações sigilosas (aí incluídos *backups* e *restores*) utilizará a criptografia simétrica. Já na comunicação, identificação de usuários e em sessões *Web*, será utilizada a criptografia assimétrica, em consonância com a ICP-Brasil.

A desvinculação entre as aplicações de criptografia torna o modelo mais ágil: por um lado, a dinâmica envolvida na criptografia externa fica sob responsabilidade da ICP-Brasil, sem necessidade de duplicação e atualização de informação dentro do GestãoDoc. Em particular, o GestãoDoc não terá a preocupação com aspectos como temporalidade de senhas públicas, listas de certificados ou a obsolescência tecnológica de *hardwares* e *softwares*.

Por outro lado, a utilização de criptografia simétrica no gerenciamento interno da informação, além da simplicidade conceitual, possibilita mais autonomia nos processos de controle do GestãoDoc. Por exemplo: cópias de segurança e *backups* passam a contar com mecanismos de proteção criptográfica com funcionamento independente do estado tecnológico da criptografia do mundo externo.

Uma conexão importante entre os "dois mundos" acontece no momento da captura de documentos. Aí compreendida a fase de autenticação, que valer-se-á da força conferida pela certificação digital. Uma vez, porém, assimilado ao GestãoDoc, a autenticidade passa a ser preservada com auxílio da garantia de integridade e sigilo apoiados nos controles internos do sistema. Inversamente, sempre que informações devem sair do GestãoDoc, endereçadas a agentes remotos, entra novamente em cena a criptografia assimétrica.

O quadro a seguir resume esquematicamente o exposto:

TECNOLOGIA DE CRIPTOGRAFIA	CLASSE DE ACESSO	FUNCIONALIDADE NO GESTÃODOC	PADRONIZAÇÃO
Simétrica	Interno	Armazenamento e recuperação	Interna ao GestãoDoc

TECNOLOGIA DE CRIPTOGRAFIA	CLASSE DE ACESSO	FUNCIONALIDADE NO GESTÃO DOC	PADRONIZAÇÃO
Assimétrica	Externo	Sessões Web	ICP-Brasil

RSE6.6.1	Utilizar a criptografia no armazenamento e na transmissão de documentos digitais sigilosos.	O
RSE6.6.2	Limitar o acesso aos documentos cifrados àqueles usuários portadores da chave de decifração.	O
RSE6.6.3	Registrar os seguintes metadados sobre um documento cifrado: <ul style="list-style-type: none"> <li>▪ Indicação se está cifrado ou não.</li> <li>▪ Algoritmos usados na cifração.</li> </ul>	O
RSE6.6.4	Permitir a captura de documentos cifrados.	D
RSE6.6.5	Garantir que somente o administrador seja capaz de alterar características dos mecanismos criptográficos internos. Em tais casos, deverão obrigatoriamente ser registradas, em trilha de auditoria, as seguintes informações: <ul style="list-style-type: none"> <li>▪ Descrição técnica da alteração.</li> <li>▪ Data e hora da alteração.</li> <li>▪ Identificação do executor da operação.</li> <li>▪ Motivo da alteração.</li> </ul>	O
RSE6.6.6	Nos casos de aplicação do item anterior, prover mecanismos para convivência temporária de dois sistemas de criptografia distintos. O objetivo é viabilizar a transição para o novo sistema sem indisponibilizar a operação do GestãoDoc.	O
RSE6.6.7	Impedir a abertura ( <i>disclosure</i> ) de senhas, bem como a remoção de criptografia de documentos, mesmo para o administrador. Casos de contingência, no impedimento de recuperação de informação sigilosa (por exemplo, pela morte do usuário detentor da senha) poderão ser tratados em sistemas de custódia de senhas, externos ao GestãoDoc.	O
RSE6.6.8	Possuir uma arquitetura capaz de receber atualizações tecnológicas quanto à plataforma criptográfica.	O

## 6.7 Marcas d'água digitais

Marcas d'água servem para marcar uma imagem digital com informação sobre sua proveniência e características e são utilizadas para proteger a propriedade intelectual. As marcas d'água sobrepõem, no mapa de *bits* de uma imagem, um desenho complexo, visível ou invisível, o qual só pode ser suprimido mediante a utilização de um algoritmo ou de uma chave protegida. Tecnologias semelhantes podem ser aplicadas a sons e a imagens em movimento digitalizadas.

O GestãoDoc deve manter, recuperar e assimilar novas tecnologias de marcas d'água.

REF.	REQUISITO	O BRIG.
RSE6.7.1	Recuperar informação contida em marcas d'água digitais, mediante anuência do usuário autorizado.	O
RSE6.7.2	Armazenar documentos institucionais digitais que contenham marcas d'água digitais, assim como informação de apoio relacionada à marca d'água.	O
RSE6.7.3	Possuir uma arquitetura capaz de receber atualizações tecnológicas quanto à plataforma de geração e de detecção de marca d'água digital.	O

## 6.8 Acompanhamento de transferência

Durante seu ciclo de vida, os documentos institucionais e seus respectivos metadados podem ser transferidos de uma mídia de suporte, ou de um local, para outro, à medida que sua utilização decresce e/ou se modifica. Essa transferência pode ser interna, (por exemplo, um deslocamento de armazenamento *on-line* para *off-line*), como externa, podendo trazer deslocamento para outra instituição (por exemplo, em casos de alteração de competência).

É necessário um recurso de acompanhamento, a fim de se registrar a mudança de local, tanto para facilitar o acesso como para cumprir requisitos regulamentares.

REF.	REQUISITO	O BRIG.
RSE6.8.1	Manter, para cada documento ou cada processo/dossiê, o histórico das movimentações e transferências de mídia sofridas por aquele documento ou processo/dossiê.	O
RSE6.8.2	Monitorar e registrar informações acerca do local atual e da transferência de processos/dossiês digitais e não-digitais.	O



REF.	REQUISITO	OBRIG.
RSE6.8.3	Registrar metadados que incluam: <ul style="list-style-type: none"> <li>▪ Número identificador dos documentos atribuído pelo sistema.</li> <li>▪ Localização atual e também as localizações anteriores, definidas pelo usuário.</li> <li>▪ Data e hora de envio/transferência.</li> <li>▪ Data e hora da recepção no novo local.</li> <li>▪ Destinatário.</li> <li>▪ Usuário responsável pela transferência.</li> <li>▪ Método de transferência.</li> </ul>	O

## 6.9 Autoproteção

Num ambiente digital, a autoproteção consiste na capacidade do sistema de computação de verificar a integridade de programas e de dados de controle como uma medida de proteção inicial. As técnicas de autoproteção aumentam a confiança no funcionamento correto dos programas de computador.

Esta seção trata dos requisitos relativos à capacidade do GestãoDoc de se autoprotetger contra quaisquer erros, falhas ou ataques ao próprio sistema.

Além dos requisitos de autoproteção, o GestãoDoc deverá interagir com outros sistemas de proteção, tais como: antivírus, *firewall*, *anti-spyware* etc.

REF.	REQUISITO	OBRIG.
RSE6.9.1	Negar a efetivação da captura sem a verificação de vírus ou código malicioso.	O
RSE6.9.2	Possuir dispositivos e procedimentos que reduzam as possibilidades de erros, falhas e descontinuidades no seu funcionamento que causem danos ou perdas aos documentos institucionais digitais.	D
RSE6.9.3	Entrar em modo de manutenção, no qual a possibilidade de restaurar o sistema para um estado seguro é oferecida, após falha ou descontinuidade do sistema, quando a recuperação automática não for possível.  Na restauração ao estado seguro, um GestãoDoc deve recuperar informações no maior nível tecnicamente viável.	O

REF.	REQUISITO	OBRIG.
RSE6.9.4	<p>Garantir que os dados de segurança, quando replicados, sejam consistentes.</p> <p>Permissões de controle de acesso, chaves criptográficas e parâmetros de algoritmos criptográficos são exemplos de dados de segurança.</p>	O
RSE6.9.5	<p>Preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários, quando quaisquer dos seguintes erros ocorrerem:</p> <ul style="list-style-type: none"> <li>▪ Falha de comunicação entre cliente e servidor.</li> <li>▪ Perda de integridade das informações de controle de acesso.</li> <li>▪ Impossibilidade de registro em trilha de auditoria.</li> </ul>	O
RSE6.9.6	<p>Permitir ao gestor a definição do limite de tentativas de acesso. Quando esse valor for atingido, o acesso deve ser bloqueado.</p>	O

## 6.10 Alteração, ocultação e exclusão de documentos institucionais

Os documentos institucionais completos não podem, em regra, ser alterados e excluídos, exceto no término do seu ciclo de vida num GestãoDoc. No entanto, os usuários autorizados podem precisar retificar operações, para corrigir erros de usuário (por exemplo, documentos juntados indevidamente a um processo/dossiê). Para cumprir requisitos jurídicos, usuários autorizados podem precisar ocultar informação sensível sem afetar o documento correspondente.

REF.	REQUISITO	OBRIG.
RSE6.10.1	<p>Permitir a anulação da operação em caso de erro do usuário, de acordo com as normas vigentes.</p> <p>Essa anulação tem que ser registrada nos metadados.</p> <p>Exemplos:</p> <ul style="list-style-type: none"> <li>▪ Após a distribuição do processo judicial, o ato não poderá ser anulado. A correção se dará pelo lançamento de evento baixa definitiva por erro de distribuição.</li> <li>▪ No caso de documento juntado a processo errado, a ação adequada é o desentranhamento.</li> </ul>	O

REF.	REQUISITO	OBRIG.
RSE6.10.2	Impedir a exclusão (permanente ou lógica) de documentos ou lotes de documentos fora do processo regular de eliminação.  O processo regular de eliminação é aquele previsto no programa de gestão documental.	O
RSE6.10.3	Permitir aos usuários autorizados a retificação dos metadados, com registro inclusive, na trilha de auditoria.	O
RSE6.10.4	Permitir a ocultação de dados ou partes do documento para visualização ou consulta em casos de sigilo/segredo de justiça.  As formas de ocultação devem compreender: <ul style="list-style-type: none"> <li>▪ Omissão de páginas de um documento.</li> <li>▪ Adição de retângulos opacos para ocultar nomes ou palavras sensíveis.</li> <li>▪ Outros recursos necessários para formatos de vídeo ou de áudio.</li> </ul>	D
RSE6.10.5	Quando uma cópia truncada é produzida, registrar essa ação nos metadados do documento, incluindo, pelo menos, a data, a hora, o motivo, e a pessoa que a produziu.	O

## 7 Tramitação e fluxo de trabalho

Os requisitos desta seção tratam apenas dos casos em que um GestãoDoc inclui recursos de automação de fluxo de trabalho (*workflow*). Abrangem funções para controle do fluxo de trabalho e atribuição de metadados para registro da tramitação dos documentos, incluindo o *status* do documento (minuta ou via original).

Os recursos de um GestãoDoc para controle do fluxo de trabalho podem compreender:

- Tramitação de um documento antes de seu registro/captura.
- Tramitação posterior a seu registro/captura.

As tecnologias de fluxo de trabalho gerenciam o registro de eventos e as movimentações de documentos digitais ou não-digitais sob o controle automatizado de um programa. São geralmente usadas para:

- Gestão de processos ou de tarefas, tais como registro e destinação de documentos e processos/dossiês.
- Verificação e aprovação de documentos ou processos/dossiês antes do registro.
- Encaminhamento de documentos ou processos/dossiês de forma

controlada, de um usuário para outro, com a identificação das ações a serem realizadas, tais como: verificar documento, marcar audiência, vista ao Ministério Público.

- Comunicações aos usuários internos e externos sobre a disponibilidade de um documento, a saber: as processuais (citação, intimação) e as administrativas (contatos com fornecedores, autorização para pagamentos).
- Distribuição de documentos ou processos/dossiês.
- Publicação de documentos ou processos/dossiês.

Um participante de um fluxo de trabalho pode ser um indivíduo específico, um grupo de trabalho ou mesmo um *software*. Um participante é o responsável pela realização de uma tarefa estabelecida ao longo de um fluxo de trabalho predefinido. No caso de o participante ser um indivíduo, a tarefa é direcionada a determinado usuário com identificação específica. No caso de o participante ser um grupo de trabalho, a tarefa é direcionada para o grupo (formado por vários usuários, cada um com sua identificação no sistema). A tarefa tem de ser distribuída entre os usuários do grupo, e o documento, após ser cumprido por um membro do grupo, segue o fluxo previsto. Quando o participante é um *software*, a tarefa é direcionada a uma função de programa, que a realiza automaticamente e reencaminha o documento ao fluxo previsto.

## 7.1 Controle do fluxo de trabalho

REF.	REQUISITO	OBRIG.
RTF7.1.1	Fornecer os passos necessários para o cumprimento de trâmites preestabelecidos ou fluxos alternativos. Nesse caso, cada passo significa o deslocamento de um documento ou processo/dossiê, a fim de serem objeto de ações.	O
RTF7.1.2	Possuir capacidade, sem limitações, para estabelecer o número necessário de trâmites nos fluxos de trabalho.	O
RTF7.1.3	Disponibilizar uma função para avisar a um participante do fluxo que um documento lhe foi enviado, especificando a ação necessária.	O
RTF7.1.4	Permitir a utilização de qualquer sistema de comunicação eletrônica para que um usuário possa informar outros usuários sobre documentos que requeiram sua atenção.	D
RTF7.1.5	Permitir que fluxos de trabalho pré-programados sejam definidos, alterados e mantidos exclusivamente pelo gestor.	O

REF.	REQUISITO	OBRIG.
RTF7.1.6	<p>Possibilitar que tarefas e ações sejam redistribuídas, quando necessário, em um fluxo de trabalho, a um usuário ou grupo diferente do que havia sido previsto.</p> <p>Um usuário pode precisar enviar um documento a outro usuário, em função do conteúdo daquele ou em razão de eventual afastamento do usuário responsável.</p>	O
RTF7.1.7	Registrar na trilha de auditoria todas as alterações ocorridas nesse fluxo.	O
RTF7.1.8	Registrar a tramitação de todos os documentos a fim de que os usuários possam conhecer a situação de cada um deles no processo.	O
RTF7.1.9	Efetuar a gestão dos documentos em filas de espera que possam ser examinadas e controladas pelo gestor.	O
RTF7.1.10	Permitir que os usuários visualizem a fila de espera do trabalho a eles destinado e que selecionem os itens a trabalhar.	O
RTF7.1.11	<p>Fornecer fluxos condicionais de acordo com os dados de entrada do usuário ou os dados do sistema.</p> <p>Os fluxos que remetem o documento a um dos participantes dependem de uma condição determinada por um deles. Por exemplo, um fluxo pode levar um documento a um participante ou outro, conforme os dados de entrada do participante anterior; ou a definição do fluxo pode depender de um valor calculado pelo sistema.</p>	D
RTF7.1.12	<p>Fornecer um histórico de movimentação dos documentos.</p> <p>O histórico de movimentação corresponde a um conjunto de metadados de datas de entrada e saída; nomes de responsáveis; título do documento, providências, etc.</p>	O
RTF7.1.13	<p>Permitir que usuários autorizados interrompam ou suspendam temporariamente um fluxo com o objetivo de executar outro trabalho.</p> <p>O fluxo só prosseguirá com a autorização do usuário.</p>	D
RTF7.1.14	Incluir processamento condicional, permitindo que um fluxo de trabalho seja suspenso para aguardar a chegada de um documento e prossiga, como definido pelo próprio fluxo, quando o documento é recebido.	O

REF.	REQUISITO	OBRIG.
RTF7.1.15	Associar limites de tempo a trâmites e/ou procedimentos individuais em cada fluxo e comunicar os itens que expiraram, de acordo com tais limites.	O
RTF7.1.16	Reconhecer indivíduos e grupos de trabalho como participantes.	O
RTF7.1.17	Prever a forma de distribuição dos documentos entre os membros do grupo, sempre que o participante for um grupo de trabalho.	O
RTF7.1.18	Permitir que a captura de documentos desencadeie automaticamente fluxos de trabalho.	O
RTF7.1.19	Fornecer meios de elaboração de relatórios completos para permitir que gestores monitorem a tramitação dos documentos e o desempenho dos participantes.	O
RTF7.1.20	Registrar a tramitação de um documento em seus metadados. Os metadados referentes à tramitação devem registrar, dentre outros, data e hora de envio e de recebimento e identificação dos usuários.	O
RTF7.1.21	Manter versões dos fluxos alterados e estabelecer vínculos entre os documentos já processados ou em processamento nos fluxos alterados.	O
RTF7.1.22	Assegurar que qualquer modificação nos atributos dos fluxos, como extinção ou ampliação do número de pessoas ou extinção de autorização, leve em conta os documentos vinculados.	O

## 7.2 Controle de versões e do *status* do documento

Um GestãoDoc tem de ser capaz de estabelecer — pelo seu recurso de fluxo de trabalho — o *status* do documento: minuta ou via original. No caso dos documentos digitais, esse *status* é estabelecido de acordo com o fluxo do documento no GestãoDoc. Assim, por exemplo:

- O documento é uma minuta enquanto estiver no espaço individual ou do grupo.
- Um documento transmitido do espaço individual ou do grupo, para o espaço geral, onde não poderá mais ser alterado, e daí para fora da instituição, será sempre recebido como um original.
- Um documento enviado do espaço individual ao grupo, para fins de comentários, é uma minuta, que deverá ter seu número de versões devidamente controlado.
- Quando um usuário autorizado recupera um documento do espaço geral e o armazena em seu espaço, ele cria uma nova minuta.

REF.	REQUISITO	OBRIG.
RTF7.2.1	Registrar o <i>status</i> de transmissão do documento: minuta ou via original.	<input type="radio"/>
RTF7.2.2	Controlar as diversas versões de um documento em produção.	<input type="radio"/>
RTF7.2.3	Manter o identificador único do documento e registrar, em metadados específicos, o controle de versões.	<input type="radio"/>

## 8 Avaliação e destinação

### Avaliação, temporalidade e destinação

A avaliação é uma atividade vital em um programa de gestão de documentos, pois permite racionalizar o acúmulo dos documentos nas fases corrente e intermediária, facilitando a constituição dos arquivos permanentes.

A avaliação é o processo de análise dos documentos e visa estabelecer os prazos de guarda e a destinação, de acordo com os valores primário e secundário que lhes são atribuídos. Os prazos de guarda e as ações de destinação devem estar formalizados nos instrumentos de classificação, temporalidade e destinação do programa de gestão documental da instituição.

Os prazos de guarda referem-se ao tempo necessário para o arquivamento dos documentos nas fases corrente e intermediária, visando atender exclusivamente às necessidades da administração que os gerou, com base em estimativas de utilização. Nesse sentido, nenhum documento deve ser conservado por tempo maior que o necessário.

A aplicação dos critérios de avaliação é feita com suporte na teoria das três idades e efetiva-se, primeiramente, nos arquivos correntes, a fim de se distinguirem os documentos de valor eventual (de eliminação sumária) daqueles de valor probatório e/ou informativo.

Deve-se evitar a transferência para os arquivos intermediários de documentos que não tenham sido anteriormente avaliados, pois as atividades de avaliação e seleção nesses arquivos são extremamente onerosas do ponto de vista técnico e gerencial.

A destinação dos documentos é efetivada após a atividade de seleção, que consiste na separação dos documentos de valor permanente daqueles passíveis de eliminação, mediante critérios e técnicas estabelecidos nos instrumentos de classificação, temporalidade e destinação.

A complexidade e a abrangência de conhecimentos exigidos pelo processo de avaliação, que implica o estabelecimento de critérios de valor, requerem a constituição de comissões ou grupos multidisciplinares permanentes de avaliação documental.

Um GestãoDoc deve identificar, sempre que possível, a temporalidade e a destinação prevista para o documento no momento da captura e do registro, de acordo com os prazos e as ações definidas nos instrumentos de classificação, temporalidade e destinação do

programa de gestão documental. Essa informação deve ser registrada em um metadado associado ao documento.

O sistema de gestão de documentos deve também ter capacidade de identificar aqueles que já cumpriram sua temporalidade para que se implemente a destinação prevista. No caso de um GestãoDoc, esse sistema deverá ser capaz de listar os documentos que tenham cumprido o prazo definido nos instrumentos de classificação, temporalidade e destinação.

As determinações sobre a destinação devem ser aplicadas aos documentos de forma sistemática no curso rotineiro das atividades do órgão. Essas mesmas determinações não poderão ser implementadas nos processos em tramitação, nos documentos que estejam com pendências, sob litígio ou investigação.

O sistema de gestão de documentos deve prever as seguintes ações:

- Retenção dos documentos, por um determinado período, no arquivo corrente do órgão que os gerou.
- Eliminação física.
- Transferência.
- Recolhimento à unidade de arquivo.

Um documento incluído automaticamente em um relatório de eliminação gerado por um GestãoDoc, em razão dos critérios estabelecidos, precisa ser individualmente avaliado pela Comissão de Avaliação da instituição antes de ser efetivamente eliminado. Em razão das suas peculiaridades, um documento pode ser considerado de valor permanente.

Um GestãoDoc deve ser capaz de permitir a preservação de conjuntos amostrais representativos dos documentos eliminados segundo os critérios previamente estabelecidos.

### **Eliminação**

Eliminar significa destruir os documentos que, na avaliação, foram considerados sem valor para a guarda permanente.

