

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/341327343>

# CSharp Programming Language

Book · January 2016

---

CITATIONS

0

---

READS

209

1 author:



[Alvaro F. Pinheiro](#)

University of Pernambuco

29 PUBLICATIONS 5 CITATIONS

SEE PROFILE

# Fundamentos da Engenharia de Software

## Linguagem de Programação C#

**PINHEIRO, Álvaro Farias**  
Autor

# Fundamentos da Engenharia de Software

## Linguagem de Programação C#

Publicação 2016

O autor acredita que todas as informações aqui apresentadas estão corretas e podem ser utilizadas para qualquer fim legal. Entretanto, não existe qualquer garantia explícita ou implícita, de que o uso de tais informações conduzirá sempre ao resultado desejado. Os nomes de sites e empresas, por ventura, mencionados, foram utilizados apenas para ilustrar os exemplos, não tendo vínculo nenhum com o livro, não garantindo a sua existência nem divulgação. Eventuais erratas estarão disponíveis para download no site de publicação.

Algumas imagens utilizadas neste livro foram obtidas a partir da Internet.

Dados da Publicação

**Pinheiro, Álvaro Farias**

Fundamentos da Engenharia de Software: Linguagem de Programação C#

1. ed. – Volume VIII

Recife, 2016

Selo Editorial: Publicação Independente

### **Bibliografia**

**ASIN B01HAKC03M**

**ISBN: 9781519094278**

1. Linguagem de Programação C#
2. Linguagem de Marcação ASPX e CSS
3. Linguagem de Script JavaScript
4. Linguagem de Declaração SQL

# Fundamentos da Engenharia de Software

## Linguagem de Programação C#

Publicação Independente  
Livro em português com o título  
Fundamentos da Engenharia de Software  
Linguagem de Programação C#  
1ª Edição  
Recife - Pernambuco – Brasil  
Primeira revisão em Junho de 2016

# Fundamentos da Engenharia de Software

## Linguagem de Programação C#

À minha esposa Fabiana Pinheiro e aos meus filhos Alberto Pinheiro, Argel Pinheiro e Gabriel Pinheiro.

# Fundamentos da Engenharia de Software

## Linguagem de Programação C#

“O DNA humano é como um programa de computador, mas muito mais avançado do que qualquer software jamais inventado.”  
Bill Gates

# SUMÁRIO

|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| 1 Introdução às Tecnologias Web .....                                              | 1  |
| 1.1 Internet .....                                                                 | 1  |
| 1.2 Protocolos de Comunicação .....                                                | 1  |
| 1.3 Servidor Web.....                                                              | 4  |
| 1.4 Aplicação Web .....                                                            | 4  |
| 1.5 Web 2.0 .....                                                                  | 5  |
| 1.6 Conceitos da Web 2.0.....                                                      | 6  |
| 1.7 Conceitos, Siglas, Serviços e Software da Web.....                             | 6  |
| 1.8 Futuro da Web.....                                                             | 9  |
| 2 Introdução à Dot Net.....                                                        | 11 |
| 2.1 Arquitetura .....                                                              | 11 |
| 2.2 Recursos .....                                                                 | 11 |
| 2.3 Introdução ao ASPX.....                                                        | 13 |
| 2.4 Recursos do ASPX.....                                                          | 13 |
| 3 Dicas do ASPX e CSS.....                                                         | 15 |
| 3.1 Dica: Adicionar arquivo CSS a uma página ASPX. ....                            | 15 |
| 3.2 Dica: Adicionar um ícone a uma página ASPX. ....                               | 15 |
| 3.3 Dica: Adicionar título a uma página ASPX.....                                  | 15 |
| 3.4 Dica: Escrever uma função JavaScript para abrir uma URL numa página ASPX. .... | 16 |
| 3.5 Dica: Chamar uma função JavaScript em uma evento ASPX.....                     | 16 |
| 3.6 Dica: Carregar estilos de páginas para arredondar bordas de objetos ASPX. .... | 16 |
| 3.7 Dica: Carregar estilos de páginas para alinhar objetos. ....                   | 17 |
| 3.8 Dica: Carregar estilos de páginas para modificar objetos ASPX. ....            | 17 |
| 3.9 Dica: Criar menu suspenso com CSS. ....                                        | 18 |
| 3.10 Dica: Criar estilo CSS com bordas arredondadas. ....                          | 19 |
| 3.11 Dica: Criar estilo CSS com bordas com imagem.....                             | 19 |
| 3.12 Dica: Criar estilo CSS com background.....                                    | 20 |
| 3.13 Dica: Criar estilo CSS com definição de cores. ....                           | 20 |
| 3.14 Dica: Criar estilo CSS com difinição de cores gradiente. ....                 | 20 |
| 3.15 Dica: Criar estilo CSS com sombreamento. ....                                 | 20 |
| 3.16 Dica: Criar estilo CSS com ou sem quebra de palavra. ....                     | 21 |
| 3.17 Dica: Criar estilo CSS com definição de fonte. ....                           | 21 |
| 3.18 Dica: Criar estilo CSS com rotação de um elemento. ....                       | 21 |
| 3.19 Dica: Criar estilo CSS com efeito de transição.....                           | 22 |
| 3.20 Dica: Criar estilo CSS com efeito de animação.....                            | 22 |
| 3.21 Dica: Criar estilo CSS com efeito de transparência.....                       | 23 |
| 3.22 Dica: Criar estilo CSS com definição de duração de um efeito.....             | 23 |