A eliminação deve ser precedida da elaboração de listagem, do edital de ciência de eliminação e do termo de eliminação, de acordo com a legislação vigente, e deve obedecer aos seguintes princípios:

- A eliminação deverá sempre ser autorizada pelas comissões de avaliação e pelos grupos de trabalho com base no programa de gestão documental.
- A eliminação deverá ser realizada de forma a impossibilitar a recuperação posterior de qualquer informação confidencial contida nos documentos eliminados, como, por exemplo, dados de identificação pessoal ou assinatura.
- Todas as cópias dos documentos eliminados, inclusive cópias de segurança e cópias de preservação, independentemente do suporte, deverão ser destruídas.

### **Transferência**

Transferência é a passagem de documentos do arquivo corrente para o arquivo



intermediário, sob a guarda da unidade de arquivo do órgão, onde aguardarão o cumprimento dos prazos de guarda e a destinação final.

### **Recolhimento**

Recolhimento é a passagem dos documentos do arquivo intermediário para o arquivo permanente sob a guarda da unidade de arquivo do órgão.

Em alguns casos — que devem ser especificados no Programa de Gestão de Documentos — como sentenças, acórdãos, atos normativos etc., o recolhimento do documento ocorre no ato da sua publicidade.

Os procedimentos de transferência e recolhimento de arquivos digitais para a unidade de arquivo do órgão, que implicam a transposição desses documentos de um GestãoDoc para outro sistema, deverão adotar algumas providências no que diz respeito a:

- Compatibilidade de suporte e formato, de acordo com as normas previstas pelo programa de gestão documental.
- Documentação técnica necessária para interpretar o documento digital (processamento e estrutura dos dados).
- Instrumento descritivo que inclua os metadados atribuídos aos documentos digitais e informações que possibilitem a presunção de autenticidade dos documentos recolhidos à unidade de arquivo do órgão.
- Informações sobre as migrações realizadas no órgão produtor.

Os requisitos desta seção referem-se aos procedimentos de avaliação e destinação dos documentos gerenciados pelo GestãoDoc.

## **8.1 Configuração dos instrumentos de classificação, temporalidade e destinação de documentos**

Estes requisitos referem-se à criação e manutenção dos instrumentos de classificação, temporalidade e destinação do programa de gestão documental em um GestãoDoc.

REF.	REQUISITO	OBRIG.
RAD8.1.1	<p>Prover funcionalidades para definição e manutenção de todos os instrumentos de classificação, temporalidade e destinação da política de gestão documental.</p> <p>Exemplo1:</p> <p>Manter tabela de temporalidade e destinação de documentos com as seguintes informações: identificador do órgão ou entidade, identificador da classe, prazo de guarda na fase corrente, prazo de guarda na fase intermediária, destinação final, observações, evento que determina o início da contagem do prazo de retenção na fase corrente e na fase intermediária.</p> <p>Exemplo2:</p> <p>Prever que a definição dos prazos de guarda sejam expressos por:</p> <ul style="list-style-type: none"> <li>- um número inteiro de dias ou</li> <li>- um número inteiro de meses ou</li> <li>- um número inteiro de anos ou</li> <li>- uma combinação de um número inteiro de anos, meses e dias.</li> </ul>	O
RAD8.1.2	<p>Associar automaticamente a um documento, processo/dossiê administrativo e judicial a classificação, a temporalidade e a destinação previstas nos instrumentos de gestão documental.</p>	O
RAD8.1.3	<p>Associar automaticamente a um processo judicial passível de eliminação, os critérios definidos pelo programa de gestão documental.</p>	D
RAD8.1.4	<p>Prever atividades de destinação documental como: reavaliação, eliminação, transferência e recolhimento para a unidade de arquivo.</p>	O
RAD8.1.5	<p>Prever a iniciação automática da contagem dos prazos de guarda referenciados nos instrumentos de classificação, temporalidade e destinação do programa de gestão documental a partir do último arquivamento.</p> <p>Acontecimentos específicos não-detectáveis automaticamente pelo sistema devem ser informados ao GestãoDoc por usuário autorizado como, por exemplo, “documento que poderá ser eliminado 6 anos após análise e do Tribunal de Contas da União”, conforme determina os instrumentos de avaliação documental.</p>	O
RAD8.1.6	<p>Limitar ao gestor a definição e a manutenção (alteração, inclusão e exclusão) dos instrumentos de classificação, temporalidade e destinação da política de gestão documental.</p>	O

REF.	REQUISITO	OBRIG.
RAD8.1.7	<p>Permitir que um gestor altere o prazo, destinação ou classificação prevista em algum item dos instrumentos de classificação, temporalidade e destinação da política de gestão documental e garantir que a alteração tenha efeito em todos os documentos ou processos/dossiês associados àquele item.</p> <p>As alterações nos instrumentos de classificação, temporalidade e destinação de documentos das instituições do Judiciário só poderão ser feitas após a aprovação dessas alterações pelos órgãos responsáveis pela aprovação desses instrumentos.</p>	O
RAD8.1.8	Manter o histórico das alterações realizadas nos instrumentos de classificação, temporalidade e destinação de documentos.	O
RAD8.1.9	Importar e exportar total ou parcialmente um instrumento de classificação, temporalidade e destinação de documentos.	O
RAD8.1.10	<p>Prover funcionalidades para a elaboração de relatórios que apóiem a gestão dos instrumentos de classificação, temporalidade e destinação de documentos. incluindo a capacidade de:</p> <ul style="list-style-type: none"> <li>▪ Gerar relatório completo do instrumento de classificação, temporalidade e destinação de documentos.</li> <li>▪ Gerar relatório parcial do instrumento de classificação, temporalidade e destinação de documentos a partir de um ponto determinado na hierarquia do plano de classificação.</li> <li>▪ Gerar relatório dos documentos ou processos/dossiês aos quais está atribuído um determinado prazo de guarda.</li> <li>▪ Identificar as inconsistências existentes entre os instrumentos de classificação, temporalidade e destinação de documentos e o plano de classificação.</li> </ul>	O

## 8.2 Aplicação dos instrumentos de classificação, temporalidade e destinação de documentos

Estes requisitos referem-se à aplicação dos instrumentos de classificação, temporalidade e destinação de documentos: procedimentos de controle e verificação dos prazos e da destinação prevista, antes de se proceder às ações de destinação propriamente ditas.

REF.	REQUISITO	OBRIG.
------	-----------	--------

REF.	REQUISITO	OBRIG.
RAD8.2.1	Prover funcionalidades para informar o usuário autorizado sobre os documentos ou processos/dossiês que já cumpriram ou estão para cumprir o prazo de guarda previsto.	O
RAD8.2.2	Pedir confirmação antes de realizar os procedimentos de destinação (emissão de relatórios, editais, etc.), conforme normas de gestão documental estabelecidas.	O
RAD8.2.3	Restringir as funcionalidades de destinação a usuários autorizados da unidade de arquivo.	O
RAD8.2.4	Quando um usuário autorizado reclassifica documentos ou processos/dossiês de uma classe ou assunto para outro, adotar automaticamente a temporalidade e a destinação vigentes na nova classe ou assunto.	O
RAD8.2.5	Os documentos previamente definidos pelo Programa de Gestão Documental como de guarda permanente (sentenças, inteiro teor de acórdão, etc.) deverão ter sua guarda garantida.	O
RAD8.2.6	<p>Quando um documento digital — não previamente definido pelo Programa de Gestão Documental como de guarda permanente — estiver associado a mais de um dossiê ou processo, e tiver prazos de guarda diferentes associados a ele, esse prazo deverá ser o mais abrangente.</p> <p>Quando um documento digital estiver associado a mais de um dossiê ou processo, o GestãoDoc deverá criar um registro para cada referência desse documento. Cada registro estará vinculado ao mesmo objeto digital.</p> <p>O objeto digital não poderá ser eliminado sem antes ter ocorrido a verificação da temporalidade de todas as referências associadas a ele. O objeto digital só poderá ser eliminado quando os prazos de guarda de todas as referências tiverem sido cumpridos. Antes disso, só se pode fazer a eliminação de cada registro individualmente.</p>	O

### 8.3 Exportação de documentos

Um GestãoDoc deve ter capacidade de exportar documentos, processos/dossiês para apoiar a ação de transferência de documentos, ou ainda para realizar uma migração ou enviar uma cópia para outro local ou sistema.

É absolutamente necessário que as ações sejam executadas de maneira controlada, havendo registro nos metadados e na trilha de auditoria e verificação dos documentos relacionados.

<b>REF.</b>	<b>REQUISITO</b>	<b>OBRIG.</b>
RAD8.3.1	Exportar documentos e processos/dossiês digitais e seus metadados para outro sistema, dentro ou fora do órgão.	O
RAD8.3.2	Exportar um documento e processo/dossiê ou grupo de documentos e processos/dossiês numa seqüência de operações, de modo que: <ul style="list-style-type: none"> <li>▪ O conteúdo, o contexto e a estrutura dos seus documentos não se degradem.</li> <li>▪ Todos os componentes de um documento digital sejam exportados como uma unidade.</li> <li>▪ Todos os metadados do documento sejam relacionados a ele, de forma que os vínculos sejam mantidos no novo sistema.</li> <li>▪ Todas as ligações entre documentos, volumes e processos/dossiês sejam mantidas.</li> </ul>	O
RAD8.3.3	Exportar processos/dossiês: <ul style="list-style-type: none"> <li>▪ Em seu formato nativo ou no formato para o qual foi migrado.</li> <li>▪ De acordo com os formatos definidos em padrões de interoperabilidade adotados pelo Judiciário brasileiro.</li> </ul>	O
RAD8.3.4	Exportar todos os tipos de documentos que estiver apto a capturar.	D
RAD8.3.5	Produzir um relatório detalhado sobre qualquer falha que ocorra durante uma exportação. O relatório tem de identificar os documentos e processos/dossiês que tenham originado erros de processamento ou cuja exportação não tenha sido bem sucedida.	O
RAD8.3.6	Conservar todos os documentos e processos/dossiês digitais que tiverem sido exportados, pelo menos até que tenham sido importados no sistema destinatário com êxito.	O
RAD8.3.7	Manter metadados relativos a documentos e processos/dossiês exportados.	O
RAD8.3.8	Possibilitar a ordenação dos documentos e processos/dossiês digitais a serem exportados de acordo com elementos de metadados selecionados pelo usuário.	D

REF.	REQUISITO	OBRIG.
RAD8.3.9	Exigir do usuário autorizado, ao exportar documentos e processos/dossiês híbridos, a confirmação de que a parte não-digital dos mesmos documentos e processos/dossiês tenha sido recebida adequadamente antes de confirmar a exportação da parte sob forma digital.	<input type="radio"/>
RAD8.3.10	Permitir que documentos sejam exportados mais de uma vez. Exemplo: O magistrado determina a remessa de processo eletrônico para o Superior Tribunal de Justiça (STJ) e também para o Supremo Tribunal Federal (STF).	<input type="radio"/>

## 8.4 Eliminação

A eliminação de documentos institucionais deve ser realizada de acordo com o previsto nos instrumentos de classificação, temporalidade e destinação de documentos, após a avaliação dos documentos e de acordo com a regulamentação do programa de gestão documental.

Os procedimentos para eliminação de documentos institucionais em um GestãoDoc têm de ser executados de forma controlada, fazendo-se registro nos metadados e trilha de auditoria.

REF.	REQUISITO	OBRIG.
RAD8.4.1	Restringir a função de eliminação de documentos ou processos/dossiês a usuários autorizados.	<input type="radio"/>
RAD8.4.2	Solicitar confirmação da eliminação a um usuário autorizado antes que qualquer ação seja tomada com relação ao documento e processo/dossiê.	<input type="radio"/>
RAD8.4.3	<ul style="list-style-type: none"> <li>▪ Avisar o usuário autorizado quando um documento ou processo/dossiê passível de eliminação encontrar-se relacionado a outro documento ou processo/dossiê.</li> </ul>	<input type="radio"/>
RAD8.4.4	Permitir a eliminação de documentos ou processos/dossiês de forma irreversível, a fim de que não possam ser restaurados por meio da utilização normal do GestãoDoc.	<input type="radio"/>
RAD8.4.5	Quando um documento tiver várias referências armazenadas no sistema, garantir que todas essas referências sejam verificadas antes de eliminar o objeto digital. Ver RAD8.2.6	<input type="radio"/>

REF.	REQUISITO	OBRIG.
RAD8.4.6	Produzir um relatório detalhando qualquer falha que ocorra durante uma eliminação. O relatório deverá identificar os documentos cuja eliminação não tenha sido bem sucedida.	O
RAD8.4.7	Quando eliminar documentos ou processos/dossiês híbridos, exigir do usuário autorizado a confirmação de que a parte não-digital dos mesmos tenha sido eliminada também antes de confirmar a eliminação da parte sob forma digital.	O
RAD8.4.8	Gerar relatório com os documentos e processos/dossiês: <ul style="list-style-type: none"> <li>▪ Passíveis de eliminação (editais).</li> <li>▪ Selecionados para guarda permanente pela aplicação do plano amostral.</li> <li>▪ Definitivamente eliminados (termo de eliminação).</li> </ul>	O
RAD8.4.9	Manter metadados relativos a documentos e processos/dossiês eliminados.	O

## 8.5 Avaliação e destinação de documentos institucionais não-digitais e híbridos

Os documentos institucionais não-digitais e os híbridos gerenciados pelo GestãoDoc devem ter os procedimentos de avaliação e destinação controlados pelo GestãoDoc, assim como os documentos digitais.

REF.	REQUISITO	OBRIG.
RAD8.5.1	Aplicar os mesmos instrumentos de classificação, temporalidade e destinação de documentos para os documentos não-digitais, digitais ou híbridos.	O
RAD8.5.2	Alertar ao usuário autorizado sobre a existência e localização de uma parte não-digital associada a um documento híbrido que esteja destinado a ser exportado, transferido ou eliminado.	O
RAD8.5.3	Permitir a exportação de metadados de documentos e processos/dossiês não-digitais.	O
RAD8.5.4	Permitir ao usuário autorizado selecionar outra destinação final aos processos/dossiês e documentos diferente daquela originalmente existente da tabela de temporalidade, somente quando a nova destinação for a guarda permanente.  Estes casos são válidos para realização de amostragem antecedente à eliminação e alienação.	O

## 9 Pesquisa, localização e apresentação de documentos

Um sistema de gestão de documentos deve prever funções de recuperação e acesso aos documentos institucionais e às informações neles contidas, de forma a satisfazer a condução das atividades e os requisitos relativos à transparência do órgão. A recuperação inclui a pesquisa, a localização e a apresentação dos documentos.

Em um GestãoDoc a apresentação dos documentos consiste em exibí-los por meio de um ou mais dispositivos de apresentação, tais como monitor de vídeo, impressora, caixa de som, etc. No âmbito do sistema de gestão de documentos, a pesquisa é feita a partir de bases de dados e produtos decorrentes. Já em um GestãoDoc a pesquisa é flexibilizada por parâmetros predefinidos, selecionados dentre as informações coletadas no momento do registro do documento e dentre os metadados a ele associados.

Todos os recursos de pesquisa, localização e apresentação de documentos têm de ser submetidos aos controles de acesso e segurança descritos na seção específica.

### 9.1 Recuperação de informação

REF.	REQUISITO	OBRIG.
RPL9.1.1	Fornecer facilidades para pesquisa, localização e apresentação dos documentos.	O
RPL9.1.2	Fornecer interface de pesquisa, localização e apresentação opcionais via um ambiente <i>Web</i> .	D
RPL9.1.3	Prever a navegação gráfica do plano de classificação, a navegação direta de uma classe ou assunto, para os documentos institucionais criados nessa classe ou assunto, e a seleção, recuperação e apresentação direta dos documentos institucionais e de seus conteúdos por meio desse mecanismo.	D
RPL9.1.4	Restringir a recuperação de informações de documentos e processos/dossiês sigilosos e em segredo de justiça aos usuários que possuam credencial de segurança adequada.	O

### 9.2 Pesquisa e localização

A pesquisa é o processo de identificação de documentos institucionais por meio de parâmetros definidos pelo usuário com o objetivo de confirmar, localizar e recuperar esses documentos, bem como seus respectivos metadados.



<b>REF.</b>	<b>REQUISITO</b>	<b>OBRIG.</b>
RPL9.2.1	Fornecer uma série de funções que atuem sobre os metadados relacionados com os diversos níveis de agregação (documento, unidade de arquivamento e classe) e sobre os conteúdos dos documentos institucionais, por meio de parâmetros definidos pelo usuário, com o objetivo de localizar e acessar os documentos e/ou metadados, quer individualmente quer reunidos em grupo.	O
RPL9.2.2	Permitir a pesquisa regional de forma integrada, apresentando todos os documentos e processos/dossiês, sejam eles digitais, híbridos ou não-digitais, que satisfaçam aos parâmetros da pesquisa.	O
RPL9.2.3	Permitir a pesquisa nacional de forma integrada, apresentando todos os documentos e processos/dossiês, sejam eles digitais, híbridos ou não-digitais, que satisfaçam aos parâmetros da pesquisa.	D
RPL9.2.4	Permitir que todos os metadados de gestão de um documento ou processo/dossiê sejam pesquisados. O usuário deve ser informado quando a pesquisa não obtiver resultado.	O
RPL9.2.5	Permitir que os conteúdos sob a forma de texto dos documentos sejam pesquisados. O usuário deve ser informado quando a pesquisa não obtiver resultado.	D
RPL9.2.6	Permitir que um documento ou processo/dossiê seja recuperado por meio de todas as formas de identificação implementadas, incluindo no mínimo: <ul style="list-style-type: none"> <li>▪ Identificador</li> <li>▪ Título ou descrição abreviada</li> <li>▪ Datas</li> <li>▪ Unidade de origem/destino</li> <li>▪ Signatário/redator/parte/advogado/magistrado/interessado</li> <li>▪ Classificação de acordo com os instrumentos de classificação</li> </ul>	O
RPL9.2.7	Fornecer uma interface que possibilite a pesquisa combinada de metadados e de conteúdo do documento por meio dos operadores lógicos: “E”, “OU” e “NÃO”.	D

REF.	REQUISITO	OBRIG.
RPL9.2.8	Permitir que os termos utilizados na pesquisa sejam qualificados, especificando-se um metadado ou o conteúdo do documento como fonte de busca.	D
RPL9.2.9	Permitir a utilização de caracteres coringa e busca fonética para a pesquisa de metadados.	D
RPL9.2.10	Permitir que os usuários possam refinar as pesquisas já realizadas.	D
RPL9.2.11	Quando o órgão utilizar tesouros ou vocabulário controlado, realizar pesquisa dos documentos e processos/dossiês por meio da navegação destes instrumentos.	D
RPL9.2.12	Permitir a pesquisa previamente parametrizada, de acordo com o perfil ou necessidade do usuário.	O
RPL9.2.13	Permitir a pesquisa e recuperação de uma unidade de arquivamento e exibir a lista de todos os documentos que o compõem.	O

### 9.3 Apresentação: texto, imagem, som e vídeo

Um GestãoDoc pode conter documentos institucionais com formatos e estruturas os mais diversos e deve ter capacidade para apresentá-los ao usuário sem adulterá-los, utilizando-se adequadamente de suportes tecnológicos para texto, imagem, som e vídeo.

O sistema deverá informar os programas (*software*) adicionais necessários e a configuração adequada, como, por exemplo, *plug-in*, configuração de navegador.

REF.	REQUISITO	OBRIG.
RPL9.3.1	Apresentar o resultado da pesquisa como uma lista de documentos e processos/dossiês digitais, não-digitais ou híbridos que cumpram os parâmetros daquela.	O
RPL9.3.2	Após apresentar o resultado da pesquisa, além de informar a quantidade de itens recuperados, deve-se permitir ao usuário as seguintes opções: <ul style="list-style-type: none"> <li>▪ Acessar os documentos e processos/dossiês resultantes da pesquisa.</li> <li>▪ Redefinir os parâmetros de pesquisa e fazer nova consulta.</li> </ul>	O

REF.	REQUISITO	OBRIG.
RPL9.3.3	Permitir que os documentos e processos/dossiês apresentados em uma lista de resultados sejam selecionados e, em seguida, abertos.	D
RPL9.3.4	Permitir ao gestor a configuração de pesquisas, possibilitando as seguintes parametrizações: <ul style="list-style-type: none"> <li>▪ Determinação do número máximo de itens recuperáveis em uma pesquisa.</li> <li>▪ Definição dos metadados que devem ser exibidos nas listas de resultados de pesquisa.</li> </ul>	D
RPL9.3.5	Permitir a configuração do formato da lista de resultados de pesquisa pelo usuário incluindo as funcionalidades: <ul style="list-style-type: none"> <li>▪ Seleção da ordem em que os resultados de pesquisa são apresentados.</li> <li>▪ Determinação do número de resultados de pesquisa exibidos na tela de cada vez.</li> <li>▪ Armazenamento dos resultados de uma pesquisa.</li> </ul>	D
RPL9.3.6	Fornecer recursos que permitam a um usuário “navegar” para o nível imediatamente superior ou inferior, como, por exemplo: <ul style="list-style-type: none"> <li>▪ De um documento para a unidade de arquivamento em que está incluído.</li> <li>▪ De uma unidade de arquivamento para os documentos nela incluídos.</li> <li>▪ De uma unidade de arquivamento para a classe respectiva.</li> <li>▪ De uma classe para as unidades de arquivamento a ela relacionadas.</li> </ul>	D
RPL9.3.7	Apresentar o conteúdo de todos os tipos de documentos institucionais digitais capturados, preservando as características e os formatos.	O
RPL9.3.8	Reproduzir os documentos capturados, preservando o formato produzido pelas aplicações geradoras.	O
RPL9.3.9	Permitir que todos os documentos de um processo/dossiê sejam impressos ou armazenados em uma única operação, na seqüência determinada pelo usuário.	O

REF.	REQUISITO	OBRIG.
RPL9.3.10	Apresentar os documentos institucionais em formatos padronizados para publicação digital e interoperabilidade, além do formato nativo.  No que se refere à interoperabilidade com outros sistemas, ver Capítulo 12, Interoperabilidade.	O
RPL9.3.11	Realizar pesquisa e exibição de documentos e processos/dossiês simultaneamente para diversos usuários.	O
RPL9.3.13	Permitir ao gestor o estabelecimento de permissões para armazenamento e reprodução de documentos, processos/dossiês.	O

## 10 Funções administrativas

### 10.1 Monitoração do sistema

REF.	REQUISITO	OBRIG.
RFA10.1.1	Permitir que o gestor, de maneira controlada e sem esforço excessivo, recupere, identifique, visualize e reconfigure os parâmetros do sistema e os atributos dos perfis dos usuários.	O
RFA10.1.2	Fornecer ao gestor relatórios flexíveis para o gerenciamento dos volumes e itens e sua utilização, que apresentem no mínimo: <ul style="list-style-type: none"> <li>▪ Quantidade de processos/dossiês, volumes e itens a partir de parâmetros ou atributos definidos (tempo, classe, unidade administrativa etc.).</li> <li>▪ Estatísticas de operações relativas a processos/dossiês, volumes e itens.</li> <li>▪ Relatórios de operações por usuário.</li> </ul>	O
RFA10.1.3	Prover documentação cobrindo aspectos de administração do sistema. A documentação deve incluir todas as informações necessárias para o correto gerenciamento do sistema.	O

## 10.2 Manutenção e evolução

REF.	REQUISITO	OBRIG.
RFA10.2.1	Possuir documentação de implementação.	O
RFA10.2.2	Ser aderente à normatização do CNJ nos aspectos de processo de desenvolvimento de <i>software</i> .	O
RFA10.2.3	Possuir um ambiente de homologação para avaliação de novas versões de <i>software</i> , que permita testes: <ul style="list-style-type: none"> <li>▪ Funcionais</li> <li>▪ De preservação da integridade do acervo digital</li> </ul>	O

## 11 Usabilidade

O projeto de um sistema de *software* com boa usabilidade<sup>6</sup> exige preocupação com a facilidade de utilização. A objetividade de apresentação de informações pelo sistema deve possibilitar a realização segura e eficiente das tarefas, ao mesmo tempo em que oferece interação agradável com o usuário.

Boa parte do sucesso de um sistema depende de sua aceitação. Um sistema desenvolvido objetivando facilidade de utilização tem mais chance de satisfazer os usuários. Além disso, exigirá menores custos de manutenção, treinamento e suporte.

REF.	REQUISITO	OBRIG.
RUS11.1.1	Possuir documentação completa, clara, inteligível e organizada para utilização do <i>software</i> .	O
RUS11.1.2	Possuir sistema de ajuda on-line.	O
RUS11.1.3	Vincular o sistema de ajuda <i>on-line</i> à função ou tarefa executada (sensível ao contexto). Exemplo: Quando se executa uma operação de edição, uma vez acionada a ajuda, ela deve remeter para o tópico de ajuda da edição.	D

<sup>6</sup> O conceito de usabilidade é tratado pela norma ISO/IEC 9126:1991 Information technology — *Software product evaluation: quality characteristics and guidelines for their use*.

REF.	REQUISITO	OBRIG.
RUS11.1.4	<p>Permitir ao gestor a personalização de conteúdo de ajuda <i>on-line</i> por adição de texto ou edição do texto existente.</p> <p>Exemplo: O responsável pela gestão do conteúdo da ajuda pode adicionar esclarecimentos ou alterar o conteúdo das descrições, de modo a facilitar o entendimento das funções.</p>	D
RUS11.1.5	Toda mensagem de erro produzida deve ser clara e significativa, de modo a permitir ao usuário corrigir ou cancelar a operação.	O
RUS11.1.6	<p>A interface deve seguir padrões preestabelecidos e consolidados como boas práticas de projeto gráfico, validados cientificamente.</p> <p>Normas ou regras de interface podem ser relativas à utilização de padrão de identidade visual (ligado à “marca” da instituição ou alguma legislação específica do Judiciário brasileiro), bem como a utilização de guias de estilo para implementação e verificação da padronização da interface, desde que não interfira nos princípios básicos da ergonomia cognitiva.</p>	O
RUS11.1.7	<p>Utilização de um conjunto simples e consistente de regras de interface, privilegiando a facilidade de aprendizado de operação pelos seus usuários.</p> <p>A utilização de um conjunto de regras consistentes com o ambiente operacional em que o GestãoDoc será executado permite que ele apresente menus, comandos e outras facilidades consistentes em toda aplicação.</p> <p>Essas regras de interface, quando compatíveis com outras aplicações principais já instaladas, levam à padronização da terminologia utilizada para funções, rótulos e ações consistentes em toda a aplicação.</p>	D
RUS11.1.8	A interface de visualização dos documentos institucionais deve fornecer o recurso de arrastar e soltar, se apropriado no ambiente operacional do GestãoDoc.	D
RUS11.1.9	Permitir que a estrutura de classes, assuntos e processos/dossiês seja visualizada em diferentes formas de apresentação.	D
RUS11.1.10	<p>Personalizar a interface gráfica, quanto aos seguintes aspectos:</p> <ul style="list-style-type: none"> <li>▪ Conteúdos de menus.</li> <li>▪ Formatos de telas.</li> <li>▪ Utilização de teclas de função.</li> <li>▪ Alteração de cores, fontes e tamanhos de fontes em telas e janelas dentro de parâmetros ergonômicos.</li> <li>▪ Avisos sonoros, incluindo tom e volume.</li> </ul>	D

REF.	REQUISITO	OBRIG.
RUS11.1.11	Utilizar barras de ferramentas, permitindo ao usuário a possibilidade de configuração e de habilitar/desabilitar esse tipo de recurso. Porém, de forma a não infringir a recomendação de utilização de um conjunto simples e consistente de regras de interface.	D
RUS11.1.12	Permitir a utilização de janelas, sua movimentação, redimensionamento e gravação das modificações da aparência, possibilitando a personalização por perfil de usuário dentro de parâmetros ergonômicos.	D
RUS11.1.13	Permitir a gravação de opções <i>default</i> para entrada de dados de configuração: <ul style="list-style-type: none"> <li>▪ Valores iguais aos de um item anterior.</li> <li>▪ Valores que possam ser selecionados de uma lista configurável.</li> <li>▪ Valores derivados do contexto, como data, referência do processo/dossiê, identificador do usuário.</li> <li>▪ Valores predefinidos por um administrador (para campos de metadados como, por exemplo, o nome da organização que está utilizando o sistema).</li> </ul>	D
RUS11.1.14	A interface tem de possibilitar a utilização às pessoas portadoras de necessidades especiais, de modo a atender o Decreto 5.296, de 2004.	O
RUS11.1.15	A interação deve permitir a interface com leitores de telas para portadores de deficiências visuais.	O
RUS11.1.16	A utilização não tem de tornar obrigatório o uso de aparelho selecionador específico ( <i>mouse</i> , por exemplo).	O
RUS11.1.17	Permitir a realização de transações ou tarefas mais freqüentemente executadas com um pequeno número de iterações (cliques de <i>mouse</i> , por exemplo) e sem mudanças excessivas de contexto.	D
RUS11.1.18	Integração com o sistema de comunicação eletrônica da organização, de forma a permitir a geração de mensagens com possibilidade de manipular documentos digitais, sem necessidade de sair do GestãoDoc.	D
RUS11.1.19	No caso de integração com o sistema de comunicação eletrônica, deve ser possível fazer referências a documentos institucionais sem necessidade de envio de cópias adicionais.	D

REF.	REQUISITO	OBRIG.
RUS11.1.20	Possuir integração com o sistema padrão de edição de documentos.	D
RUS11.1.21	Permitir a definição e utilização de referências cruzadas entre documentos institucionais digitais correlacionados, possibilitando uma fácil navegação entre eles, inclusive com uso de <i>hyperlinks</i> .	D
RUS11.1.22	Restringir o acesso às funcionalidades administrativas impossibilitando sua visualização ao usuário.	O



## 12 Interoperabilidade

A adoção de regras e padrões de comunicação já consolidados permite a consulta entre sistemas heterogêneos, sem que o usuário perceba as operações envolvidas, convergindo para uma relação sinérgica entre as partes.

Esta seção estabelece requisitos mínimos para que um GestãoDoc possa interoperar com outros sistemas de informação, inclusive sistemas legados, respeitando normas de segurança de acordo com padrões abertos de interoperabilidade.

Por interoperabilidade, entende-se: “Intercâmbio coerente de informações e serviços entre sistemas. A interoperabilidade deve possibilitar a substituição de qualquer componente ou produto usado nos pontos de interligação por outro de especificação similar, sem comprometimento das funcionalidades do sistema”<sup>7</sup>. Isto se faz mediante a utilização de regras e padrões de comunicação.

O GestãoDoc deverá adotar o padrão de interoperabilidade regulamentado pelo CNJ.

REF.	REQUISITO	OBRIG.
RIN12.1.1	Interoperar com outros sistemas, permitindo pelo menos consulta, recuperação, importação e exportação de documentos e seus metadados.  As operações de interoperabilidade devem respeitar a legislação vigente e a política de segurança da informação das instituições do Judiciário.	O
RIN12.1.2	Interoperar com outros sistemas por intermédio de padrões abertos de interoperabilidade que deverão ser regulamentados pelo CNJ.	O
RIN12.1.3	Aplicar os requisitos de segurança descritos neste documento para execução das operações de interoperabilidade.  Isso é fundamental para que as operações, feitas em ambiente com interoperabilidade, não afetem a integridade dos documentos e impossibilitem acessos não autorizados.	O

<sup>7</sup> <<https://www.governoeletronico.gov.br/acoes-e-projetos/e-ping-padres-de-interoperabilidade>>. Acesso em: 20 jul 2007.

## 13 Disponibilidade

Requisitos de disponibilidade descrevem as exigências mínimas sobre prontidão de atendimento de um sistema.

Os requisitos de disponibilidade devem ser especificados pelo administrador do GestãoDoc de acordo com o nível de serviço a ser fornecido. Por exemplo, os períodos previstos de atendimento (“8x5” indica 8 horas por dia útil, “24x7” indica atendimento contínuo), bem como o tempo máximo tolerável em interrupções previstas. O grau de disponibilidade a ser estabelecido deve levar em conta fatores como as regras de negócio da organização, a necessidade de realização de *backup*, manutenções planejadas, entre outros.

REF.	REQUISITO	OBRIG.
RDI13.1.1	Garantir disponibilidade de operação durante o período definido pela instituição, ressalvadas as falhas causadas por problemas de infra-estrutura.	O

## 14 Desempenho e escalabilidade

Os requisitos de desempenho enfocam a eficiência no atendimento às requisições de usuários, de acordo com suas expectativas quanto aos tempos de resposta. O tempo de resposta tanto é influenciado por requisitos de qualidade do *software* quanto por fatores externos, como, por exemplo, infra-estrutura de rede, volume de tráfego de dados e dimensionamento dos servidores e das estações de trabalho. O desempenho é medido avaliando-se a velocidade de processamento, o tempo de resposta e o consumo de recursos.

A escalabilidade de um componente ou de um *software* relaciona-se à capacidade do sistema manter o mesmo desempenho — tempo de resposta — quando há um aumento no número de usuários e/ou de requisições simultâneas.

Sobre desempenho e escalabilidade, investimentos em *hardware* devem refletir no aumento de desempenho do sistema. Quando se acrescentam mais máquinas, os investimentos em *hardware* caracterizam a escalabilidade horizontal. Quando se aumenta o poder de processamento das máquinas existentes, a escalabilidade é vertical. Melhor escalabilidade possibilita distribuir e configurar a execução da aplicação para satisfazer vários volumes de transação. Um sistema é dito escalável quando o investimento necessário à melhoria do desempenho é proporcional ao resultado obtido.

A organização deve manter indicadores do valor da sua infra-estrutura de informação olhando para a relação entre o capital investido e os níveis de performance obtidos.

Para um GestãoDoc, entende-se escalabilidade como a capacidade do sistema responder a um aumento de usuários e volume de documentos processados, mantendo-se o desempenho das respostas do sistema.

REF.	REQUISITO	OBRIG.
RDE14.1.1	Manter estatísticas dos tempos de atendimento, discriminados por tipo de operação.	D
RDE14.1.2	Ser expansível até comportar um número máximo preestabelecido de usuários simultâneos, provendo continuidade efetiva de serviços.	O
RDE14.1.3	Manter registros de atualização de versão de infra-estrutura e do próprio sistema.	O
RDE14.1.4	Permitir adaptação a instituições de estruturas similares, mas de diferentes tamanhos.	D

REF.	REQUISITO	OBRIG.
RDE14.1.5	<p>Fornecer evidências do grau de escalabilidade ao longo do D tempo, mantendo avaliações quantitativas de:</p> <ul style="list-style-type: none"> <li>▪ Número máximo de <i>sites</i> remotos suportados com desempenho adequado.</li> <li>▪ Tamanho máximo do repositório.</li> <li>▪ Número máximo de usuários simultâneos que possam ser atendidos com desempenho adequado.</li> <li>▪ Sobrecarga administrativa, expectativa de crescimento do número de usuários.</li> <li>▪ Expectativa de crescimento das bases de dados.</li> <li>▪ Expectativa de crescimento do número de estações.</li> </ul>	D

## 15 Glossário

Administrador	Responsável por manter o ambiente operacional do sistema.
Anexação	Ato de reunir documentos organizados em volumes próprios a um determinado processo. Os documentos que formam os anexos tramitam junto ao processo, mas não são atuados como um processo.
Anexo	<p>Documentos organizados em volume próprio, que acompanham um processo, mas não são atuados como um processo.</p> <p>Documentos que acompanham e estão vinculados a um documento principal ou mensagem, independentemente do suporte em que se apresentam.</p>
Apensação	<p>Reunião de dois ou mais processos, permanecendo cada processo com seu respectivo número.</p> <p>Nos processos judiciais, a apensação ocorre por determinação legal ou judicial em processos que estejam em movimento, suspensos ou baixados. Nos processos administrativos, a apensação ocorre por determinação da autoridade competente.</p>

Avaliação de documentos	Processo de análise de documentos, que estabelece os prazos de guarda e a destinação, de acordo com os valores que lhes são atribuídos.
Autuação	Formar autos. Reunir em forma de processo (a petição e documentos apresentados em juízo), designando número, identificando partes, procuradores, assunto, classe processual e outras informações relevantes.
Captura	Incorporação de documento ao sistema.
Checksum	Valor calculado a partir dos dados, para comprovação de integridade.
Ciclo de vida dos documentos	As sucessivas etapas pelas quais os documentos passam: produção, tramitação, uso, avaliação, arquivamento e destinação (guarda permanente ou eliminação).
Classe	Primeira divisão de um plano de classificação ou de um código de classificação.
Código de classificação	Conjunto de símbolos, normalmente letras e/ou números, derivado de um plano de classificação.
Conversão	Técnica de migração que pode se configurar de diversas formas, tais como: 1. conversão de dados: mudança de um formato para outro. 2. conversão de sistema computacional: mudança do modelo de computador e de seus periféricos.
Desapensação	Separação de processos que estavam apensados. Nos processos judiciais, via de regra, é o efeito de uma decisão judicial que determina a separação de processos que estavam reunidos. No caso dos processos administrativos, a desapensação ocorre por determinação da autoridade competente.
Desentranhamento	Ato de retirar peças juntadas em processo judicial ou administrativo.
Desmembramento	Ato de dividir um processo em dois ou mais processos. Ocorre, nos processos judiciais, por decisão judicial e nos administrativos por determinação da autoridade competente.
Documento institucional	Documento produzido e/ou recebido por um órgão

(arquivístico)	do Poder Judiciário, no decorrer das suas atividades, qualquer que seja o suporte, e dotado de organicidade.
Documento institucional digital	Documento codificado em dígitos binários, acessível por meio de sistema computacional.
Dossiê	Conjunto de documentos relacionados entre si por ação, evento, pessoa, lugar, projeto, que constitui uma unidade.
Emulação	Utilização de recursos computacionais que fazem uma tecnologia funcionar com as características de outra, aceitando as mesmas entradas e produzindo as mesmas saídas.
Ergonomia cognitiva	Processos mentais, tais como percepção, memória, raciocínio e resposta motora conforme afetem as interações entre seres humanos e outros elementos de um sistema. Os tópicos relevantes incluem o estudo da carga mental de trabalho, tomada de decisão, desempenho especializado, interação homem computador, <i>stress</i> e treinamento conforme esses se relacionem a projetos envolvendo seres humanos e sistemas.
Fluxo de trabalho	Automatização de uma atividade, no todo ou em parte, durante a qual documentos, informação ou tarefas transitam de um participante para outro com vistas a serem submetidos a ações, de acordo com um conjunto de normas processuais.
Gestão de documentos	Conjunto de procedimentos e operações técnicas que engloba a produção, a tramitação, a utilização, a avaliação e o arquivamento em fase corrente e intermediária, visando a sua eliminação ou recolhimento para guarda permanente. (Lei nº 8.159, de 1991, art. 3º)
Gestor	Responsável pelo gerenciamento das regras de negócio do sistema. Cabe ao gestor, entre outras atividades, a criação de usuários e atribuição de privilégios.
Memória primária	De funcionamento essencial, é necessária a qualquer sistema computacional. É nela que o <i>software</i> e os dados são armazenados durante a execução. Representantes típicas dessa classe

	<p>são as memórias <i>Random Access Memory</i> (RAM). São memórias extremamente rápidas, de conteúdo dinâmico e volátil, permanecendo registrado apenas durante a execução do <i>software</i>.</p>
Memória secundária	<p>Apresenta volume maior de armazenamento que a primária; entretanto é mais lenta e não-volátil. São exemplos os discos rígidos magnéticos (<i>hard disk</i> — HD), que podem ser usados isolados ou combinados em <i>disk arrays</i>. Diversas tecnologias permitem, através do uso de <i>disk arrays</i>, obter-se maior desempenho e confiabilidade do que com os discos isoladamente.</p>
Memória terciária	<p>Compreende fitas magnéticas, discos ópticos e outros. Usos típicos incluem armazenamento do acervo digital e cópias de segurança. Outra nomenclatura corrente para essa classe de memória é "mídias de armazenamento". A memória terciária tem característica não-volátil na preservação de dados. Seu preço unitário é tão pequeno que requisitos de confiabilidade devem prevalecer: em caso de desastre, o prejuízo da perda de dados é superior ao preço das mídias que fisicamente os contêm.</p>
Metadado	<p>Informação que descreve e contextualiza o dado.</p> <p>Dado estruturado que descreve e permite encontrar, gerenciar, compreender e/ou preservar documentos arquivísticos ao longo do tempo.</p>
Minuta	<p>Versão preliminar de documento sujeita à aprovação.</p>
Plano de classificação	<p>Esquema de distribuição de documentos em classes, de acordo com métodos de arquivamento específicos, elaborado a partir do estudo das estruturas e funções de uma instituição e da análise do arquivo por ela produzido. Expressão geralmente adotada em arquivos correntes.</p>
Processo	<p>Conjunto de documentos oficialmente reunidos no decurso de uma ação administrativa ou judicial, que constitui uma unidade.</p>
Rejuvenescimento	<p>Técnica de migração que consiste em copiar os dados de um suporte para outro sem mudar sua codificação, para evitar perdas de dados</p>

Recolhimento	Entrada de documentos em arquivos permanentes.
Teoria das três idades	Base do conceito de gestão de documentos, essa teoria os classifica em três fases: <ul style="list-style-type: none"><li>▪ <b>Corrente</b> — Documentos que estão em curso (tramitando ou arquivados), mas objeto de consultas freqüentes. São conservados nos locais onde foram produzidos sob a responsabilidade do órgão produtor.</li><li>▪ <b>Intermediária</b> — Documentos que não são mais de uso corrente, mas que por conservarem ainda algum interesse administrativo, aguardam no arquivo intermediário o cumprimento do prazo estabelecido nos instrumentos de classificação, temporalidade e destinação, para serem eliminados ou recolhidos ao arquivo permanente.</li><li>▪ <b>Permanente</b> — Documentos que devem ser definitivamente preservados em função de seu valor histórico, probatório ou informativo.</li></ul>
Tramitação	Curso do documento desde sua produção ou recepção até o cumprimento de sua função administrativa e judicial. Também chamado “movimentação ou trâmite”.
Transferência	Passagem de documentos do arquivo corrente para o arquivo intermediário.
Unidade de arquivamento	Documento tomado por base para fins de classificação, arranjo, armazenamento e notação. Uma unidade de arquivamento pode ser um dossiê, um processo ou ainda uma pasta em que estão reunidos documentos sob o mesmo código de classificação, como por exemplo, as folhas de ponto de um determinado ano, relatórios de atividades relativos a um determinado período ou atas de reunião.
Usuário	<ul style="list-style-type: none"><li>▪ <b>Gestão de documentos</b></li></ul> Responsáveis, em todos os níveis, pela produção e uso dos documentos institucionais em suas atividades rotineiras, conforme estabelecido pelo programa de gestão. Aquele que é identificável,



	habilitado a interagir com o sistema. <ul style="list-style-type: none"><li>▪ GestãoDoc</li></ul> Aquele que é cadastrado no sistema. Aquele que interage com o sistema.
Usuário autorizado	Aquele que possui níveis de acesso diferenciados atribuídos pelo gestor.
Valor primário	Atribuído aos documentos considerando sua utilidade administrativa imediata, que são, de fato, as razões pelas quais esses documentos foram criados.
Valor secundário	Refere-se ao valor atribuído aos documentos em função do interesse que possam ter para a entidade produtora e outros usuários, tendo em vista a sua utilidade para fins diferentes daqueles para os quais foram originalmente produzidos.
Versão	Estado de um documento em determinada fase de sua elaboração.
Via original	Primeiro documento completo e efetivo.