# SUMÁRIO

|                                                                                                   |    |
|---------------------------------------------------------------------------------------------------|----|
| 3.23 Dica: Criar estilo CSS com definição de múltiplas colunas. ....                              | 23 |
| 3.24 Dica: Criar estilo CSS com efeito de alinhamento container. ....                             | 23 |
| 3.25 Dica: Criar estilo CSS com efeito de alinhamento de itens de um container. ....              | 24 |
| 3.26 Dica: Criar estilo CSS com efeito de alinhamento de itens de um container. ....              | 24 |
| 4 Dicas do JavaScript .....                                                                       | 25 |
| 4.1 Dica: Retornar as dimensões (altura e largura) de uma página cliente. ....                    | 25 |
| 4.2 Dica: Adicionar arquivo de som a uma página ASPX.....                                         | 25 |
| 4.3 Dica: Adicionar uma função de alerta popup página ASPX. ....                                  | 26 |
| 4.4 Dica: Adicionar uma função de impressão numa página ASPX.....                                 | 26 |
| 4.5 Dica: Como resolver mensagem de erro no uso do Timer com o UpdatePanel. ....                  | 26 |
| 5 Dicas do CSharp.....                                                                            | 27 |
| 5.1 Dica: Rotina para Capitalizar a primeira letra de cada nome de uma cadeia de caracteres. .... | 27 |
| 5.2 Dica: Rotina para enviar e-mail pelo SMTP do GMail. ....                                      | 28 |
| 5.3 Dica: Importar um arquivo armazenando seu binário no banco de dados. ....                     | 28 |
| 5.4 Dica: Definir tema (skin) de página na pré-inicialização.....                                 | 30 |
| 5.5 Dica: Imprimir dados do buffer de impressão .....                                             | 30 |
| 5.6 Dica: Definir estilo dos objetos na carga da página.....                                      | 31 |
| 5.7 Dica: Obter dados do navegador (Browser) na carga da página. ....                             | 32 |
| 5.8 Dica: Obter usuário logado na rede autenticada windows. ....                                  | 33 |
| 5.9 Dica: Configurar uma janela popup. ....                                                       | 33 |
| 5.10 Dica: Enviar e-mails num determinado horário.....                                            | 37 |
| 5.11 Dica: Gerar um registro de acesso ao sistema armazenando numa tabela de LOGs. ....           | 38 |
| 5.12 Dica: Exibir data com mês por extenso. ....                                                  | 39 |
| 5.13 Dica: Obter configurações do banco de dados do arquivo web.config. ....                      | 39 |
| 5.14 Dica: Executar um JavaScript para setar o focus num objeto ASPX no código CSharp. ....       | 39 |
| 5.15 Dica: Baixar (download) de arquivo no cliente armazenado no servidor. ....                   | 39 |
| 5.16 Dica: Carregar configurações de estilo de uma arquivo CSS via CSharp. ....                   | 40 |
| 5.17 Dica: Realizar compartilhamento de informações nas redes sociais. ....                       | 40 |
| 5.18 Dica: Abrir uma janela popup em JavaScript via código CSharp. ....                           | 43 |
| 5.19 Dica: Varrer objeto de uma página ASPX no código CSharp. ....                                | 43 |
| 5.20 Dica: Converter arquivo binário para string. ....                                            | 44 |
| 5.21 Dica: Enviar email pelo GMail. ....                                                          | 44 |
| 5.22 Dica: Obter o string do SELECT de um GridView. ....                                          | 45 |
| 5.23 Dica: Obter o string do SELECT de um DetailView. ....                                        | 46 |
| 5.24 Dica: Retornar o dia da semana de uma data. ....                                             | 47 |
| 5.25 Dica: Retornar o mês por extenso. ....                                                       | 49 |
| 5.26 Dica: Exibir uma saudação baseado no horário. ....                                           | 50 |