## 16 Modelos de referência, legislação, regulamentações, normas e referências bibliográficas

### 16.1 Modelos de requisitos para sistemas informatizados de gestão arquivística de documentos

- CONSELHO NACIONAL DE ARQUIVOS — Câmara Técnica De Documentos Eletrônicos (Brasil). *Modelo de Requisitos para Sistemas Informatizados de gestão Arquivística de Documentos*. 2006. Versão 1. Disponível em: <<http://www.conarq.arquivonacional.gov.br/Media/publicacoes/earqbrasilv1.pdf>>. Acesso em: 18 set 2007.
- INSTITUTO DOS ARQUIVOS NACIONAIS/TORRE DO TOMBO, E INSTITUTO DE INFORMÁTICA (Portugal). Modelo de Requisitos para a Gestão de Arquivos Electrónicos. *In: Recomendações para a gestão de*

*documentos de arquivo electrónicos*. Lisboa: O Instituto, 2002. v. 2. Disponível em: <[http://www.iantt.pt/downloads/SIADE\\_Caderno1.pdf](http://www.iantt.pt/downloads/SIADE_Caderno1.pdf)>. Acesso em: 8 set 2007.

- SERCO CONSULTING. *Model Requirements for the management of electronic records*. 2008. Versão 1.02. Disponível em: <<http://www.moreq2.eu/downloads.htm>>. Acesso em: 8 abr 2008.
- THE NATIONAL ARCHIVES OF ENGLAND, WALES AND THE UNITED KINGDOM. *Requirements for electronic records management systems: 1: Functional requirements — 2002 revision: final revision*. Kew: The Archives, 2002. Disponível em: <<http://www.nationalarchives.gov.uk/documents/requirementsfinal.pdf>>. Acesso em: 8 set 2007.

## 16.2 Legislação federal

Um GestãoDoc deve cumprir a legislação e regulamentações vigentes, em especial as relacionadas a seguir:

- **Lei nº 11.419, de 19 de dezembro de 2006**, que dispõe sobre a informatização do processo judicial.
- **Lei nº 11.280, de 16 de fevereiro de 2006**, que altera os arts. 112, 114, 154, 219, 253, 305, 322, 338, 489 e 555 da Lei nº 5.869, de 11 de janeiro de 1973 — Código de Processo Civil, relativos à incompetência relativa, meios eletrônicos, prescrição, distribuição por dependência, exceção de incompetência, revelia, carta precatória e rogatória, ação rescisória e vista dos autos; e revoga o art. 194 da Lei n. 10.406, de 10 de janeiro de 2002 — Código Civil.
- **Lei nº 11.111, de 5 de maio de 2005**, que regulamenta a parte final do disposto no inc. XXXIII do *caput* do art. 5º da Constituição Federal e dá outras providências. (sobre acesso aos documentos públicos de interesse particular ou de interesse coletivo e a questão do sigilo)
- **Decreto nº 5.301, de 9 de dezembro de 2004**, que Regulamenta o disposto na Medida Provisória nº 228, de 9 de dezembro de 2004, que dispõe sobre a ressalva prevista na parte final do disposto no inc. XXXIII do art. 5º da Constituição, e dá outras providências.
- **Decreto nº 5.296, de 2 de dezembro de 2004**, que regulamenta as Leis ns 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências.
- **Decreto nº 4.553, de 27 de dezembro de 2002**, que dispõe sobre a salvaguarda de dados, informações, documentos e materiais sigilosos de interesse da segurança da sociedade e do Estado, no âmbito da

Administração Pública Federal, e dá outras providências.

- **MP nº 2.200-2, de 24 de agosto de 2001**, que institui a Infra-Estrutura de Chaves Públicas Brasileira — ICP-Brasil, transforma o Instituto Nacional de Tecnologia da Informação em autarquia, e dá outras providências.
- **Lei nº 8.159, de 8 de janeiro de 1991**, que dispõe sobre a política nacional de arquivos públicos e privados, em seu art. 20, define a competência e o dever inerente aos órgãos do Poder Judiciário Federal de proceder à gestão de documentos produzidos em razão do exercício de suas funções.

### 16.3 Resoluções do Conselho Nacional de Arquivos — Conarq

- **Resolução nº 14, de 24 de outubro de 2001**, que aprova a versão revisada e ampliada da Resolução do Conarq nº 4, de 28 de março de 1996, que dispõe sobre o Código de Classificação de Documentos de Arquivo para a Administração Pública: Atividades-Meio, a ser adotado como modelo para os arquivos correntes dos órgãos e entidades integrantes do Sistema Nacional de Arquivos (SINAR), e os prazos de guarda e a destinação de documentos estabelecidos na Tabela Básica de Temporalidade e Destinação de Documentos de Arquivo Relativos às Atividades-Meio da Administração Pública.
- **Resolução nº 20, de 16 de julho de 2004**, que dispõe sobre a inserção dos documentos digitais em programas de gestão arquivística de documentos dos órgãos e entidades integrantes do Sistema Nacional de Arquivos.
- **Resolução nº 24, de 3 de agosto de 2006**, que estabelece diretrizes para a transferência e recolhimento de documentos arquivísticos digitais para instituições arquivísticas públicas.
- **Resolução nº 25, de 27 de abril de 2007**, que dispõe sobre a adoção do Modelo de Requisitos para Sistemas Informatizados de Gestão Arquivística de Documentos — e-ARQ Brasil pelos órgãos e entidades integrantes do Sistema Nacional de Arquivos — SINAR.
- **Resolução nº 27, de 17 de fevereiro de 2009**, que dispõe sobre a adoção da Norma Brasileira de Descrição Arquivística - NOBRADE pelos órgãos e entidades integrantes do Sistema Nacional de Arquivos - SINAR, institui o Cadastro Nacional de Entidades Custodiadoras de Acervos Arquivísticos e estabelece a obrigatoriedade da adoção do Código de Entidades Custodiadoras de Acervos Arquivísticos - CODEARQ.

### 16.4 Normas

- ISO 9660:1988 — Information processing — Volume and file structure of CD-ROM for information interchange.

- ISO 14721:2003 — Space data and information transfer systems — Open archival information system (OAIS) — Reference model.
- ISO 15408 — Common Criteria 2.X.
- ISO/IEC 9126:1991 Information technology — Software product evaluation: quality characteristics and guidelines for their use.
- ISO/IEC 17799:2005 — Information technology — Security techniques — Code of practice for information security management.
- ISO/IEC 27001:2005 — Information technology — Security techniques — Information security management systems — Requirements.
- AS ISO 15489.1 — Australian Standard Records Management. Part 1: General, 2002.
- AS ISO 15489-2 — Australian Standard Records Management. Part 2: Guidelines, 2002.

## 16.5 Referências bibliográficas

ARELLANO, Miguel Ángel Márdero. Preservação Digital. In: *Seminário de Gestão da Informação Jurídica em Espaços Digitais*, Supremo Tribunal Federal, Brasília. 2007. Disponível em: <[www.stf.gov.br/sijed/Palestras/17.pdf](http://www.stf.gov.br/sijed/Palestras/17.pdf)>. Acesso em: 9 set 2007.

ARELLANO, Miguel Ángel Márdero, ANDRADE, Ricardo Sodré. Preservação digital e os profissionais da informação. DataGramZero — *Revista de Ciência da Informação* — v.7 n.5 out/06. Disponível em: <[http://www.dgz.org.br/out06/Art\\_05.htm](http://www.dgz.org.br/out06/Art_05.htm)>. Acesso em: 9 set 2007.

ARQUIVO NACIONAL (Brasil). *Dicionário Brasileiro de Terminologia Arquivística*. Rio de Janeiro: Arquivo Nacional, 2005.

CONSELHO INTERNACIONAL DE ARQUIVOS. Comitê de arquivos correntes em ambiente eletrônico. *Documentos de arquivo eletrônicos: manual para arquivistas*. ICA, Estudo nº 16. 2005. Disponível em: <<http://www.ica.org/en/node/30273>>. Acesso em: 8 set 2007.

CONSELHO NACIONAL DE ARQUIVOS. *Carta para a Preservação do Patrimônio Arquivístico Digital*, 2004. Disponível em: <<http://www.conarq.arquivonacional.gov.br/Media/publicacoes/cartapreservpatrimarqdigitalconarq2004.pdf>>. Acesso em: 21 jun 2007.

INNARELLI Humberto Celeste. Documentos digitais e sua fragilidade em relação ao suporte. In: *II Simpósio Internacional de Bibliotecas Digitais*. IBICT, Unicamp, 2004. Disponível em: <<http://libdigi.unicamp.br/document/?view=8397>>. Acesso em: 21 jun 2007.

*Manual de Gestão de Autos Findos do Programa de Gestão Documental da Justiça Federal*. Brasília, 2005. 36 p. Disponível em: <[http://daleth.cjf.gov.br/Download/Manual%20Gestão%20Documental\\_21.doc](http://daleth.cjf.gov.br/Download/Manual%20Gestão%20Documental_21.doc)>. Acesso em: 29 abr 2007.

*Manual de Procedimentos do Programa de Gestão Documental da Justiça Federal*, CJP, Brasília, 2001. 59 p. Disponível em: <<http://www.jf.gov.br/portal/gestaodocumental/documentos/MANUAL%20DE%20PROCEDIMENTOS.pdf>>. Acesso em: 29 abr 2007.

NATIONAL LIBRARY OF AUSTRALIA — NLA. *Guidelines for the preservation of digital heritage*. Paris: UNESCO, 2003. 177p. Disponível em: <<http://unesdoc.unesco.org/images/0013/001300/130071e.pdf>>. Acesso em: 6 set 2007.

OCLC, RLG. *Data Dictionary for Preservation Metadata*. 2005. 237p. Disponível em: <<http://www.loc.gov/standards/premis/>>. Acesso em: 25 mar 2008.

ROCHA, Cláudia Lacombe. *Gestão e preservação de documentos arquivísticos digitais*. GED RIO, 1, 2006, Rio de Janeiro. Anais. Rio de Janeiro: Centro Nacional de Gerenciamento da Informação, 2006.

RONDINELLI, Rosely Cury. *Gerenciamento arquivístico de documentos eletrônicos: uma abordagem teórica da diplomática arquivística contemporânea*. Rio de Janeiro: Editora FGV, 2002. 160 p.

SANTOS, Vanderlei Batista dos. *Gestão de documentos eletrônicos: uma visão arquivística*. Brasília: ABARQ, 2002. 140p.

THOMAZ, Kátia P. *Gestão e preservação de documentos eletrônicos de arquivo: revisão de literatura: parte 1*. Arquivística.net, v. 2, n. 1, 2006. Disponível em: <<http://www.arquivistica.net>>. Acesso em: 8 set 2007.

THOMAZ, Kátia P. *Gestão e preservação de documentos eletrônicos de arquivo: revisão de literatura: parte 2*. Arquivística.net, v. 2, n. 1, 2006. Disponível em: <<http://www.arquivistica.net/>>. Acesso em: 08 set 2007.

UNESCO. División de la Sociedad de la Información. *Directrices para la preservación del patrimonio digital*. Preparado por la Biblioteca Nacional de Australia. Canberra: Biblioteca Nacional de Austrália, 2002. 176p. Disponível em: <[http://unesdoc.unesco.org/ulis/cgi-bin/ulis.pl?database=ged&req=2&by=3&sc1=1&look=new&sc2=1&text\\_p=inc&text=Directrices+para+la+preservaci%F3n+del+patrimonio+digital&submit=GO](http://unesdoc.unesco.org/ulis/cgi-bin/ulis.pl?database=ged&req=2&by=3&sc1=1&look=new&sc2=1&text_p=inc&text=Directrices+para+la+preservaci%F3n+del+patrimonio+digital&submit=GO)>. Acesso em: 8 ago 2006.

# Apêndice 1 - Metadados

Este apêndice apresenta os elementos de metadados de segurança, auditoria e preservação do Modelo de Requisitos para Sistemas Informatizados de Gestão de Processos e Documentos do Judiciário brasileiro (MoReq-Jus) por serem comuns a qualquer sistema, independente da funcionalidade. Os metadados de conteúdo serão desenvolvidos de acordo com a especificidade de cada sistema.

Segundo o MoReq-Jus, metadado é o dado estruturado que descreve e permite encontrar, gerenciar, compreender e/ou preservar documentos institucionais ao longo do tempo. Um elemento de metadado, por sua vez, pode ser visto como uma variável utilizada para guardar um valor de metadado. Por exemplo, um elemento de metadado denominado Data empregado para guardar o valor “08/2008”.

Neste apêndice, os elementos de metadados são apresentados em uma tabela similar à Tabela 1.

Código de identificação	
Nome	
Definição	
Aplica-se a	
Obrigatoriedade	
Ocorrências	
Responsável	
Herança	
Condições de uso	
Comentário	
Requisitos	
Referências	

Tabela 1 Tabela de elementos de metadados

As informações que identificam um elemento de metadado – código de identificação e nome – encontram-se em destaque na tabela.

O código de identificação inicia com a letra M, seguido de duas letras relacionadas ao assunto a que se refere o elemento de metadado e uma seqüência numérica de três dígitos (x.y.z). Desta forma:

- MSGx.y.z – identifica elementos de metadados de segurança
- MADx.y.z – identifica elementos de metadados de auditoria
- MPRx.y.z – identifica elementos de metadados de preservação

O nome é composto por um ou mais termos relevantes para o elemento de metadado. Se mais de um termo é utilizado, eles são separados por underscore “\_”, por exemplo:

- Idioma
- Captura\_data

Cada tabela provê informações sobre um único elemento de metadado. As informações apresentadas podem variar de elemento para elemento, incluindo:

**Código de identificação** – identificador alfanumérico único do elemento.

**Nome** – designação única do elemento.

**Definição** – uma breve descrição do elemento.

**Aplica-se a** – a que se aplica o elemento:

- Classificação, processo/dossiê/volume, anexo e/ou documento. Esta opção, quando utilizada, deve ser agrupada a Obrigatoriedade, Ocorrências e Responsável, conforme Tabela 2. Os anexos também compreendem os apensos baixados. Apendos em andamento são tratados como processos.
- Documento ou componente.
- Componente.

**Obrigatoriedade** – se um valor para o elemento é obrigatório ou opcional.

**Ocorrências** – se mais de um valor é permitido para o elemento. O termo “uma” indica que apenas um valor é permitido. O termo “várias” indica que um ou mais valores são permitidos. Ambos permitem que os valores não sejam atribuídos ao elemento, se a obrigatoriedade estiver definida como opcional.

**Responsável** – responsável pela criação do valor do elemento: sistema, gestor ou usuário.

**Herança** – indica a herança dos valores de metadados. Apresentada apenas nas situações em que a herança pode ser relevante.

**Condições de uso** – condições e regras que controlam o uso e valor(es) do elemento.

**Comentário** – informações adicionais sobre o elemento.

**Requisitos** – requisitos formais do MoReq-Jus relacionados ao elemento. Em alguns casos, a lista de requisitos pode não ser completa.

**Referências** – referências externas utilizadas como base para especificação do elemento.

Código de identificação				
Nome				
Definição				
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade				
Ocorrências				
Responsável				
Herança				
Condições de uso				
Comentário				
Requisitos				
Referências				

Tabela 2 Tabela de elementos de metadados aplicada a classificação, processo/dossiê/volume, anexo e/ou documento



## 1.1 Elementos de metadados de segurança

Código de identificação	MSG1.1.1			
Nome	Assinatura_digital_data			
Definição	Data e hora da verificação da assinatura digital de um documento.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Não se aplica	Obrigatório
Ocorrências		Uma		Uma
Responsável		Sistema		Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de processo/dossiê/volume e documento assinado digitalmente. Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.5.1; RSE6.5.3; RSE6.5.4			
Referências	MoReq2: M114 ISO: 17799-12.3.2-b			

Código de identificação	MSG1.1.2			
Nome	Assinatura_digital_validação			
Definição	Registro de que a assinatura digital foi verificada com sucesso.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Não se aplica	Obrigatório
Ocorrências		Uma		Uma
Responsável		Sistema		Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de processo/dossiê/volume e documento assinado digitalmente. Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.5.1; RSE6.5.3; RSE6.5.4			
Referências	MoReq2: M113			

Código de identificação	MSG1.1.3			
Nome	Certificado_digital			
Definição	O certificado digital do signatário.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Não se aplica	Obrigatório
Ocorrências		Uma		Uma
Responsável		Sistema		Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de processo/dossiê/volume e documento assinado digitalmente. Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.5.2, RSE6.5.5			
Referências	MoReq2: M110 ISO: 17799-12.3.2-b			

Código de identificação	MSG1.1.4			
Nome	Assinatura_digital_assinatura			
Definição	Cópia da assinatura digital recebida com o documento.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Opcional	Não se aplica	Obrigatório
Ocorrências		Uma		Várias
Responsável		Sistema		Sistema
Herança	Não			
Condições de uso	<p>Opcional no caso de processo/dossiê/volume, exceto quando a regra de negócio exigir a assinatura digital.</p> <p>Obrigatório no caso de documento assinado digitalmente.</p> <p>Não pode ser modificado.</p>			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.5.5			
Referências	MoReq2: M076			

Código de identificação	MSG1.1.5			
Nome	Assinatura_digital_validador			
Definição	Identificação da pessoa que atestou a verificação da assinatura digital.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Não se aplica	Obrigatório
Ocorrências		Uma		Uma
Responsável		Sistema		Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de processo/dossiê/volume e documento assinado digitalmente. Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.5.4			
Referências	MoReq2: M147			

Código de identificação	MSG1.1.6			
Nome	Criptografia_recebida			
Definição	Indica se o documento foi recebido em formato criptografado.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Opcional	Não se aplica	Obrigatório
Ocorrências		Uma		Uma
Responsável		Gestor ou sistema		Gestor ou sistema
Herança	Não			
Condições de uso	Opcional no caso de processo/dossiê/volume, exceto quando a regra de negócio exigir a criptografia. Obrigatório no caso de documento criptografado. Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.6.1; RSE6.6.3; RSE6.6.4			
Referências	MoReq2: M143			

Código de identificação	MSG1.1.7			
Nome	Criptografia_recebida_algoritmo			
Definição	Identificação do algoritmo usado em documento recebido em formato criptografado.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Sistema
Herança	Não			
Condições de uso	Só deverão ser aceitos os métodos criptográficos autorizados para uso no sistema.			
Comentário	Deve ser registrado o tipo de algoritmo e nível de criptografia utilizado.			
Requisitos	MoReq-Jus: RSE6.6.1; RSE6.6.3; RSE6.6.4			
Referências	MoReq2: M116			

Código de identificação	MSG1.1.8			
Nome	Criptografia_armazenada			
Definição	Identifica se o documento foi armazenado em formato criptografado.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Sistema
Herança	Não			
Condições de uso	Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.6.1; RSE6.6.3; RSE6.6.4			
Referências	ISO: 17799-12.3.1-b			



Código de identificação	MSG1.1.9			
Nome	Criptografia_armazenada_algoritmo			
Definição	Nome do algoritmo usado em documento armazenado em formato criptografado.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de documento armazenado em formato criptografado. Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.6.1; RSE6.6.3; RSE6.6.4; RSE6.6.5			
Referências	ISO: 17799-12.3.1-b			

Código de identificação	MSG1.1.10			
Nome	Ocultamento_razão			
Definição	Descrição da razão de ocultamento de informações.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Gestor
Herança	Não			
Condições de uso	Obrigatório no caso de documento contendo previsão de ocultamento.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.10.5			
Referências	MoReq2: M080			

Código de identificação	MSG1.1.11			
Nome	Ocultamento_data			
Definição	Data e hora da definição de ocultação no documento.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de documento contendo previsão de ocultamento.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.10.5			
Referências	MoReq2: M081			

Código de identificação	MSG1.1.12			
Nome	Ocultamento_identificação			
Definição	Identificação do documento com os ocultamentos.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de documento com previsão de ocultamento.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.10.5			
Referências	MoReq2: M060			

Código de identificação	MSG1.1.13			
Nome	Ocultamento_criador			
Definição	Identificação da pessoa que criou o ocultamento.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de documento com previsão de ocultamento.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.10.5			
Referências	MoReq2: M079			

Código de identificação	MSG1.1.14			
Nome	Ocultamento_referência_ao_original			
Definição	Identificação do documento matriz (sem ocultamentos).			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Sistema
Herança	Não			
Condições de uso	A identificação do documento matriz tem que ser automaticamente modificada nos casos de relocação dentro do esquema de classificação.			
Comentário	O ocultamento de um documento deve acarretar a correspondente reclassificação de sigilo do documento original.			
Requisitos	MoReq-Jus: - RSE6.10.5			
Referências	MoReq2: M083			

Código de identificação	MSG1.1.15			
Nome	Perfil_leitura			
Definição	Identificação do perfil de usuário para leitura do documento.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Várias
Responsável				Gestor
Herança	Sim			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.2.13			
Referências	-			

Código de identificação	MSG1.1.16			
Nome	Perfil_alteração_corpo			
Definição	Identificação do perfil de usuário para alteração de documento ou processo/dossiê/volume.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Não se aplica	Obrigatório
Ocorrências		Várias		Várias
Responsável		Gestor		Gestor
Herança	Sim			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.2.13			
Referências	-			

Código de identificação	MSG1.1.17			
Nome	Perfil_alteração_metadado			
Definição	Identificação do perfil de usuário para alteração de metadado de documento ou processo/dossiê/volume.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Obrigatório	Obrigatório	Não se aplica	Obrigatório
Ocorrências	Várias	Várias		Várias
Responsável	Gestor	Gestor		Gestor
Herança	Sim			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.2.13			
Referências	-			

Código de identificação	MSG1.1.18			
Nome	Perfil_alteração_classificação			
Definição	Identificação do perfil de usuário para alteração de classificação.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Obrigatório	Obrigatório	Não se aplica	Obrigatório
Ocorrências	Várias	Várias		Várias
Responsável	Gestor	Gestor		Gestor
Herança	Sim			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.2.13			
Referências	-			

Código de identificação	MSG1.1.19			
Nome	Perfil_eliminação			
Definição	Identificação do perfil de usuário para eliminação.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Várias	Várias	Várias
Responsável		Gestor	Gestor	Gestor
Herança	Sim			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.2.13			
Referências	-			

Código de identificação	MSG1.1.20			
Nome	Sigilo_recebido			
Definição	Identificação do sigilo original em documento recebido.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Gestor	Gestor	Gestor
Herança	Sim			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.3.5, RSE6.3.12			
Referências	-			

Código de identificação	MSG1.1.21			
Nome	Sigilo_armazenado			
Definição	Identificação do sigilo em documento armazenado.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Gestor	Gestor	Gestor
Herança	Sim			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.3.2; RSE6.3.5; RSE6.6.1			
Referências	-			

Código de identificação	MSG1.1.22			
Nome	Marca_d'água_presente			
Definição	Indica se o documento contém marca d'água.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Opcional
Ocorrências				Uma
Responsável				Gestor
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.7.2			
Referências	-			



Código de identificação	MSG1.1.23			
Nome	Marca_d'água_descrição			
Definição	Descrição técnica das características da marca d'água.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Opcional
Ocorrências				Uma
Responsável				Gestor
Herança	Não			
Condições de uso	-			
Comentário	Só são aceitáveis as marcas d'água reconhecidas pelo Judiciário.			
Requisitos	MoReq-Jus: RSE6.7.2			
Referências	-			

## 1.2 Elementos de metadados de auditoria

Código de identificação	MAD1.2.1			
Nome	Proteção_contra_eliminação_aplicador			
Definição	Identificação do responsável pela aplicação da proteção contra eliminação.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Sim			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.4.2, RSE6.10.2, RPC2.1.8, RPC2.3.8			
Referências	MoReq2: M035			

Código de identificação	MAD1.2.2			
Nome	Proteção_contra_eliminação_desativador			
Definição	Identificação do responsável pela desativação da proteção contra eliminação.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.4.2, RPC2.3.3			
Referências	MoReq2: M134			

Código de identificação	MAD1.2.3			
Nome	Proteção_contra_eliminação_data_de_aplicação			
Definição	Data e hora da aplicação da proteção contra eliminação.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.4.2, RSE6.10.2, RPC2.1.8, RPC2.3.8			
Referências	MoReq2: M034			

Código de identificação	MAD1.2.4			
Nome	Proteção_contra_eliminação_data_de_desativação			
Definição	Data e hora da desativação da proteção contra eliminação.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.4.2, RPC2.3.3			
Referências	MoReq2: M102			

Código de identificação	MAD1.2.5			
Nome	Proteção_contra_eliminação_razão_da_aplicação			
Definição	Razão da aplicação da proteção contra eliminação.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Gestor	Gestor	Gestor
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.4.2, RSE6.10.2, RPC2.1.8, RPC2.3.5, RPC2.3.8			
Referências	MoReq2: M135			

Código de identificação	MAD1.2.6			
Nome	Proteção_contra_descarte_razão_da_desativação			
Definição	Razão da desativação da proteção contra descarte.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Gestor	Gestor	Gestor
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.4.2, RPC2.3.3			
Referências	MoReq2: M136			

Código de identificação	MAD1.2.7			
Nome	Sigilo_atribuído_data			
Definição	Data e hora de atribuição do sigilo.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	-			
Referências	-			

Código de identificação	MAD1.2.8			
Nome	Sigilo_atribuído_responsável			
Definição	Identificação do responsável pela atribuição de sigilo.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Sim			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.3.8, RSE6.3.10			
Referências	-			

Código de identificação	MAD1.2.9			
Nome	Sigilo_revogado_data			
Definição	Data e hora de revogação do sigilo.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	-			
Referências	-			

Código de identificação	MAD1.2.10			
Nome	Sigilo_revogado_responsável			
Definição	Identificação do responsável pela revogação de sigilo.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.3.8, RSE6.3.10			
Referências	-			

Código de identificação	MAD1.2.11			
Nome	Ocultamento_atribuído_data			
Definição	Data e hora de atribuição do ocultamento.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de documento contento previsão de ocultamento.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.10.5			
Referências	MoReq2: M025			

Código de identificação	MAD1.2.12			
Nome	Ocultamento_atribuído_responsável			
Definição	Identificação do responsável pelo ocultamento.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Sim			
Condições de uso	Obrigatório no caso de documento contento previsão de ocultamento.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.10.3, RSE6.10.5			
Referências	MoReq2: M079			



Código de identificação	MAD1.2.13			
Nome	Ocultamento_revogado_data			
Definição	Data e hora de revogação do ocultamento.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de documento contento previsão de ocultamento.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.10.5			
Referências	MoReq2: M025			

Código de identificação	MAD1.2.14			
Nome	Ocultamento_revogado_responsável			
Definição	Identificação do responsável pelo ocultamento.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Obrigatório no caso de documento contento previsão de ocultamento.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.10.3, RSE6.10.5			
Referências	-			

Código de identificação	MAD1.2.15			
Nome	Destinação_data_agendada			
Definição	Data e hora do agendamento para a destinação do item.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Não se aplica	Obrigatório
Ocorrências		Uma		Uma
Responsável		Gestor ou sistema		Gestor ou sistema
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RAD8.1.2			
Referências	MoReq2: M138			

Código de identificação	MAD1.2.16			
Nome	Destinação_ avaliação			
Definição	Avaliação da destinação do item.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Não se aplica	Obrigatório
Ocorrências		Uma		Uma
Responsável		Gestor ou sistema		Gestor ou sistema
Herança	Não			
Condições de uso	Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RAD8.1.2			
Referências	MoReq2: M053			

Código de identificação	MAD1.2.17			
Nome	Eliminação_responsável			
Definição	Identificação do responsável pela eliminação do item.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Eliminação antecipada ou autônoma de um anexo quando determinada pela lei. Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.4.2, RSE6.4.3			
Referências	MoReq2: M161			

Código de identificação	MAD1.2.18			
Nome	Movimentação_responsável			
Definição	Identificação do responsável pela movimentação ou transferência do item.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.8.3			
Referências	MoReq2: M097			

Código de identificação	MAD1.2.19			
Nome	Eliminação_registro			
Definição	Registro de que o item foi eliminado.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Não pode ser modificado. O valor pode ser <i>Sim</i> ou <i>Não</i> .			
Comentário	-			
Requisitos	MoReq-Jus: RAD8.4.9			
Referências	MoReq2: M155			

Código de identificação	MAD1.2.20			
Nome	Transferência_registro			
Definição	Registro de que o item foi transferido.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Não pode ser modificado. O valor pode ser <i>Sim</i> ou <i>Não</i> .			
Comentário	-			
Requisitos	MoReq-Jus: RAD8.3.7			
Referências	MoReq2: M155			

Código de identificação	MAD1.2.21			
Nome	Reclassificação_razão			
Definição	Razão da reclassificação do item.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Obrigatório	Opcional	Opcional	Opcional
Ocorrências	Uma	Uma	Uma	Uma
Responsável	Gestor	Usuário ou sistema	Usuário ou sistema	Usuário ou sistema
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.4.2, RAD8.2.4, RPC2.3.3, RPC2.3.4			
Referências	MoReq2: M140			

Código de identificação	MAD1.2.22			
Nome	Eliminação_data			
Definição	Registro da data da eliminação do item.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.4.2			
Referências	MoReq2: M141			

Código de identificação	MAD1.2.23			
Nome	Transferência_data			
Definição	Registro da data da transferência do item.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RAD8.1.5, RSE6.8.1			
Referências	MoReq2: M133			



Código de identificação	MAD1.2.24			
Nome	Reclassificação_data			
Definição	Data da última reclassificação do item.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Obrigatório	Obrigatório	Obrigatório	Obrigatório
Ocorrências	Uma	Uma	Uma	Uma
Responsável	Sistema	Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.4.2, RAD8.1.8			
Referências	MoReq2: M156			

Código de identificação	MAD1.2.25			
Nome	Transferência_para			
Definição	Identificação do destino para o qual o item foi transferido.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Gestor	Gestor	Gestor
Herança	Não			
Condições de uso	Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RPL9.1.1, RSE6.8.1			
Referências	MoReq2: M194			

Código de identificação	MAD1.2.26			
Nome	Critério_de_guarda			
Definição	Identificação do critério de guarda atribuído ao item.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Obrigatório	Obrigatório	Obrigatório	Obrigatório
Ocorrências	Uma	Uma	Uma	Uma
Responsável	Gestor	Gestor ou sistema	Gestor ou sistema	Gestor ou sistema
Herança	Não			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RAD8.1.3, RAD8.2.5			
Referências	MoReq2: M087			

Código de identificação	MAD1.2.27			
Nome	Backup_data			
Definição	Registro da data e hora em que o item recebeu a mais recente ação de backup interno.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Obrigatório	Obrigatório	Obrigatório	Obrigatório
Ocorrências	Uma	Uma	Uma	Uma
Responsável	Sistema	Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RSE6.1.2			
Referências	-			

Código de identificação	MAD1.2.28				
Nome	Restore_data				
Definição	Registro da data e hora em que o item foi restaurado de backups internos.				
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento	
Obrigatoriedade	Obrigatório	Obrigatório	Obrigatório	Obrigatório	
Ocorrências	Uma	Uma	Uma	Uma	
Responsável	Sistema	Sistema	Sistema	Sistema	
Herança	Não				
Condições de uso	Não pode ser modificado.				
Comentário	-				
Requisitos	MoReq-Jus: RSE6.1.2				
Referências	-				

Código de identificação	MAD1.2.29				
Nome	Sigilo_reclassificação				
Definição	Registro da reclassificação do grau de sigilo.				
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento	
Obrigatoriedade	Obrigatório	Obrigatório	Obrigatório	Obrigatório	
Ocorrências	Uma	Uma	Uma	Uma	
Responsável	Gestor	Gestor	Gestor	Gestor	
Herança	Não				
Condições de uso	-				
Comentário	-				
Requisitos	MoReq-Jus: RSE6.3.1, RSE6.3.2				
Referências	-				

Código de identificação	MAD1.2.30			
Nome	Eliminação_motivo			
Definição	Registro do motivo da eliminação.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Obrigatório	Obrigatório
Ocorrências		Uma	Uma	Uma
Responsável		Gestor	Gestor	Gestor
Herança	Sim			
Condições de uso	-			
Comentário	-			
Requisitos	MoReq-Jus: RAD8.4.9			
Referências	-			

### 1.3 Elementos de metadados de preservação

Código de identificação	MPR1.3.1			
Nome	Entidade_identificação			
Definição	Identificação da entidade.			
Aplica-se a	o Classificação	Processo / dossiê / volume	Anexo	o Document
Obrigatoriedade	Obrigatório	o Obrigatóri	Obrigatóri	o Obrigatóri
Ocorrências	Uma	Uma	Uma	Uma
Responsável	Sistema	Sistema	Sistema	Sistema
Herança	Não			
Condições de uso	-			
Comentário	Entidade refere-se a qualquer classificação, processo/dossiê/volume, anexo ou documento.			
Requisitos	MoReq-Jus: RCA3.1.9, RCA3.1.10, RCA3.1.11, RCA3.1.16, RPR5.3.7			
Referências	MoReq2: M020			

Código de identificação	MPR1.3.2			
Nome	Idioma			
Definição	Idioma em que o conteúdo do documento está escrito.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Opcional
Ocorrências				Várias
Responsável				Usuário ou sistema
Herança	Não			
Condições de uso	-			
Comentário	Esta informação pode ser necessária para acesso ao documento (fontes específicas, por exemplo).			
Requisitos	-			
Referências	MoReq2: M145			

Código de identificação	MPR1.3.3			
Nome	Captura_data			
Definição	Data e hora em que o documento foi capturado.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Obrigatório	Não se aplica	Obrigatório
Ocorrências		Uma		Uma
Responsável		Sistema		Sistema
Herança	Não			
Condições de uso	Não pode ser modificado.			
Comentário	-			
Requisitos	MoReq-Jus: RCA3.1.16			
Referências	MoReq2: M071			

Código de identificação	MPR1.3.4			
Nome	Status_permanente			
Definição	Indica se uma entidade precisa ser conservada permanentemente.			
Aplica-se a	o Classificaçã	o Processo / dossiê / volume	o Anexo	o Document
Obrigatoriedade	o Obrigatório	o Obrigatóri	o Obrigatóri	o Obrigatóri
Ocorrências	Uma	Uma	Uma	Uma
Responsável	Usuário ou sistema	Usuário ou sistema	Usuário ou sistema	Usuário ou sistema
Herança	Não			
Condições de uso	O valor deve ser <i>Sim</i> ou <i>Não</i> .			
Comentário	Aplica-se associado à tabela de temporalidade ou aos critérios de guarda de processos/dossiê e documentos.			
Requisitos	MoReq-Jus: RAD8.1.3, RAD8.2.5			
Referências	MoReq2: M031			

Código de identificação	MPR1.3.5			
Nome	Status_essencial			
Definição	Indica se um documento é essencial.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Usuário
Herança	Não			
Condições de uso	O valor <i>Sim</i> indica que a entidade é essencial. O valor <i>Não</i> indica que a entidade não é essencial.			
Comentário	-			
Requisitos	-			
Referências	MoReq2: M005, Res. CJF n. 23/2008.			

Código de identificação	MPR1.3.6			
Nome	Status_físico			
Definição	Indica se a entidade é física (não digital).			
Aplica-se a	Classificaçã o	Processo / dossiê / volume	Anexo	Document o
Obrigatoriedade	Não se aplica	Obrigatóri o	Obrigatóri o	Obrigatóri o
Ocorrências		Uma	Uma	Uma
Responsável		Usuário	Usuário	Usuário
Herança	Não			
Condições de uso	O valor <i>Sim</i> indica que a entidade é física. O valor <i>Não</i> indica que a entidade não é física.			
Comentário	-			
Requisitos	-			
Referências	MoReq2: M084			



Código de identificação	MPR1.3.7			
Nome	Formato_físico			
Definição	O formato de uma entidade física.			
Aplica-se a	Classificaçã o	Process o / dossiê / volume	Anexo	Document o
Obrigatoriedad e	Não se aplica	Opcional	Opciona l	Opcional
Ocorrências		Várias	Várias	Várias
Responsável		Usuário	Usuário	Usuário
Herança	Não			
Condições de uso	Deve ser usado somente para entidades físicas (entidades em que o valor do elemento de metadado Status_físico é <i>Sim</i> ).			
Comentário	Pode ser usado para descrever qualquer característica física. Recomenda-se o uso de vocabulário controlado para atribuição de valor a este elemento.			
Requisitos	-			
Referências	MoReq2: M092			

Código de identificação	MPR1.3.8			
Nome	Referência_documento_convertido			
Definição	Referência à identificação de um documento convertido associado ao documento originário.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Opcional
Ocorrências				Várias
Responsável				Sistema
Herança	Não			
Condições de uso	<p>Obrigatório onde existe uma conversão de documento. Deve incluir os identificadores encadeados das entidades superiores no ramo apropriado da hierarquia. Pode ser modificado somente se houver alteração nos identificadores encadeados.</p>			
Comentário	<p>Por questão de preservação, quando houver conversão de formato, é preciso manter o original para casos de perda de informação.</p>			
Requisitos	-			
Referências	MoReq2: M148			

Código de identificação	MPR1.3.9			
Nome	Documento_convertido_identificação			
Definição	Identificação única de um documento convertido.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Sistema
Herança	Não			
Condições de uso	Obrigatório para documentos convertidos. Não pode ser modificado.			
Comentário	Aplicado ao documento convertido, não ao documento originário.			
Requisitos	-			
Referências	MoReq2: M125			

Código de identificação	MPR1.3.10			
Nome	Conversão_data			
Definição	Data e hora em que o documento foi convertido.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Várias
Responsável				Sistema
Herança	Não			
Condições de uso	Usado quando pelo menos um componente do documento é convertido. Não pode ser modificado.			
Comentário	Aplicado ao documento convertido, não ao documento originário.			
Requisitos	-			
Referências	MoReq2: M127			

Código de identificação	MPR1.3.11			
Nome	Referência_documento_originário			
Definição	Referência à identificação do documento que foi objeto da conversão.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Uma
Responsável				Sistema
Herança	Não			
Condições de uso	Obrigatório para documentos convertidos. Deve incluir os identificadores encadeados das entidades superiores no ramo apropriado da hierarquia. Pode ser modificado somente se houver alteração nos identificadores encadeados.			
Comentário	-			
Requisitos	-			
Referências	MoReq2: M149			

Código de identificação	MPR1.3.12			
Nome	Conversão_motivo			
Definição	Motivo da conversão do documento.			
Aplica-se a	Classificação	Processo / dossiê / volume	Anexo	Documento
Obrigatoriedade	Não se aplica	Não se aplica	Não se aplica	Obrigatório
Ocorrências				Várias
Responsável				Usuário
Herança	Não			
Condições de uso	Obrigatório para documentos convertidos. Não pode ser modificado.			
Comentário	Aplicado ao documento convertido, não ao documento originário.			
Requisitos	-			
Referências	MoReq2: M126			

Código de identificação	MPR1.3.13
Nome	Componente_identificação
Definição	Identificação de um componente de documento.
Aplica-se a	Componente
Obrigatoriedade	Obrigatório
Ocorrências	Uma
Responsável	Sistema
Herança	Não
Condições de uso	Não pode ser modificado.
Comentário	-
Requisitos	MoReq-Jus: RCA3.4.4, RPR5.3.7
Referências	MoReq2: M064

Código de identificação	MPR1.3.14
Nome	Referência_componente_convertido
Definição	Referência à identificação de um componente convertido associado ao componente originário.
Aplica-se a	Componente
Obrigatoriedade	Obrigatório
Ocorrências	Várias
Responsável	Sistema
Herança	Não
Condições de uso	Obrigatório onde existe uma conversão de componente. Deve incluir os identificadores encadeados das entidades superiores no ramo apropriado da hierarquia. Pode ser modificado somente se houver alteração nos identificadores encadeados.
Comentário	-
Requisitos	-
Referências	MoReq2: M150

Código de identificação	MPR1.3.15
Nome	Componente_convertido_identificação
Definição	Identificação única de um componente convertido.
Aplica-se a	Componente
Obrigatoriedade	Obrigatório
Ocorrências	Uma
Responsável	Sistema
Herança	Não
Condições de uso	Obrigatório para componentes convertidos. Não pode ser modificado.
Comentário	Aplicado ao componente convertido, não ao componente originário.
Requisitos	-
Referências	MoReq2: M130

Código de identificação	MPR1.3.16
Nome	Conversão_componente_data
Definição	Data e hora em que o componente foi convertido.
Aplica-se a	Componente
Obrigatoriedade	Obrigatório
Ocorrências	Uma
Responsável	Sistema
Herança	Não
Condições de uso	Obrigatório para componentes convertidos. Não pode ser modificado.
Comentário	Aplicado ao componente convertido, não ao componente originário.
Requisitos	-
Referências	MoReq2: M132

Código de identificação	MPR1.3.17
Nome	Referência_componente_originário
Definição	Referência à identificação do componente que foi objeto da conversão.
Aplica-se a	Componente
Obrigatoriedade	Obrigatório
Ocorrências	Uma
Responsável	Sistema
Herança	Não
Condições de uso	Obrigatório para componentes convertidos. Deve incluir os identificadores encadeados das entidades superiores no ramo apropriado da hierarquia. Pode ser modificado somente se houver alteração nos identificadores encadeados.
Comentário	-
Requisitos	-
Referências	MoReq2: M151



Código de identificação	MPR1.3.18
Nome	Conversão_componente_motivo
Definição	Motivo da conversão do componente.
Aplica-se a	Componente
Obrigatoriedade	Obrigatório
Ocorrências	Várias
Responsável	Usuário
Herança	Não
Condições de uso	Não pode ser modificado.
Comentário	Aplicado ao componente convertido, não ao componente original.
Requisitos	-
Referências	MoReq2: M131

Código de identificação	MPR1.3.19
Nome	Formato_arquivo_digital
Definição	Formato do arquivo em que o componente ou documento está codificado.
Aplica-se a	Documento ou componente
Obrigatoriedade	Obrigatório
Ocorrências	Uma
Responsável	Sistema
Herança	Não
Condições de uso	Não pode ser modificado.
Comentário	Recomenda-se o uso de vocabulário controlado para atribuição de valor a este elemento.
Requisitos	MoReq-Jus: RCA3.4.1, RCA3.4.5
Referências	MoReq2: M128; Premis: formatName