# SUMÁRIO

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| 5.27 Dica: Transformar a primeiro letra em maiúscula e as demais em minúsculas. ....  | 50 |
| 5.28 Dica: Exportar dados de uma gridview para formato EXCEL ou HTML ou JPG. ....     | 50 |
| 5.29 Dica: Preparar os diversos tipos de controles para serem exportados. ....        | 51 |
| 5.30 Dica: Enviar email formatado em HTML. ....                                       | 52 |
| 5.31 Dica: Ativar recurso de som. ....                                                | 58 |
| 5.32 Dica: Alerta sonoro em javascript. ....                                          | 58 |
| 5.33 Dica: Exibir janela de alerta (popup) em javascript. ....                        | 58 |
| 5.34 Dica: Converter String para ArrayList usando caracter especial para quebra. .... | 58 |
| 5.35 Dica: Mudar a cor de célula de uma gridview. ....                                | 59 |
| 5.36 Dica: Converter HTML para String. ....                                           | 60 |
| 5.37 Dica: Retornar um DataTable de uma SELECT. ....                                  | 60 |
| 5.38 Dica: Executar um comando SQL. ....                                              | 60 |
| 5.39 Dica: Adicionar meta tags a uma página ASPX. ....                                | 61 |
| 5.40 Dica: Inserir código antes e depois do código herdado. ....                      | 62 |
| 5.41 Dica: Obter dados do navegador. ....                                             | 62 |
| 5.42 Dica: Limpar mensagens de erro da aplicação. ....                                | 63 |
| 5.43 Dica: Abrir uma página ASPX via javascript. ....                                 | 63 |
| 5.44 Dica: Converter binário para uma imagem. ....                                    | 64 |
| 5.45 Dica: Modificar as características de uma gridview. ....                         | 64 |
| 5.46 Dica: Aplicar uma expressão regular em uma string. ....                          | 65 |
| 6 Dicas do SQL .....                                                                  | 66 |
| 6.1 Structure Query Language (SQL) .....                                              | 66 |
| 6.1.1 Inserção de Metadados (Criação de Tabelas) .....                                | 66 |
| 6.1.2 Inserção de Dados (Linhas   Registros   Ocorrências   Tuplas) .....             | 67 |
| 6.1.3 Alteração de Metadados (Modificar Estruturas) .....                             | 68 |
| 6.1.4 Alteração de Dados (Atualização) .....                                          | 69 |
| 6.1.5 Algumas Funções .....                                                           | 70 |
| 6.1.6 Índices .....                                                                   | 70 |
| 6.1.7 Dicionário de Dados .....                                                       | 71 |
| 6.1.8 Consulta de Dados .....                                                         | 71 |
| 6.1.9 Precedência .....                                                               | 72 |
| 6.1.10 Comparação .....                                                               | 72 |
| 6.1.11 Projeção .....                                                                 | 72 |
| 6.1.12 Seleção .....                                                                  | 72 |
| 6.1.13 Junção .....                                                                   | 72 |
| 6.1.14 Consulta Básica .....                                                          | 72 |
| 6.1.15 Apelido (Alias) .....                                                          | 73 |

# SUMÁRIO

|                                            |    |
|--------------------------------------------|----|
| 6.1.16 Campos Calculados.....              | 73 |
| 6.1.17 Valores Nulos.....                  | 73 |
| 6.1.18 Duplicidade .....                   | 74 |
| 6.1.19 Condicionamento.....                | 74 |
| 6.1.20 Ordenação .....                     | 75 |
| 6.1.21 Aninhamento.....                    | 76 |
| 6.1.22 Funções .....                       | 76 |
| 6.1.23 Funções Caracteree .....            | 76 |
| 6.1.24 Funções Numérica.....               | 77 |
| 6.1.25 MOD   % .....                       | 77 |
| 6.1.26 Funções Data.....                   | 77 |
| 6.1.27 Função Conversão .....              | 78 |
| 6.1.28 Função de Busca e Troca.....        | 78 |
| 6.1.29 Função de Condicionamento.....      | 78 |
| 6.1.30 Produto Cartesiano.....             | 79 |
| 6.1.31 Junção .....                        | 79 |
| 6.1.32 Equijunção .....                    | 79 |
| 6.1.33 Não-equijunção.....                 | 80 |
| 6.1.34 Junção Externa .....                | 80 |
| 6.1.35 Autojunção .....                    | 80 |
| 6.1.36 Funções de Grupo .....              | 81 |
| 6.1.37 Grupos de Dados.....                | 81 |
| 6.1.38 Grupo de Dados com Filtro .....     | 82 |
| 6.1.39 ROLLUP.....                         | 82 |
| 6.1.40 CUBE .....                          | 83 |
| 6.1.41 Ordenação de Dados.....             | 83 |
| 6.1.42 Aninhamento de Grupo de Dados ..... | 83 |
| 6.1.43 Subconsulta .....                   | 84 |
| 6.1.44 ANY.....                            | 84 |
| 6.1.45 ALL.....                            | 85 |
| 6.1.46 Correlacionadas.....                | 85 |
| 6.1.47 EXISTS .....                        | 85 |
| 6.1.48 União.....                          | 85 |
| 6.1.49 Intersecção .....                   | 86 |
| 6.1.50 Subtração.....                      | 86 |
| 6.1.51 Macrossubstituição.....             | 86 |
| 6.1.52 Seleção de Seleção .....            | 86 |

# SUMÁRIO

- 6.1.53 Análise Top-N .....86
- 6.1.54 Excluir Registros .....87
- 6.1.55 Excluir e recriar tabela: .....87
- 6.1.56 Excluir Tabela .....87
- 6.1.57 Transações .....87
- 6.1.58 Visão .....87
- 6.1.59 Controle de Usuários .....88
- 6.1.60 Privilégios .....89
- 6.1.61 Sequências .....89
- 6.1.62 Vínculos .....90
- 6.2 Objetos de Bancos de Dados .....90
- 6.3 Principais Funções .....90
- 6.4 Tipos de Dados .....95
- 6.5 Usuários e Senhas Padrões .....96
- 6.6 Exemplos .....96
- 6.7 Exercícios .....97
- 7 Referências Bibliográficas .....108
- 8 Referências Web .....109
- 9 Biografias .....110
- 10 Artefatos (Templates) para Projetos de *Software* .....113
- 11 Ferramentas .....114
- 12 Apresentações e Vídeos .....115
- 13 Sugestões de Leitura .....116

## **SIGLAS e ACRÔNIMOS**

AJAX - *Asynchronous Javascript and XML*  
ANSI - *American National Standards Institute*  
ASPX - *Active Server Pages eXtensible*  
HTML - *Hypertext Markup Language*  
IIS - *Internet Information Services*  
MVC - *Model–View–Controller*  
OOP - *Object Oriented Programming*  
SQL - *Structured Query Language*  
XHTML - *eXtensible Hypertext Markup Language*  
XML - *eXtensible Markup Language*

# NOTAÇÕES

Este livre adota as seguintes notações, que devem ser observadas para o perfeito entendimento do material aqui produzido.

## Notações Técnicas:

- <..> - Menor que / Maior que, nos comandos SQL usado para metadados.
- <..> - Menor / Maior, no texto indica que é um *hyperlink* (*link* para um endereço web).
- [..] - Abre e fecha parênteses, no texto indica utilização de uma sigla.
- (..) - Abre e fecha colchetes, nos comandos SQL indica opcionalidade.
- | - Barra, para indicar "ou lógico".
- Sublinhado, nos comandos SQL indica o *default* (padrão).
- Sublinhado, no texto indica que é um *hyperlink* (*link* para um endereço web).
- Negritos** - Títulos de capítulos, seções e subseções.
- Itálico* - Para palavras de língua estrangeira.

## SOBRE O AUTOR



**Álvaro Farias Pinheiro** tem mais de 30 anos de experiência no campo da computação e nestes, mais de 15 anos na área acadêmica, lecionando disciplinas com foco em engenharia de software.