Código de identificação	MPR1.3.20
Nome	Versão_formato_arquivo
Definição	Versão do formato do arquivo em que o componente ou o documento está codificado.
Aplica-se a	Documento ou componente
Obrigatoriedade	Obrigatório
Ocorrências	Uma
Responsável	Sistema
Herança	Não
Condições de uso	Não pode ser modificado.
Comentário	Recomenda-se o uso de vocabulário controlado para atribuição de valor a este elemento.
Requisitos	-
Referências	MoReq2: M129, Premis: formatVersion

Código de identificação	MPR1.3.21
Nome	Formato_arquivo_original
Definição	Formato do arquivo em que o componente ou o documento estava codificado no momento do recebimento.
Aplica-se a	Documento ou componente
Obrigatoriedade	Obrigatório
Ocorrências	Uma
Responsável	Sistema
Herança	Não
Condições de uso	Não pode ser modificado.
Comentário	Recomenda-se o uso de vocabulário controlado para atribuição de valor a este elemento.
Requisitos	-
Referências	MoReq2: M133

Código de identificação	MPR1.3.22
Nome	Versão_formato_arquivo_original
Definição	Versão do formato do arquivo em que o componente ou documento estava codificado no momento do recebimento.
Aplica-se a	Documento ou componente
Obrigatoriedade	Obrigatório
Ocorrências	Uma
Responsável	Sistema
Herança	Não
Condições de uso	Não pode ser modificado.
Comentário	Recomenda-se o uso de vocabulário controlado para atribuição de valor a este elemento.
Requisitos	-
Referências	MoReq2: M142

Código de identificação	MPR1.3.23
Nome	Algoritmo_checksum
Definição	Algoritmo utilizado para gerar o checksum para o componente ou o documento.
Aplica-se a	Documento ou componente
Obrigatoriedade	Obrigatório
Ocorrências	Uma
Responsável	Sistema
Herança	Não
Condições de uso	Obrigatório para componente ou documento que possui checksum. Pode ser modificado. Nesse caso, o checksum do componente ou documento precisa ser recalculado e armazenado no elemento Checksum.
Comentário	Recomenda-se o uso de vocabulário controlado para atribuição de valor a este elemento.
Requisitos	MoReq-Jus: RPR5.2.2
Referências	Premis: messageDigestAlgorithm

Código de identificação	MPR1.3.24
Nome	Checksum
Definição	Checksum do componente ou documento, gerado com o algoritmo especificado em Algoritmo_checksum.
Aplica-se a	Documento ou componente
Obrigatoriedade	Opcional
Ocorrências	Uma
Responsável	Sistema
Herança	Não
Condições de uso	Modificado automaticamente se o componente for alterado de maneira autorizada. Caso contrário, não pode ser modificado.
Comentário	Recomenda-se o uso do checksum para documentos essenciais e de guarda permanente.
Requisitos	MoReq-Jus: RPR5.2.2
Referências	Premis: messageDigest

Código de identificação	MPR1.3.25
Nome	Dependência_software
Definição	Software necessário para apresentar ou usar um documento ou um componente.
Aplica-se a	Documento ou componente
Obrigatoriedade	Opcional
Ocorrências	Várias
Responsável	Usuário
Herança	Não
Condições de uso	Não pode ser modificado.
Comentário	Recomenda-se o uso de vocabulário controlado para atribuição de valor a este elemento.
Requisitos	-
Referências	Premis: software

Código de identificação	MPR1.3.26
Nome	Dependência_hardware
Definição	Hardware necessário para o software referenciado em Dependência_software ou para o usuário desse software.
Aplica-se a	Documento ou componente
Obrigatoriedade	Opcional
Ocorrências	Várias
Responsável	Usuário
Herança	Não
Condições de uso	Não pode ser modificado.
Comentário	Recomenda-se o uso de vocabulário controlado para atribuição de valor a este elemento.
Requisitos	MPR5.1.1
Referências	Premis: hardware

---

▪ **Equipe técnica responsável pela elaboração do MoReq-Jus (Justiça Federal)**

Alexandre Libonati	Juiz Federal da Seção Judiciária do Rio de Janeiro
Eduardo César Weber	Diretor da Secretaria de Registros e Informações Processuais – Tribunal Regional Federal da 4ª Região
Jader Carlos Videira	Diretor da Divisão de Informática dos Juizados Especiais Federais – Tribunal Regional Federal da 3ª Região
Jany Rocha Wursch	Chefe da Seção de Integração com as Seções Judiciárias – Tribunal Regional Federal da 5ª Região
Lúcio Melre da Silva	Secretário de Tecnologia da Informação do Conselho da Justiça Federal
Luis Carlos de Freitas	Diretor da Subsecretaria de Informática da Seção Judiciária do Rio de Janeiro
Nádia Barbosa da Cruz Santana	Diretora da Divisão de Arquivo e Memória Institucional – Tribunal Regional Federal da 1ª Região
Neide Alves Dias De Sordi	Secretária de Pesquisa e Informação Jurídicas do Centro de Estudos Judiciários do Conselho da Justiça Federal
Patrícia Reis Longhi	Diretora-Geral da Seção Judiciária do Rio de Janeiro
Rita Helena dos Anjos	Coordenadora de Estudos e Pesquisas do Centro de Estudos Judiciários do Conselho da Justiça Federal

---

**Participaram da versão 1.1 - Metadados de Segurança, Preservação e Auditoria**

Paulo Martins Inocêncio	Conselho da Justiça Federal
Marlon Borba	Tribunal Regional Federal da 3ª Região
Leonardo Pires	Superior Tribunal de Justiça
Susana Franco	Tribunal Regional Federal da 4ª Região
Ricardo de Abreu Neves	Tribunal Regional Federal da 4ª Região
Wilmar Barros de Castro	Coordenador de Gestão Documental do Superior Tribunal de Justiça.

**Equipe responsável pela adaptação do MoReq-Jus**

Alexandre Libonati de Abreu	Juiz Federal da Seção Judiciária do Rio de Janeiro
Carlos Leonardo Pires	Superior Tribunal de Justiça
Eduardo César Weber	Diretor da Secretaria de Registros e Informações Processuais – Tribunal Regional Federal da 4ª Região
Giuseppe Dutra Janino	Secretário de Tecnologia da Informação do TSE
Jader Carlos Videira	Diretor da Divisão de Informática dos Juizados Especiais Federais – Tribunal Regional Federal da 3ª Região
Lúcio Melre da Silva	Secretário de Tecnologia da Informação do Supremo Tribunal Federal
Luis Carlos de Freitas	Diretor da Subsecretaria de Informática da Seção Judiciária do Rio de Janeiro
Marlon Borba	Tribunal Regional Federal da 3ª Região
Neide Alves Dias De Sordi	Diretora-Executiva do Departamento de Pesquisa Judiciária do Conselho Nacional de Justiça

Patrícia Reis Longhi	Diretora-Geral da Seção Judiciária do Rio de Janeiro
Paulo Martins Inocêncio	Conselho da Justiça Federal
Rafael Almeida de Paula	Secretário de Tecnologia da Informação do TST
Wilmar Barros de Castro	Coordenador de Gestão Documental do Superior Tribunal de Justiça.
Rejane Soares Canuto	Conselho Nacional de Justiça





Poder Judiciário

*Conselho Nacional de Justiça*

**RESOLUÇÃO Nº 335, DE 29 DE SETEMBRO DE 2020.**

Institui política pública para a governança e a gestão de processo judicial eletrônico. Integra os tribunais do país com a criação da Plataforma Digital do Poder Judiciário Brasileiro – PDPJ-Br. Mantém o sistema PJe como sistema de Processo Eletrônico prioritário do Conselho Nacional de Justiça.

**O PRESIDENTE DO CONSELHO NACIONAL DE JUSTIÇA (CNJ),**  
no uso de suas atribuições legais e regimentais,

**CONSIDERANDO** as diretrizes contidas na Lei nº 11.419/2006, que dispõe sobre a informatização do processo judicial, especialmente o disposto no art. 18, que autoriza a regulamentação pelos órgãos do Poder Judiciário;

**CONSIDERANDO** os benefícios advindos da substituição da tramitação de autos em meio físico pelo meio eletrônico, como instrumento de celeridade e qualidade da prestação jurisdicional;

**CONSIDERANDO** a necessidade de racionalização da utilização dos recursos orçamentários pelos órgãos do Poder Judiciário;

**CONSIDERANDO** as vantagens advindas da adoção de instrumentos tecnológicos que permitam a adequação do funcionamento do Poder Judiciário aos princípios da proteção ambiental;



Poder Judiciário

## *Conselho Nacional de Justiça*

**CONSIDERANDO** a necessidade de se convergir esforços para a adoção de soluções desenvolvidas e compartilhadas pelos próprios tribunais brasileiros, sem a dependência de fornecedores privados, atentando-se para requisitos de segurança e de interoperabilidade e racionalização dos gastos com elaboração e aquisição de softwares e permitindo o emprego desses valores financeiros e de pessoal em atividades mais dirigidas à finalidade do judiciário;

**CONSIDERANDO** a necessidade da implementação de um conceito comunitário, no qual todos os tribunais, independentemente da esfera, possam contribuir com as melhores soluções tecnológicas para aproveitamento comum, e melhor gestão da jurisdição digital, respeitando-se a autonomia dos tribunais;

**CONSIDERANDO** que a criação de plataforma digital do judiciário brasileiro facilitará essa convergência de esforços, gerando identidade única do judiciário nacional;

**CONSIDERANDO** que para a criação dessa plataforma é necessária a implantação de uma política de governança e de gestão unificadas com o estabelecimento de padrões que possam atender à diversidade de soluções existentes em todos os segmentos do Judiciário;

**CONSIDERANDO** que a governança e a gestão devem ser pautadas por padrões que atendam a melhor experiência dos usuários no aspecto da usabilidade e acessibilidade, na velocidade na entrega de novas soluções de forma a atender ao cidadão e a sociedade com maior efetividade;

**CONSIDERANDO** a deliberação do Plenário do CNJ na 318ª Sessão Ordinária, realizada em 22 de setembro de 2020, nos autos do Ato Normativo nº 0007555-97.2020.2.00.0000;



Poder Judiciário

## *Conselho Nacional de Justiça*

**RESOLVE:**

### **CAPÍTULO I DA PLATAFORMA DIGITAL DO JUDICIÁRIO BRASILEIRO**

Art. 1º Fica instituída a política pública para a governança e gestão de processo judicial eletrônico, integrando todos os tribunais do país com a criação da Plataforma Digital do Poder Judiciário Brasileiro – PDPJ-Br, mantendo-se o sistema PJe como sistema de Processo Eletrônico prioritário do Conselho Nacional de Justiça.

Art. 2º A PDPJ-Br tem por objetivo:

I – integrar e consolidar todos os sistemas eletrônicos do Judiciário brasileiro em um ambiente unificado;

II – implantar o conceito de desenvolvimento comunitário, no qual todos os tribunais contribuem com as melhores soluções tecnológicas para aproveitamento comum;

III – estabelecer padrões de desenvolvimento, arquitetura, experiência do usuário (*User Experience* - UX) e operação de software, obedecendo as melhores práticas de mercado e disciplinado em Portaria da Presidência do CNJ; e

IV – instituir plataforma única para publicação e disponibilização de aplicações, microsserviços e modelos de inteligência artificial (I.A.), por meio de computação em nuvem.

Art. 3º A PDPJ-Br funcionará como modelo de convergência, será provida por um repositório (*marketplace*) de soluções que estará disponível para uso por todos os sistemas de processo judicial eletrônico do Poder Judiciário nacional.

Art. 4º A PDPJ-Br adotará obrigatoriamente soluções que abranjam os seguintes conceitos:

I – processo eletrônico em plataforma pública;

II – desenvolvimento comunitário que possibilite o compartilhamento entre todos os segmentos e esferas do Poder Judiciário;



Poder Judiciário

## *Conselho Nacional de Justiça*

III – ampla cobertura de testes, baixo acoplamento, alta coesão, modularização;

IV – microsserviços;

V – computação em nuvem;

VI – autenticação uniformizada;

VII – interoperabilidade;

VIII – portabilidade;

IX – mobilidade;

X – acessibilidade;

XI – usabilidade;

XII – segurança da informação;

XIII – adaptável ao uso de ferramentas de aprendizado de máquina (*machine learning*) e de I.A.;

XIV – otimização de fluxos de trabalhos (*workflow*), padronizando-os sempre que possível;

XV – automação de atividades rotineiras ou sequenciais que possam ser substituídas por ações de sistema;

XVI – incremento da robotização e técnicas disruptivas de desenvolvimento de soluções;

XVII – foco prioritário na redução da taxa de congestionamento dos processos e significativa melhora na qualidade dos serviços prestados;

XVIII – adequação à Lei nº 13.709/2018 (LGPD); e

XIX – utilização preferencial de tecnologias com código aberto (*open source*).

Art. 5º Fica proibida a contratação de qualquer novo sistema, módulo ou funcionalidade privados, mesmo de forma não onerosa, que cause dependência tecnológica do respectivo fornecedor e que não permita o compartilhamento não oneroso da solução na PDPJ-Br.

§ 1º A dependência tecnológica indicada no *caput* diz respeito à hipótese em que o tribunal contratante não tenha direito à propriedade do que for desenvolvido e não



Poder Judiciário

## *Conselho Nacional de Justiça*

tenha direito aos códigos fonte, documentação e quaisquer outros artefatos que venham a ser produzidos.

§ 2º Os tribunais que possuem contratos nas condições previstas no § 1º deste artigo terão prazo fixado em ato normativo próprio para início de projeto-piloto de adequação.

§ 3º O descumprimento da regra prevista no *caput* poderá ensejar:

I – a responsabilização do ordenador de despesas por improbidade administrativa, sem prejuízo da comunicação ao Tribunal de Contas respectivo;

II – apuração de possível responsabilidade disciplinar dos gestores de TIC e da administração do respectivo tribunal.

### **CAPÍTULO II DA POLÍTICA DE GOVERNANÇA E GESTÃO DA PDPJ-Br**

Art. 6º Fica instituída a política de governança e gestão para implantação e sustentação da PDPJ-Br.

Art. 7º A política de governança e gestão da PDPJ-Br será coordenada pelo CNJ, contando, sempre que possível, com a participação de representantes do Poder Judiciário e do Sistema de Justiça, e será regulamentada em ato da Presidência do CNJ.

Art. 8º Qualquer solução pública existente que atenda a todos os requisitos estabelecidos na política de governança e gestão, poderá ser aceita na PDPJ-Br, após aprovação da equipe técnica do CNJ.

§ 1º Em casos excepcionais será permitido o desenvolvimento de soluções tecnológicas que impliquem sobreposição, desde que respeitada as diretrizes da presente Resolução e nos termos previstos no parágrafo único do art. 18 desta Resolução.

§ 2º O descumprimento da regra presente neste artigo poderá ensejar as consequências previstas nos incisos I e II do §3º do art. 5º.

Art. 9º O Ato da Presidência que disciplinar a política de governança e gestão PDPJ-Br deverá estabelecer também os seguintes requisitos para os sistemas:

I – padrões de desenvolvimento, documentação e operação de software;



Poder Judiciário

## *Conselho Nacional de Justiça*

II – padrões de comunicação e interoperabilidade entre sistemas e aplicações;

III – arquitetura de desenvolvimento e de infraestrutura;

IV – padrão de autenticação;

V – permissão para o desenvolvimento compartilhado;

VI – definição dos padrões mínimos de interface, com aplicação dos conceitos de usabilidade, de acessibilidade e de experiência do usuário (*user experience*);

VII – disciplina da plataforma única para os modelos de I.A;

VIII – delimitação dos critérios e serviços para computação em nuvem; e

IX – instituição do desenvolvimento:

a) em microsserviços;

b) com ampla cobertura de testes;

c) com baixo acoplamento e alta coesão; e

d) modularizado.

Parágrafo único. A política de governança e gestão da PDPJ-Br poderá adotar outros requisitos face a evolução tecnológica da plataforma, nos termos disciplinados por ato da Presidência do CNJ.

Art. 10. A política de governança e gestão PDPJ-Br deverá estabelecer também os seguintes requisitos para os dados e documentos:

I – padrões das tabelas unificadas;

II – possibilidade de bases centralizadas ou descentralizadas, sempre uniformes;

III – padrões de dados mínimos para o atendimento da Lei nº 13.709/2018;

IV – padrões de documentos digitais; e

V – padrões de assinaturas digitais.

Art. 11. Caberá ao CNJ definir e coordenar a força-tarefa para o desenvolvimento do portal com interface nacional única para os usuários externos.

Parágrafo único. Todos os sistemas judiciais atuais deverão aderir à solução prevista no *caput*, integrando-a aos seus sistemas como um microsserviço.



Poder Judiciário

## *Conselho Nacional de Justiça*

### **CAPÍTULO III DA GESTÃO DA PDPJ-Br**

Art. 12. Para uma avaliação precisa do estágio atual de desenvolvimento dos sistemas judiciais eletrônicos, o CNJ deverá coordenar e promover as seguintes ações:

I – elaborar censo para identificar os sistemas processuais empregados em todos os tribunais e o grau de adesão ao PJe, com identificação das tecnologias empregadas, práticas de desenvolvimento utilizadas, atividade no repositório, contribuições efetivas de outros tribunais, além de identificar os sistemas processuais onerosos que ainda são empregados; e

II – fixar diretrizes para alinhamento da governança com todos os tribunais.

Art. 13. Para garantir a eficiência operacional da PDPJ-Br o CNJ deverá garantir por meio de monitoramento, indicadores e metas:

I – a agilidade na tramitação dos processos judiciais e administrativos;

II – a razoável duração do processo;

III – a excelência na gestão de custos operacionais;

IV – a economicidade dos recursos por meio da racionalização na aquisição e utilização de todos os materiais, bens e serviços;

V – a responsabilidade ambiental;

VI – melhor alocação dos recursos humanos necessários à prestação jurisdicional, principalmente na área de tecnologia da informação e comunicações (TIC); e

VII – promover e facilitar o acesso à Justiça e ao Poder Judiciário, com o objetivo de democratizar a relação do cidadão com os órgãos judiciais e garantir equidade no atendimento à sociedade.

Art. 14. A PDPJ-Br será hospedada em nuvem, podendo se valer de serviço de computação em nuvem provido por pessoa jurídica de direito privado, inclusive na modalidade de integrador de nuvem (*broker*), desde que observado o seguinte:

I – armazenamento dos dados em *datacenter* abrigado em território nacional;



Poder Judiciário

## *Conselho Nacional de Justiça*

II – cumprimento da Lei Geral de Proteção de Dados Pessoais (Lei nº 13.709/2018);

III – atendimento aos requisitos de disponibilidade, de escalabilidade, de redundância e de criptografia;

IV – capacidade de mensuração de uso dos recursos da nuvem de forma individualizada por cliente de cada serviço provido na PDPJ-Br; e

V – conformidade com as normas técnicas e outras estabelecidas em ato próprio da Presidência do CNJ.

### **CAPÍTULO IV DOS SISTEMAS ATUAIS**

Art. 15. O projeto PJe, coordenado pelo CNJ, face a seu avançado estágio de desenvolvimento aderente a PDPJ-Br, será mantido e aprimorado capitaneando a nova Plataforma.

§ 1º A PDPJ-Br proverá aplicações, módulos e microsserviços, em especial o PJe, por meio do conceito de “nuvem nacional”, para todos os integrantes da rede do Poder Judiciário Nacional que já utilizem ou pretendem utilizar solução em nuvem, na qual estarão centralizadas todas as bases de dados, documentos e aplicações.

§ 2º O CNJ coordenará as ações para contratação e implantação da nuvem nacional.

§ 3º Os custos de processamento de microsserviços específicos de um integrante da rede do Poder Judiciário Nacional, disponibilizados na PDPJ-Br, bem como o armazenamento das bases de dados e documentos de cada tribunal que adotar a PDPJ-Br na nuvem nacional, deverão ser rateados proporcionalmente, considerando, no mínimo a volumetria, dentre outros quesitos.

§ 4º A atual nuvem administrada pelo CNJ será mantida nas mesmas condições e com os mesmos tribunais integrantes, enquanto a nuvem nacional não for implantada.





Poder Judiciário

## *Conselho Nacional de Justiça*

§ 5º As regras para adoção e implantação da nuvem nacional, bem como a utilização e divisão de custos por rateio pelos tribunais serão regulamentadas em ato próprio da Presidência do CNJ.

Art. 16. Os demais projetos de sistema processual público coordenados por outros tribunais também poderão ser mantidos e aprimorados para se adequar a PDPJ-Br, desde que:

I – estejam aderentes a plataforma de interoperabilidade entres os sistemas, definida pela Presidência do CNJ;

II – seja possível a coexistência mediante desenvolvimento colaborativo; e

III – seus novos módulos e evoluções sejam disponibilizados na Plataforma Nacional para permitir a utilização por toda a rede do Poder Judiciário.

Art. 17. Tribunais que não possuem projetos de sistema processual público poderão aderir à PDPJ, inclusive colaborando no desenvolvimento de microsserviços.

Parágrafo único. As melhorias e evoluções dos sistemas de que trata o *caput* terão que ser adequados à PDPJ-Br; e os novos desenvolvimentos deverão tornar-se públicos e compartilhados com todos os tribunais.

Art. 18. O CNJ coordenará a definição de critérios para evolução de sistemas considerando o conceito de desenvolvimento comunitário, bem como monitorará o desenvolvimento e manutenção dos sistemas legados.