É mestre em Engenharia de Software pelo Centro de Estudos e Sistemas Avançados do Recife (CESAR) com a dissertação *Uso do Balanced Scorecard na Avaliação de Fábricas de Software* utilizadas como Instrumento de Ensino de Tecnologia da Informação Baseado em Problemas Reais. Especialista em Gestão Pública pela Universidade de Pernambuco (UPE) e em Metodologias de Desenvolvimento pela União Brasileira de Tecnologia. É Bacharel em Sistemas de Informação pela Faculdade Integrada do Recife (FIR), iniciando sua vida acadêmica no Bacharelado em Ciência da Computação da Universidade Católica de Pernambuco (UNICAP).

Iniciou as atividades profissionais em 1986 como Programador na Elógica Microsistemas desenvolvendo para baixas e altas plataformas nas linguagens dBase, Clipper, FoxPro, Cobol e Natural. Como Analista de Sistemas passou pelas instituições: Centro de Prestação de Serviços Técnicos de Pernambuco (CETEPE), Empresa de Fomento da Informática do Estado de Pernambuco (FISEPE), Instituto de Tecnologia em Informática (ITECI), Apply Soluções e está hoje na Agência Estadual de Tecnologia da Informação do Estado de Pernambuco (ATI-PE) por concurso público, trabalhando como Analista em Gestão da Tecnologia da Informação e Comunicação, em exercício na Procuradoria Geral do Estado de Pernambuco (PGE-PE), atuando como Coordenador da Área de Sistemas.

Começou a lecionar em 1992 em cursos técnicos nas empresas ITECI e IBRATEC, e desde 2006 é professor de cursos de graduação e pós-graduação em computação nas instituições de ensino superior da região, a citar, Universidade Federal Rural de Pernambuco (UFRPE), Instituto Federal de Pernambuco (IFPE), Faculdade Estácio do Recife, Faculdade Integrada do Recife (FIR), Faculdade Joaquim Nabuco, Faculdade do Recife, Unibratrec e durante esses anos realizou diversos cursos de extensão como os de Análise de Ponto de Função, Administração de Redes, Linguagens de Programação Natural, Centura, Delphi e CSharp, Banco de Dados Adabas, MySQL, Postgre e Oracle, Análise Estruturada e Orientada a Objetos, entre outros.

Possui experiência na área de Engenharia de *Software* nas disciplinas de Teoria da Computação, Sistemas de Informação e Colaborativos, Desenvolvimento Estrutural e Orientado a Objetos para aplicações Desktop, Web e Mobile, Processos de Desenvolvimento IRUP, XP e Scrum, Banco de Dados SQLServer e Oracle, e Gestão de Projetos com PMBOK e SWEBOOK. Trabalha atualmente na coordenação de projetos e com educação presencial e a distância, ministrando aulas e realizando pesquisas em ambientes virtuais de ensino e linguagens de programação.

<http://www.alvarofpinheiro.eti.br>  
<http://br.linkedin.com/in/alvarofpinheiro>  
<https://sites.google.com/site/alvarofpinheiro>  
<http://www.facebook.com/alvarofpinheiroaulas>

# PREFÁCIO

Foi com surpresa e felicidade que recebemos o convite de Álvaro Farias Pinheiro para elaborar o prefácio do oitavo livro da série Fundamentos de Engenharia de Software.

A surpresa decorre da solicitação para prefaciar essa obra a dois Procuradores, com formação em Direito, e sem qualquer conhecimento técnico em tecnologia da informação. Porém, a surpresa se tornou um desafio a ser escrito não como especialistas no assunto, mas como usuários e entusiastas. Aí está a felicidade e a honra em poder contribuir com o autor, oferecendo essas palavras iniciais, e como o leitor, ao transmitir o quanto esses fundamentos em engenharia de software podem transformar e facilitar a vida de usuários como nós.

E a felicidade maior decorre de o convite ter vindo de Álvaro. Nós o conhecemos quando assumimos as funções, respectivamente, de Secretária Geral da Procuradoria do Estado de Pernambuco (PGE/PE) e Coordenador do Centro de Estudos Jurídicos. Imediatamente, ficamos impressionados com a inteligência, dedicação e a simplicidade dele. O talento que ele tem é enorme de transformar ideias em aplicativos e a capacidade de, quase instantaneamente, apresentar soluções tecnológicas para atendimento às diversas necessidades da Procuradoria.