Parágrafo único. O CNJ não interferirá no desenvolvimento de soluções tecnológicas pelos tribunais ainda que sobrepostas a outras já existentes, desde que desenvolvidas na forma da presente Resolução e se mostrem justificadas pelas peculiaridades regionais ou pela metodologia de trabalho adotada no órgão.

Art. 19. Os tribunais, juntamente com os coordenadores dos projetos vigentes de sistema de tramitação processual eletrônica, deverão promover ações que possibilitem a troca de informações com os demais sistemas e redução dos custos de TIC com ações isoladas.

## **CAPÍTULO IV DAS DISPOSIÇÕES FINAIS E TRANSITÓRIAS**



Poder Judiciário

## *Conselho Nacional de Justiça*

Art. 20. Esta Resolução será disciplinada por atos próprios da Presidência do CNJ, inclusive quanto aos prazos para as ações da PDPJ-Br.

Art. 21. Os casos omissos que possuam caráter nacional serão resolvidos pela Presidência do Conselho Nacional de Justiça.

Art. 22. Ficam revogados os artigos 34 e seus parágrafos, 44 e seu parágrafo único, 45 e 46 da Resolução CNJ nº 185/2013.

Art. 23. A presente Resolução não se aplica às soluções tecnológicas que não tratam de processo judicial eletrônico ainda que sirvam ao Poder Judiciário.

Art. 24. A Corregedoria Nacional do Conselho Nacional de Justiça e a Corregedoria-Geral da Justiça Federal, a Corregedoria-Geral da Justiça do Trabalho, a Corregedoria-Geral da Justiça Eleitoral e a Corregedoria da Justiça Militar da União deverão incluir em suas rotinas de fiscalização a verificação de cumprimento da presente Resolução.

Art. 25. Esta Resolução entra em vigor na data de sua publicação, revogando-se disposições contrárias.

Ministro **LUIZ FUX**



Poder Judiciário

*Conselho Nacional de Justiça*

**PORTARIA Nº 253, DE 18 DE NOVEMBRO DE 2020.**

Institui os critérios e diretrizes técnicas para o processo de desenvolvimento de módulos e serviços na Plataforma Digital do Poder Judiciário Brasileiro – PDPJ-Br.

**O PRESIDENTE DO CONSELHO NACIONAL DE JUSTIÇA (CNJ)**, no uso de suas atribuições legais e regimentais,

**CONSIDERANDO** as diretrizes contidas na Resolução CNJ nº 335/2020, que institui a política pública para a governança e gestão de sistemas judiciais por meio da Plataforma Digital do Poder Judiciário Brasileiro – PDPJ-Br;

**CONSIDERANDO** que a Plataforma Digital do Poder Judiciário Brasileiro – PDPJ-Br é composta por um conjunto de aplicações e micros serviços mantidos de forma colaborativa por diferentes órgãos e instituições, e que deve funcionar de forma harmônica com o objetivo de padronizar o acesso ao trâmite processual, a troca de dados, a interação entre os diversos sistemas, favorecendo a centralização de acesso por todos;

**CONSIDERANDO** a conveniência da implementação de um conceito comunitário, no qual todos os tribunais, independentemente do segmento de justiça, possam contribuir com as melhores soluções tecnológicas em benefício comum, assegurando uma melhor gestão da jurisdição digital e respeitando-se a autonomia dos tribunais;



Poder Judiciário

## *Conselho Nacional de Justiça*

**CONSIDERANDO** o papel fundamental da implantação de uma política de governança e de gestão unificadas com o estabelecimento de padrões que possam atender à diversidade de soluções existentes em todos os segmentos do judiciário para o bom funcionamento da Plataforma;

**CONSIDERANDO** que a governança e a gestão devem ser pautadas por padrões que busquem a melhor experiência dos usuários no aspecto da usabilidade e acessibilidade, na velocidade da entrega de novas soluções de forma a atender ao cidadão e a sociedade com maior efetividade;

### **RESOLVE:**

Art. 1º Instituir os critérios e diretrizes técnicas para o processo de desenvolvimento de módulos e serviços na Plataforma Digital do Poder Judiciário Brasileiro – PDPJ-Br.

Art. 2º A PDPJ-Br fornecerá padrões de *Application Programming Interface* (API), de modelos de dados e domínio, de eventos e mensagens, com os respectivos documentos de referência, que permitam desenvolvedores criarem, manterem e evoluírem as aplicações que compõem a Plataforma.

Art. 3º O processo de disponibilização de soluções para a Plataforma será comunitário e descentralizado, podendo dele participar quaisquer tribunais brasileiros, órgãos públicos externos e, nos termos desta Portaria, particulares.

§ 1º Os tribunais e órgãos externos, bem como os particulares admitidos a colaborar com a Plataforma, indicarão livremente os desenvolvedores e analistas que participarão dos projetos, os quais terão acesso às ferramentas centralizadas de controle de demandas e de versionamento de código e arquivos do CNJ.



Poder Judiciário

## *Conselho Nacional de Justiça*

§ 2º Os profissionais serão identificados por órgão e área de atuação (negocial ou técnica), devendo-se adotar mecanismos de reputação, tendo em vista a efetiva participação, quantidade e qualidade das contribuições dos participantes, a fim de criar incentivos positivos da participação comunitária.

Art. 4º O CNJ disponibilizará em sítio próprio (*marketplace*) a lista de projetos em andamento, concluídos, e pendentes de serem integrados à PDPJ-Br.

§ 1º O tribunal que desejar colaborar com o desenvolvimento de algum dos serviços e soluções, ou disponibilizar serviço ou solução por ele já desenvolvida, poderá formalizar seu interesse por *e-mail* dirigido ao endereço eletrônico <gerenciaexecutivapdpj@cnj.jus.br>, devendo ser realizada, na sequência, reunião inaugural (*kickoff*) para alinhamento de expectativas.

§ 2º O CNJ acompanhará a execução dos projetos em desenvolvimento, por meio da designação de líder técnico, que será o responsável por orientar os tribunais acerca dos critérios técnicos a serem seguidos para o desenvolvimento de cada solução, conforme as diretrizes desta Portaria.

§ 3º O andamento do projeto também será acompanhado pelo Escritório de Projetos do Departamento de Tecnologia da Informação e Comunicação do CNJ.

Art. 5º Os serviços e aplicações integrados à Plataforma serão classificados da seguinte forma:

I – serviços estruturantes: serviços que implementam as funcionalidades essenciais básicas para um sistema de processo judicial de tramitação eletrônica, bem como àqueles serviços necessários à integração, à coreografia e à interoperabilidade entre os serviços e soluções que compõe a Plataforma;

II – serviços negociais: serviços que implementam necessidade de negócio relevante para a tramitação de processo judicial eletrônico e sistemas judiciais, tais como distribuição de processos, controle de custas, comunicação de atos, controle de agendamento de audiências, central de mandados, dentre outros;



Poder Judiciário

## *Conselho Nacional de Justiça*

III – serviços de integração com sistemas externos: serviços que fazem interface com sistemas, serviços e/ou aplicações externas ao Poder Judiciário, como o sistema de penhora *on-line* fornecido em parceria com o Banco Central (Sisbajud), o sistema de envio eletrônico de correspondências pela Empresa Brasileira de Correios e Telégrafos (eCARTA), dentre outros de especial interesse à prestação do serviço jurisdicional; e

IV – soluções e aplicações da comunidade externa ao judiciário: serviços desenvolvidos por entes externos ao judiciário voltados a atender às suas necessidades, adotando padrões de API que se integrem à PDPJ-Br mediante chancela do Poder Judiciário.

Art. 6º A PDPJ-Br seguirá modelo arquitetural de microsserviços, preferencialmente modelados utilizando-se a metodologia *Domain Driven Design (DDD)*.

Parágrafo único. A interação entre os microsserviços da Plataforma adotará o modelo de coreografia, em que cada microsserviço deve possuir inteligência sobre seu próprio domínio, com alto nível de desacoplamento, interagindo com os demais microsserviços por meio de serviço centralizado de mensageria (*Message Broker*).

Art. 7º O módulo ou microsserviço que se assemelhar a uma ou mais funcionalidades presentes nos sistemas processuais, deverá observar a necessidade de decomposição da aplicação monolítica existente.

§ 1º O trabalho de decomposição começará pela identificação do domínio a ser extraído, utilizando-se estratégias baseadas em *DDD*. Identificado o domínio, será iniciada a modelagem por meio de histórias construídas em conjunto por desenvolvedores e especialistas do domínio (*Domain Experts*).

§ 2º Finalizada a modelagem de domínio, deverá ser iniciada a modelagem da API, que será implementada ainda dentro da aplicação monolítica, mas com vistas à implementação do microsserviço a ser desenvolvido. A criação da API servirá para delinear a construção do microsserviço fora da aplicação monolítica.

§ 3º A aplicação monolítica precisará ser adaptada para utilização da nova API, ao invés de mecanismos nativos, por exemplo, JSF, Richfaces, componentes Seam, etc.



Poder Judiciário

## *Conselho Nacional de Justiça*

§ 4º O modelo relacional precisará ser desacoplado com a extinção de *constraints* baseadas em chaves e os dados deverão ser migrados para o novo banco de dados do microserviço.

§ 5º O CNJ proverá uma série de microserviços denominados módulos estruturantes, que já disporão de diversas APIs comuns ao relacionamento com o processo judicial, com vistas a não ser necessária a incorporação de diversas APIs nos sistemas monolíticos legados.

Art. 8º Os registros dos microserviços na PDPJ-Br ocorrerão no *Service Discovery (SD)* do CNJ, no qual um SD cataloga os serviços que compõem a Plataforma, registrando o nome da instância, o serviço ao qual ela responde e seu endereço de rede.

Parágrafo único. O SD é implementado utilizando a ferramenta *Netflix Eureka*, através da implementação do *Spring Cloud Netflix*.

Art. 9º Os microserviços deverão ser versionados utilizando o sistema de controle de versões Git.

§ 1º Os repositórios deverão ser hospedados na Plataforma disponibilizada pelo CNJ.

§ 2º A criação do repositório em seu respectivo grupo deve ser efetuada pela gestão da Divisão do Processo Judicial Eletrônico Nacional (DPJE).

§ 3º A partir da criação do repositório serão atribuídas as permissões de acesso e controle aos líderes técnicos de cada projeto.

Art. 10. Deverá ser adotada, preferencialmente, a linguagem de programação Java, utilizando-se o framework *Spring* com *Springboot*, utilizando como referência o projeto *Sample Service*, disponível no endereço eletrônico <<https://git.cnj.jus.br/pje2/pje2-servicos/sample-service>> como *template*.

Parágrafo único. Na utilização de outra linguagem de programação ou *framework* de desenvolvimento, a solução deverá atender a todos os padrões de desenvolvimento descritos nesta Portaria.



Poder Judiciário

## Conselho Nacional de Justiça

Art. 11. O acesso aos microsserviços deverá ser protegido com mecanismos de autenticação e autorização baseado em *OAuth2* a ser provido pelo serviço de *Single Sign-On (SSO)* da PDPJ-Br permitindo a interligação às diversas bases de usuários de diferentes sistemas de processo eletrônico e utilizando conceito de federação de bases pela implementação de provedores de acesso customizados.

§ 1º Todos os provedores implementados com o propósito de integração da PDPJ-Br ao PJe estão versionados no Git por meio do projeto PJe-KC.

§ 2º Os demais sistemas processuais deverão desenvolver seus próprios provedores de acesso, para integração ao serviço unificado de autenticação (SSO).

Art. 12. Os microsserviços desenvolvidos na PDPJ-Br precisam ser coesos e guardar a característica *stateless*, possibilitando maior escalabilidade em ambiente voltado à contêineres, como o Kubernetes.

Parágrafo único. A escalabilidade deverá ter elasticidade em padrão horizontal, permitindo que réplicas do mesmo microsserviço possam ser instanciadas ou desmobilizadas de forma a atender mais adequadamente a demanda pelo respectivo serviço.

Art. 13. Deverão ser utilizados recursos de *Continuous Integration (CI)*, permitindo execução de *pipelines*, testes automatizados e registro no repositório de imagens.

Parágrafo único. Sempre que possível deve ser adotada a prática de *Continuous Delivery (CD)*, quando compatível com a natureza específica de cada projeto.

Art. 14. As soluções presentes na PDPJ-Br deverão conter artefatos de testes automatizados com incentivo às práticas de *Test Driven Development (TDD)*, contando com testes de unidade e de integração.

Parágrafo único. É obrigatória a produção de relatório de cobertura de testes automatizados, o qual deverá ser atualizado a cada execução do *pipeline* de integração contínua.

Art. 15. A documentação de cunho negocial e técnico deverá ser produzida para cada solução presente na PDPJ-Br, e os microsserviços também terão que prover documentação técnica referente à API disponibilizada, utilizando o padrão OpenAPI 3.0.





Poder Judiciário

## *Conselho Nacional de Justiça*

Art. 16. Os microsserviços deverão prover APIs correspondentes a seu contexto de negócio ou funcional.

§ 1º As APIs deverão ser providas por meio de tecnologia REST e obedecer aos padrões definidos na Plataforma de documentações do CNJ, disponível em <<https://docs.pje.jus.br>>.

§ 2º A definição do *design* da interface das APIs, observando as melhores práticas e buscando atender o maior número de situações e contextos em que o microsserviço pode ser utilizado, deve constituir entrega própria e prioritária.

Art. 17. As soluções baseadas em microsserviços deverão empregar o mecanismo de *message brokering*, por meio do qual trafegam as mensagens com os eventos propagados pelos microsserviços, provocando comportamentos e ações de acordo com os contextos de cada serviço, bem como propiciando a declaração de filas para recebimento de mensagens relevantes produzidas por outros serviços.

Parágrafo único. O formato de comunicação entre os microsserviços por meio de mensageria dar-se-á conforme documentação provida pela Plataforma de documentações do CNJ.

Art. 18. As APIs disponibilizadas pelos microsserviços serão acessadas através de um serviço de borda denominado *gateway*, que é utilizado para receber requisições dos diversos clientes de API e direcionar essas requisições aos serviços responsáveis pelo processamento.

Art. 19. A identidade visual dos módulos desenvolvidos dentro da PDPJ-Br deverá adotar o conjunto de ferramentas desenvolvido pelo CNJ, denominada de UIKit, com a finalidade de padronizar a interface gráfica dos módulos, uniformizando os aspectos da usabilidade, acessibilidade e experiência do usuário.

Art. 20. As soluções da Plataforma podem utilizar recursos computacionais próprios em ambiente exclusivo ao de desenvolvimento mantido diretamente pelos órgãos e instituições aderentes ou contratados em serviço de nuvem computacional.



Poder Judiciário

## *Conselho Nacional de Justiça*

§ 1º No caso de utilização de nuvem computacional de terceiros deve-se observar os princípios e diretrizes relacionados à Segurança da Informação para o tratamento da informação em ambiente de Computação em Nuvem dispostos na Portaria CNJ nº 9/2018.

§ 2º A nuvem de serviço utilizará a infraestrutura baseada em containerização de código aberto que permitirá a criação e uso de contêineres, bem como a orquestração desses contêineres sendo gerenciada por um sistema também de código aberto.

Art. 21. Soluções que utilizam Inteligência Artificial, deverão estar de acordo com o disposto nos termos da Resolução CNJ nº 332/2020, que disciplina a ética, a transparência e a governança na produção e no uso de Inteligência Artificial no Poder Judiciário e dá outras providências.

Art. 22. Cada tribunal ou ente participante do ecossistema de Justiça, para desenvolvimento de módulos ou usufruto dos serviços disponibilizados pela Plataforma, deverá assinar Termo de Cooperação Técnica, conforme modelo estabelecido pela Gerência Executiva da PDPI-Br.

Art. 23. Cumprirá ao Departamento de Tecnologia da Informação e Comunicação do CNJ a responsabilidade de prover a manutenção e detalhamento dessas diretrizes técnicas em portal próprio, de modo a atender plenamente o disposto nesta Portaria.

Art. 24. Esta Portaria entra em vigor na data da sua publicação.

A handwritten signature in blue ink, appearing to read 'Luiz Fux', is positioned above the name of the minister.

Ministro **LUIZ FUX**



Poder Judiciário

## *Conselho Nacional de Justiça*

### **RESOLUÇÃO Nº 396, DE 7 DE JUNHO DE 2021.**

Institui a Estratégia Nacional de Segurança Cibernética do Poder Judiciário (ENSEC-PJ).

**O PRESIDENTE DO CONSELHO NACIONAL DE JUSTIÇA (CNJ)**, no uso de suas atribuições legais e regimentais,

**CONSIDERANDO** que compete ao CNJ a atribuição de coordenar o planejamento e a gestão estratégica de Tecnologia da Informação e Comunicação (TIC) do Poder Judiciário;

**CONSIDERANDO** que é imprescindível garantir a segurança cibernética do ecossistema digital do Poder Judiciário brasileiro;

**CONSIDERANDO** os termos da Resolução CNJ nº 370/2021, que institui a Estratégia Nacional de Tecnologia da Informação e Comunicação do Poder Judiciário (ENTIC-JUD) e estabelece as diretrizes para sua governança, gestão e infraestrutura;

**CONSIDERANDO** a importância de se estabelecer objetivos, princípios e diretrizes de Segurança da Informação alinhados às recomendações constantes da norma NBR ISO/IEC 27001:2013, que trata da segurança da informação;

**CONSIDERANDO** o que dispõe a Lei nº 13.709/2018, com a redação dada pela Lei nº 13.853/2019, sobre a proteção de dados pessoais, que altera a Lei - nº 12.965/2014 (Marco Civil da Internet);

**CONSIDERANDO** o disposto na Resolução CNJ nº 291/2019, que instituiu o Sistema Nacional de Segurança do Poder Judiciário;

**CONSIDERANDO** o disposto na Portaria CNJ nº 242/2020, que institui o Comitê de Segurança Cibernética do Poder Judiciário;

**CONSIDERANDO** o disposto na Portaria CNJ nº 249/2020, que designa os integrantes do Comitê de Segurança Cibernética do Poder Judiciário (CSCPJ);



Poder Judiciário

## *Conselho Nacional de Justiça*

**CONSIDERANDO** que, para contemplar aspectos fundamentais para o desenvolvimento da Política sobre a área da Segurança Cibernética, será necessário abordar aspectos da Segurança da Informação, área sistêmica e mais abrangente,

**CONSIDERANDO** a deliberação do Plenário do CNJ no Ato Normativo nº 0003201-92.2021.2.00.0000, na 87ª Sessão Virtual, realizada em 28 de maio de 2021;

**RESOLVE:**

### CAPÍTULO I DISPOSIÇÕES GERAIS

Art. 1º o Instituir a Estratégia Nacional de Segurança da Informação e Cibernética do Poder Judiciário (ENSEC-PJ), no âmbito dos órgãos do Poder Judiciário, à exceção do Supremo Tribunal Federal (STF).

Parágrafo único. A ENSEC-PJ prevista nesta Resolução contempla:

- I – temas relacionados à segurança da informação, de forma ampla, que sejam essenciais para segurança cibernética;
- II – segurança física e proteção de dados pessoais e institucionais, nos aspectos relacionados à cibersegurança;
- III – segurança física e proteção de ativos de tecnologia da informação de forma geral;
- IV – ações destinadas a assegurar a disponibilidade, a integridade, a confidencialidade e a autenticidade de dados e de informações;
- V – ações destinadas a assegurar o funcionamento dos processos de trabalho, a continuidade operacional e a continuidade das atividades fim e administrativas dos órgãos do Poder Judiciário;
- VI – ações de planejamento, de sistematização e de normatização sobre temas atinentes à segurança cibernética;
- VII – ações de comunicação, de conscientização, de formação de cultura e de direcionamento institucional com vistas à segurança cibernética; e
- VIII – ações de formação acadêmica, formação técnica, qualificação e reciclagem de profissionais de tecnologia da informação e comunicação que atuam na área de segurança cibernética.



Poder Judiciário

## *Conselho Nacional de Justiça*

### CAPÍTULO II

#### OBJETIVOS DA ESTRATÉGIA NACIONAL DE SEGURANÇA CIBERNÉTICA DO PODER JUDICIÁRIO (ENSEC-PJ)

Art. 2º A ENSEC-PJ tem o objetivo de aprimorar o nível de maturidade em segurança cibernética nos órgãos do Poder Judiciário, abrangendo os aspectos fundamentais da segurança da informação para o aperfeiçoamento necessário à consecução desse propósito.

Art. 3º Para a concretização dos objetivos da segurança cibernética instituídos na Política de Segurança Cibernética do Poder Judiciário (PSEC-PJ), estrutura-se a presente Estratégia Nacional de Segurança Cibernética com visão, objetivos e ações capazes de conduzir os órgãos do Poder Judiciário a um ambiente desenvolvido, resistente e seguro.

### CAPÍTULO III

#### DA VISÃO, DOS OBJETIVOS E DAS AÇÕES

Art. 4º A visão da ENSEC-PJ consiste em alcançar a excelência em segurança cibernética no Poder Judiciário.

Art. 5º Os objetivos da ENSEC-PJ são a base para tornar o espaço cibernético mais confiável, resistente, inclusivo e seguro e visam direcionar as ações dos órgãos do Poder Judiciário na área de segurança cibernética.

Art. 6º São objetivos da ENSEC-PJ:

I – tornar o Judiciário mais seguro e inclusivo no ambiente digital;

II – aumentar a resiliência às ameaças cibernéticas;

III – estabelecer governança de segurança cibernética e fortalecer a gestão e coordenação integrada de ações de segurança cibernética nos órgãos do Poder Judiciário; e

IV – permitir a manutenção e a continuidade dos serviços, ou o seu restabelecimento em menor tempo possível.