Embora atuando em ambiente de poucos recursos, Álvaro optou por driblar as dificuldades com criatividade, rejeitando a via fácil das lamentações tantas vezes trilhada no Serviço Público. Sua capacidade de criar muito com pouco está materializada nas múltiplas ferramentas por ele inseridas na extranet e na intranet da PGE/PE.

Esse mais novo livro, dedicado especialmente à Linguagem de Programação CSharp, tem em seu DNA muitas das qualidades do autor: é simples, inteligente e didático, apresentando utilíssimas dicas práticas para solução de problemas cotidianos dos programadores no desenvolvimento de aplicações web.

Ao autor, desejamos vida longa ao livro e muitas edições e tiragens. Ao leitor, esperamos que possa aprender, divulgar e contribuir positivamente com usuários como nós ávidos por tecnologia.

Recife, 27 de junho de 2016.

**Giovana Andréa Gomes Ferreira**

Procuradora do Estado de Pernambuco

Procuradora-Chefe da Procuradoria Consultiva

Especialista em Direito Processual Civil pela Universidade de Santa Cruz do Sul

Bacharel em Direito pela Universidade Federal de Pernambuco (UFPE)

Bacharel em Letras pela Universidade Federal de Pernambuco (UFPE)

**Paulo Rosenblatt**

Procurador do Estado de Pernambuco

Coordenador do Centro de Estudos Jurídicos

Doutor em Direito Tributário pelo Institute of Advanced Legal Studies - Universidade de Londres

Mestre em Direito Público pela Universidade Federal de Pernambuco (UFPE)

Bacharel em Direito pela Faculdade de Direito do Recife (FDR/UFPE)

O livro é uma coletânea de códigos em C#, ASPX e JavaScript linguagens que fui apresentado em 2005. Durante os anos seguintes tive a oportunidade de desenvolver várias aplicações para a iniciativa privada e pública com estas linguagens, sendo o trabalho mais recente os sites da Procuradoria Geral do Estado de Pernambuco tanto a Extranet (<http://www.pge.pe.gov.br>) como a Intranet (Portal-PGE).

O motivo que me levou a escrever oitavo livro da série Fundamentos da Engenharia de Software foi o interesse de compartilhar com a comunidade de desenvolvedores as dicas de código das poderosas linguagens de programação, marcação e script respectivamente CSharp, ASPX/CSS e JavaScript.

# PREFÁCIO

Códigos que me são úteis na codificação de soluções para problemas corriqueiros no desenvolvimento de aplicações web estão aqui disponibilizados. Assim sendo esse livro é uma coleção de dicas de código comentadas que podem ser utilizados na hora que determinada dúvida ocorrer.

Esse livro não se destina a ensinar a programar nessas linguagens e sim ser um guia rápido para desenvolvedores já experientes que buscam uma ajuda de como fazer determinada função ou mostrar uma alternativa, uma ideia como ponto de partida.

No momento estão disponibilizadas dicas de código em JavaScript linguagem de script mais utilizada pelos navegadores, dicas na linguagem de marcação ASPX e CSS, e principalmente dicas da linguagem de programação CSharp e linguagem SQL para banco de dados SQLServer.

O livro é distribuído nos seguintes capítulos:

- **Capítulo 1** *Introdução às Tecnologias Web*. Neste são abordados os conceitos básicos das tecnologias Web.
- **Capítulo 2** *Introdução à Dot Net*. Aqui se introduz nos conceitos básicos da plataforma Dot Net.
- **Capítulo 3** *Dicas do ASPX*. Dicas para a linguagem de marcação Active Serve Page.
- **Capítulo 4** *Dicas do JavaScript*. Dicas da linguagem de script mais utilizada pelos navegadores atualmente.
- **Capítulo 5** *Dicas do CSharp*. Aqui o foco está nas dicas para a linguagem de programação C#.
- **Capítulo 6** *Dicas do SQL*. Neste são abordados os conceitos básicos dos bancos de dados SQL Server.
- **Capítulo 7** *Referências Bibliográficas*. Aqui se referencia os livros utilizados para compor esse material.
- **Capítulo 8** *Referências Web*. Neste são referenciados os sites utilizados para compor o material.
- **Capítulo 9** *Biografias*. São listados os principais autores e sua biografia da área de TIC.
- **Capítulo 10** *Artefatos*. Neste são disponibilizados alguns documentos necessários para o desenvolvimento.
- **Capítulo 11** *Ferramentas*. Neste são abordados algumas das ferramentas de desenvolvimento.
- **Capítulo 12** *Apresentações de Vídeo*. Neste são disponibilizados algumas vídeo aulas sobre o tema.
- **Capítulo 13** *Sugestões de Leitura*. Aqui são sugeridas leituras complementares ao tema.