Art. 7º As ações da ENSEC-PJ foram estabelecidas com a finalidade de possibilitar o alcance dos objetivos e basearam-se no estágio de maturidade geral dos órgãos do Poder Judiciário.



Poder Judiciário

## *Conselho Nacional de Justiça*

Art. 8º Os órgãos do Poder Judiciário, com exceção do STF, devem colocar em prática as ações para o pleno alcance dos objetivos da ENSEC-PJ.

Parágrafo único. O engajamento da alta administração de cada tribunal é essencial para a consecução das finalidades e das medidas de proteção ao serviço, sobretudo quando implicarem a necessidade de rápida suspensão do acesso ao público, para evitar o alastramento de ataque cibernético e conter os danos.

Art. 9º São ações da ENSEC-PJ:

- I – fortalecer as ações de governança cibernética;
- II – elevar o nível de segurança das infraestruturas críticas;
- III – estabelecer rede de cooperação do Judiciário para a segurança cibernética; e
- IV – estabelecer modelo centralizado de governança cibernética nacional.

Art. 10. Para fortalecer as ações de governança cibernética, deve-se estabelecer um Sistema de Gestão em Segurança da Informação baseado em riscos, de acordo com recomendação do CNJ.

Art. 11. Para elevar o nível de segurança das infraestruturas críticas, deve-se:

- I – estabelecer todas as ações que possibilitem maior eficiência, ou seja, capacidade de responder de forma satisfatória a incidentes de segurança, permitindo a contínua prestação dos serviços essenciais a cada órgão;
- II – instituir e manter Equipe de Tratamento e Resposta a Incidentes de Segurança Cibernética (ETIR);
- III – elaborar e aplicar processo de resposta e tratamento a incidentes de segurança cibernética que contenha, entre outros, procedimento de continuidade do serviço prestado e seu rápido restabelecimento, além de comunicação interna e externa;
- IV – utilizar tecnologia que possibilite a análise consolidada dos registros de auditorias coletados em diversas fontes de ativos de informação e de ações de usuários, permitindo automatizar ações de segurança e oferecer inteligência à análise de eventos de segurança;
- V – utilizar tecnologia que permita a inteligência em ameaças cibernéticas em redes de informação; especialmente em fóruns, inclusive da iniciativa privada e comunidades virtuais da internet;
- VI – providenciar a realização de cópias de segurança atualizadas e segregadas de forma automática em local protegido, em formato que permita a investigação de incidentes;



Poder Judiciário

## *Conselho Nacional de Justiça*

VII – elaborar requisitos específicos de segurança cibernética relativos aos ativos sob sua jurisdição, incluindo ambientes centralizados, endpoints, equipamentos intermediários ou finais conectados em rede ou a algum sistema de comunicação, inclusive computadores portáteis e telefones celulares;

VIII – elaborar requisitos específicos de segurança cibernética relacionados com o trabalho remoto;

IX – adotar práticas e requisitos de segurança cibernética no desenvolvimento de novos projetos, tais como dupla verificação do acesso externo;

X – realizar, ao menos semestralmente, avaliação e testes de conformidade em segurança cibernética de forma a aferir a eficácia dos controles estabelecidos;

XI realizar prática em gestão de incidentes e efetivar o aprimoramento contínuo do processo; e

XII – estabelecer troca de informações e boas práticas com outros membros do poder público em geral e do setor privado com objetivo colaborativo.

### CAPÍTULO IV

#### DO MODELO CENTRALIZADO DE GOVERNANÇA NACIONAL NA SEGURANÇA CIBERNÉTICA DO PODER JUDICIÁRIO

Art. 12. O modelo centralizado de governança nacional na segurança cibernética do Poder Judiciário tem os seguintes objetivos:

I – promover a coordenação dos diversos entes relacionados com a segurança cibernética;

II – possibilitar a análise conjunta do nível de maturidade em segurança cibernética nos órgãos do Poder Judiciário;

III – estabelecer e desenvolver padrão de maturidade unificado de segurança cibernética, de forma que seja possível avaliar o nível de maturidade de cada órgão do Judiciário, por meio de indicadores estabelecidos;

IV – estabelecer rotinas de verificações de conformidade em segurança cibernética; e

V – possibilitar a convergência de esforços e iniciativas na apuração de incidentes e na promoção de ações de capacitação e educação em segurança cibernética.

Art. 13. O CNJ coordenará as ações para viabilizar a governança nacional em segurança cibernética do Poder Judiciário.



Poder Judiciário

## *Conselho Nacional de Justiça*

### CAPÍTULO V

#### DO COMITÊ GESTOR DE SEGURANÇA DA INFORMAÇÃO DO PODER JUDICIÁRIO

Art. 14. Fica instituído o Comitê Gestor de Segurança da Informação do Poder Judiciário (CGSI-PJ), com atribuição de assessorar o CNJ nas atividades relacionadas à segurança da informação.

Art. 15. Integram o CGSI-PJ:

- I – dois especialistas representantes do Conselho Nacional de Justiça;
- II – dois especialistas representantes do Supremo Tribunal Federal;
- III – um especialista representante do Superior Tribunal de Justiça;
- IV – um especialista representante do Tribunal Superior Eleitoral;
- V – um especialista representante do Tribunal Superior do Trabalho;
- VI – um especialista representante do Conselho Superior da Justiça do Trabalho;
- VII - um especialista representante do Conselho da Justiça Federal;
- VIII – um especialista representante do Superior Tribunal Militar; e
- IX – dois especialistas representantes dos Tribunais de Justiça Estaduais.

§ 1º O CGSI-PJ será coordenado por um representante do Conselho Nacional de Justiça designado pela Presidência.

§ 2º As indicações dos representantes dos incisos I e IX serão feitas pela Presidência do CNJ.

§ 3º O CGSI-PJ poderá convidar representantes de órgãos de segurança pública, do Ministério Público, das Forças Armadas e especialistas técnicos de outros órgãos públicos ou privados que pretendam subsidiar os respectivos trabalhos.

§ 4º Os integrantes do CGSI-PJ deverão ter conhecimento técnico na área de segurança da informação.

Art. 16. O CGSI-PJ se reunirá, em caráter ordinário, semestralmente, e, em caráter extraordinário, por convocação de seu coordenador.

Art. 17. Compete ao CGSI-PJ, assessorando o CNJ, nos temas relacionados à segurança da informação:





Poder Judiciário

## *Conselho Nacional de Justiça*

- I – estabelecer norma sobre a definição dos requisitos metodológicos para a implementação da gestão de risco dos ativos da informação no Poder Judiciário;
- II – aprovar políticas, diretrizes, estratégias, normas e recomendações relacionadas à segurança da informação no Poder Judiciário;
- III – elaborar e implementar programas sobre segurança da informação destinados à conscientização e à capacitação dos servidores do Poder Judiciário;
- IV – estabelecer critérios que permitam monitorar e avaliar a execução da PSEC-PJ e de seus instrumentos, bem como o nível de maturidade em segurança da informação em cada órgão do Poder Judiciário;
- V – estabelecer norma de criação e funcionamento do Centro de Prevenção, Tratamento e Resposta a Incidentes Cibernéticos do Poder Judiciário (CPTRIC-PJ), que funcionará como canal oficial de ações preventivas e corretivas, em caso de ameaças ou de ataques cibernéticos; e
- VI – promover troca de informações e experiências com os comitês gestores de segurança da informação dos outros Poderes e com a sociedade.

### CAPÍTULO VI

#### DA REDE NACIONAL DE COOPERAÇÃO DO PODER JUDICIÁRIO NA ÁREA DE SEGURANÇA CIBERNÉTICA

Art. 18. A Rede de Cooperação do Judiciário na área de segurança cibernética tem os seguintes objetivos:

- I – promover ambiente participativo, colaborativo e seguro entre os órgãos do Poder Judiciário, por meio do acompanhamento contínuo e proativo das ameaças e dos ataques cibernéticos;
- II – estimular o compartilhamento de informações sobre incidentes e vulnerabilidades cibernéticas;
- III – realizar exercícios cibernéticos com a participação de múltiplos entes;
- IV – fortalecer o Centro de Prevenção, Tratamento e Resposta a Incidentes Cibernéticos (CPTRIC-PJ) do CNJ;
- V – aperfeiçoar a estrutura judiciária para o aprimoramento de investigações de crimes cibernéticos;



Poder Judiciário

## *Conselho Nacional de Justiça*

VI – incentivar a criação e a atuação de ETIR em cada órgão do Poder Judiciário;

VII – emitir alertas e recomendações de segurança cibernética; e

VIII – ampliar parceria com outros órgãos do Poder Executivo, do Poder Legislativo, do Ministério Público, da polícia judiciária, do setor privado e do meio acadêmico, com vistas a elevar, de modo geral, o nível de segurança cibernética.

Parágrafo único. Para fins de cumprimento dos objetivos estabelecidos, todos os órgãos do Judiciário que detectarem incidentes de segurança cibernética deverão reportá-los ao CPTRIC-PJ.

Art. 19. Compete à alta administração dos órgãos do Poder Judiciário, com exceção do STF, realizar a governança da segurança da informação e especialmente:

I – implementar, no que lhe couber, a Política de Segurança Cibernética do Poder Judiciário;

II – elaborar a Política de Segurança da Informação e normas internas correlatas ao tema, observadas as normas de segurança da informação editadas pelo CNJ;

III – destinar recursos orçamentários específicos para as ações de segurança da informação;

IV – promover ações de capacitação e profissionalização dos recursos humanos em temas relacionados à segurança da informação;

V – instituir e implementar ETIR, que comporá a rede de equipes vinculadas ao CPTRIC-PJ;

VI – coordenar e executar as ações de segurança da informação no âmbito de sua atuação;

e

VII – aplicar as ações corretivas e disciplinares cabíveis nos casos de violação da segurança da informação.

Art. 20. Cada órgão do Poder Judiciário, com exceção do STF, deverá constituir CGSI, ao qual caberá:

I – assessorar a alta administração do órgão do Poder Judiciário em todas as questões relacionadas à segurança da informação;

II – propor alterações na política de segurança da informação e deliberar sobre assuntos a ela relacionados, incluindo atividades de priorização de ações e gestão de riscos de segurança;



Poder Judiciário

## *Conselho Nacional de Justiça*

III – propor normas internas relativas à segurança da informação;

IV – constituir grupos de trabalho para tratar de temas e propor soluções específicas sobre segurança da informação; e

V – consolidar e analisar os resultados dos trabalhos de auditoria sobre a gestão da segurança da informação.

§ 1º O CGSI será coordenado pela autoridade responsável pela segurança da informação no respectivo órgão do Poder Judiciário, nomeado por seu presidente.

§ 2º Os órgãos do Poder Judiciário, com exceção do STF, editarão atos para definir a forma de funcionamento dos respectivos CGSIs, observado o disposto nesta Resolução e na legislação de regência.

Art. 21. Cada órgão do Poder Judiciário, com exceção do STF, deverá constituir estrutura de segurança da informação, subordinada diretamente à alta administração do órgão e desvinculada da área de TIC.

§ 1º O titular da estrutura prevista no caput deste artigo será o gestor de segurança da informação do órgão.

§ 2º O gestor de segurança da informação terá as seguintes atribuições:

I – instituir e gerir o Sistema de Gestão de Segurança da Informação;

II – implementar controles internos fundamentados na gestão de riscos da segurança da informação;

III – planejar a execução de programas, de projetos e de processos relativos à segurança da informação com as demais unidades do órgão;

IV – implantar procedimento de tratamento e resposta a incidentes em segurança da informação; e

V – observar as normas e os procedimentos específicos aplicáveis em consonância com os princípios e as diretrizes desta Resolução e da legislação de regência. .

## CAPÍTULO VII

### DA POLÍTICA DE SEGURANÇA CIBERNÉTICA DO PODER JUDICIÁRIO

Art. 22. A PSEC-PJ tem a finalidade de prover os princípios, objetivos e instrumentos capazes de assegurar a Segurança Cibernética no Poder Judiciário.



Poder Judiciário

## *Conselho Nacional de Justiça*

Art. 23. São princípios da PSEC-PJ:

I – segurança jurídica;

II – respeito e promoção dos direitos humanos e das garantias fundamentais, em especial a liberdade de expressão, a proteção de dados pessoais, a proteção de privacidade e o acesso à informação;

III – visão abrangente e sistêmica da segurança cibernética;

IV – integração, cooperação e intercâmbio científico e tecnológico relacionado à segurança cibernética entre os órgãos da Administração Pública Federal e do meio acadêmico;

V – educação e inovação como alicerce fundamental para o fomento da cultura em segurança cibernética;

VI – orientação à gestão de riscos e à gestão da segurança da informação;

VII – prevenção, tratamento e resposta a incidentes cibernéticos;

VIII – articulação entre as ações de segurança cibernética e de proteção de dados e ativos de informação; e

IX – garantia ao sigilo das informações imprescindíveis à segurança da sociedade e do Estado e inviolabilidade da vida privada, da honra e da imagem das pessoas.

Art. 24. São objetivos da PSEC-PJ:

I – contribuir para a segurança do indivíduo, da sociedade e do Estado, por meio de ações de segurança cibernética, observados os direitos e as garantias fundamentais;

II – fomentar as atividades de pesquisa científica, de desenvolvimento tecnológico e de inovação relacionadas à segurança cibernética;

III – aprimorar continuamente o arcabouço normativo relacionado à segurança cibernética;

IV – fomentar a formação e a qualificação dos recursos humanos necessários à área de segurança cibernética;

V – fortalecer a cultura de segurança cibernética no âmbito do Poder Judiciário;

VI – aprimorar o nível de maturidade em segurança cibernética no Poder Judiciário;

VII – orientar ações relacionadas:

a) à gestão em segurança da informação;

b) à segurança da informação das infraestruturas críticas;



Poder Judiciário

## *Conselho Nacional de Justiça*

- c) ao tratamento das informações com restrições de acesso;
- d) à proteção dos dados pessoais e dos dados pessoais sensíveis, em conformidade com legislação específica;
- e) à prevenção, ao tratamento e à resposta a incidentes cibernéticos;
- f) à gestão e operação de equipe de tratamento e resposta a incidentes cibernéticos;
- g) ao estabelecimento dos níveis de maturidade em segurança cibernética; e
- h) ao estabelecimento de processo transparente de comunicação e respostas a incidentes entre o poder público e a sociedade.

Art. 25. São instrumentos da PSEC-PJ:

I – a Estratégia Nacional de Segurança Cibernética do Poder Judiciário (ENSEC-PJ);

II – o Protocolo de Prevenção de Incidentes Cibernéticos no âmbito do Poder Judiciário (PPINC-PJ);

III – o Protocolo de Gerenciamento de Crises Cibernéticas no âmbito do Poder Judiciário (PGCC-PJ);

IV – o Protocolo de Investigação para Ilícitos Cibernéticos no âmbito do Poder Judiciário (PIILC-PJ).

§ 1º Os protocolos previstos neste artigo deverão ser revisados sempre que necessário, por ato do Presidente do CNJ.

§ 2º Além dos protocolos previstos nesta Resolução, serão aprovados por ato do Presidente do CNJ os Manuais de Referência para o gerenciamento, controle e padrões necessários ao aperfeiçoamento da segurança cibernética.

Art. 26. Todos os órgãos do Poder Judiciário, à exceção do STF, deverão adotar e seguir, além dos Manuais de Referência para o gerenciamento, controle e padrões necessários ao aperfeiçoamento da segurança cibernética, o PPINC-PJ, que deverá contemplar um conjunto de diretrizes para a prevenção a incidentes cibernéticos em seu mais alto nível; o PGCC-PJ, objetivando contribuir para a resiliência corporativa por meio de resposta, tão célere e eficiente quanto possível, a incidentes em que os ativos de informação do Poder Judiciário tenham a sua integridade, confidencialidade ou disponibilidade comprometidos em larga escala ou por longo período; e o PIILC-PJ, com a finalidade de estabelecer os procedimentos básicos para coleta e preservação de evidências, bem como



Poder Judiciário

## *Conselho Nacional de Justiça*

para comunicar fatos penalmente relevantes aos órgãos de investigação e com atribuição para o início da persecução penal.

Parágrafo único. O PGCC-PJ complementa o PPINC-PJ e prevê as ações responsivas a serem colocadas em prática quando ficar evidente que um incidente de segurança cibernética não será mitigado rapidamente e poderá durar por tempo indeterminado.

Art. 27. Considerado o incidente como crise cibernética, o Comitê de Crise deverá ser acionado, nos termos do Protocolo de Gerenciamento de Incidentes e de Crises Cibernéticas.

Art. 28. Cada tribunal, com exceção do STF, deverá estabelecer em sua Política de Segurança da Informação ações para:

- I – realizar a Gestão dos Ativos de Informação e da Política de Controle de Acesso;
- II – criar controles para o tratamento de informações com restrição de acesso;
- III – promover treinamento contínuo e certificação internacional dos profissionais diretamente envolvidos na área de segurança cibernética;
- IV – estabelecer requisitos mínimos de segurança cibernética nas contratações e nos acordos que envolvam a comunicação com outros órgãos;
- V – utilizar os recursos de soluções de criptografia, ampliando o uso de assinatura eletrônica, conforme legislações específicas; e
- VI – comunicar e articular as ações de segurança da informação com a alta administração do órgão.

### CAPÍTULO VIII

#### DA GESTÃO DE USUÁRIOS

Art. 29. Cada órgão do Poder Judiciário, com exceção do STF, deverá implementar a gestão de usuários de sistemas informatizados composta de:

- I – gerenciamento de identidades;
- II – gerenciamento de acessos; e
- III – gerenciamento de privilégios.

Parágrafo único. A gestão de usuários será disciplinada por ato do Presidente do CNJ, que definirá o padrão a ser adotado para utilização de credenciais de login único e



Poder Judiciário

## *Conselho Nacional de Justiça*

interface de interação dos sistemas, com o objetivo de uniformizar e garantir a experiência única de interação com os sistemas judiciais.

### CAPÍTULO IX

#### DA POLÍTICA DE CULTURA E EDUCAÇÃO EM SEGURANÇA CIBERNÉTICA

Art. 30. Fica instituída, no âmbito dos órgãos do Poder Judiciário, à exceção do STF, a Política de Cultura e Educação em Segurança Cibernética no âmbito do Poder Judiciário (PCESC-PJ).

Parágrafo único. A PCESC-PJ será disciplinada por ato do Presidente do CNJ.

### CAPÍTULO X

#### DO ORÇAMENTO

Art. 31. Para execução das ações estratégicas, os órgãos do Poder Judiciário, objeto desta norma, deverão destinar os recursos orçamentários necessários.

Parágrafo único. Os recursos orçamentários deverão ser discriminados em rubrica específica para possibilitar que a Governança Nacional em Segurança Cibernética possa avaliar, de forma clara, os investimentos no setor.

### CAPÍTULO XI

#### DAS DISPOSIÇÕES FINAIS

Art. 32. Poderão ser instituídos planos de ações para detalhar a forma de aplicação da presente estratégia de segurança cibernética de acordo com a prioridade definida pelo CGSI-PJ.

Art. 33. Outros instrumentos complementares poderão ser elaborados e formalizados em normativos específicos do órgão desde que não contrariem as disposições estabelecidas nesta Resolução.

Art. 34. Ficam revogadas as Resoluções CNJ nº 360/2020; nº 361/2020 e nº 362/2020.

Art. 35. Ficam revogados os arts. 39 e 40 da Resolução CNJ nº 370/2021.

Art. 36. Esta Resolução entra em vigor na data de sua publicação.



Poder Judiciário

*Conselho Nacional de Justiça*

Ministro **LUIZ FUX**



## REFERÊNCIAS

Os links citados abaixo servem apenas como referência. Nos termos da lei brasileira (lei nº 9.610/98, art. 8º), não possuem proteção de direitos de autor: As ideias, procedimentos normativos, sistemas, métodos, projetos ou conceitos matemáticos como tais; Os esquemas, planos ou regras para realizar atos mentais, jogos ou negócios; Os formulários em branco para serem preenchidos por qualquer tipo de informação, científica ou não, e suas instruções; Os textos de tratados ou convenções, leis, decretos, regulamentos, decisões judiciais e demais atos oficiais; As informações de uso comum tais como calendários, agendas, cadastros ou legendas; Os nomes e títulos isolados; O aproveitamento industrial ou comercial das ideias contidas nas obras.

Caso não concorde com algum item do material entre em contato com a Domina Concursos para que seja feita uma análise e retificação se necessário. A Domina Concursos não possui vínculo com nenhuma banca de concursos, muito menos garante a vaga ou inscrição do candidato em concurso. O material é apenas um preparatório, é de responsabilidade do candidato estar atento aos prazos dos concursos.

A Domina Concursos reserva-se o direito de efetuar apenas uma devolução parcial do conteúdo, tendo em vista que as apostilas são digitais, isso, [e, não há como efetuar devolução do material.

**A Domina Concursos se preocupa com a qualidade do material, por isso todo conteúdo é revisado por profissionais especializados antes de ser publicado.**



[WWW.DOMINAONCURSOS.COM.BR](http://WWW.DOMINAONCURSOS.COM.BR)  
[contato@dominaconcursos.com.br](mailto:contato@dominaconcursos.com.br)

Rua São José, nº 645, sala 01, Pavimento  
Térreo, Ed. Jardim Germânica - Criciúma/ SC -  
Brasil/ CEP 88801-520

WhatsApp (48) 9.9695-9070