Álvaro Farias Pinheiro

### 1 Introdução às Tecnologias Web

Tecnologias Web consistem em conhecer as origens da Internet no mundo e no Brasil; Entender os fundamentos da arquitetura da Internet, pois nos dias de hoje é difícil identificar uma área que ainda não tenha investido em tecnologia da informação e se beneficiado com as facilidades trazidas pela informação tratada em tempo real e disponível sob demanda.

A combinação de texto, som, imagem, entregues em uma pluralidade de formatos para diferentes dispositivos, permite, por exemplo, que um candidato a fornecedor receba informações atualizadas ou diferenciadas no caminho para o escritório do cliente. Também permite respostas rápidas, *on-time*, favorecendo a tomada de decisão mais assertiva. Esta configuração e apropriação da tecnologia que vem se estabelecendo, implica certamente em mudança na forma de: resolver problemas, comprar, vender, estudar, se comunicar, tomar decisões e também na forma de raciocinar e agir. São novos hábitos e costumes gerando novas demandas e oportunidades.

No espaço das demandas e das oportunidades, as soluções criativas e a inovação, principalmente aquelas que se utilizam da Internet, como os portais de conhecimento, as salas de aula virtuais, os jogos para Internet, os sistemas de e-commerce têm grande apreço no mercado de trabalho do profissional de informática. Sendo assim, o capítulo Tecnologias Web trás para o leitor um primeiro contato com este campo de possibilidades. Este capítulo permite que se conheça a Internet e suas características, recursos e aplicações. É um conhecimento importante, pois contribui no processo de aprendizagem da construção dos algoritmos e na programação de sistemas Web.

#### 1.1 Internet

Vamos iniciar explicando o conceito Web ou WWW ou *World Wide Web* ou Internet ou Rede de Alcance Mundial. Trata-se de um sistema que permite o *upload* e o *download* de arquivos que possuem embutidos o conceito de hyperlink ou hipermídia, possibilitando o link ou interligação dos mesmos na Internet. Esses arquivos podem ser vídeos, sons, hipertextos e figuras.

Para ser ter acesso a esses arquivos ou documentos ou páginas, tanto para visualizá-los, como para baixá-los (*download*) ou para atualizá-los (*upload*) é necessário se ter browser ou navegador.

Essas páginas ou *Webpages* ficam armazenadas em máquinas chamadas de servidores web ou sites ou sítios, e trafegam pela rede em formato texto, mas especificadamente em um protocolo de comunicação chamado de HTTP ou *Hyper Text Transfer Protocol* ou Protocolo de Transferência de Hiper Texto. Podendo o usuário do cyberspaço pular ou navegar ou surfar de página em página e mesmo interagir com outros usuários, software ou mesmo os web servers.

Essa lógica de funcionamento foi inicialmente definida na década de 60, mas de forma bem básica que definia o conceito de hipertexto, chamado de projeto Xanadu. Porém foi nos anos 80 que ela seria evoluída por Tim Berners Lee da Organização Européia para a Investigação Nuclear (CERN) com o objetivo de compartilhar documentos de pesquisas mais facilmente. Mas, foi no final da década de 80 que a ideia tomou a forma que veria a ser protótipo do primeiro servidor Web concebido o NeXTcube, sendo desenvolvido o primeiro navegador batizado de WorldWideWeb já em 1990. Mas foi em 6 de agosto de 1991 que oficialmente a Internet tomou vida, sendo publicado para o mundo acadêmico-científico esse tipo de aplicação.

#### 1.2 Protocolos de Comunicação

Agora que já sabemos o que é uma tecnologia Web e como surgiu vamos entender o que é protocolo de comunicação e quais os protocolos que a Internet faz uso.

Protocolo é uma convenção ou padrão que controla e possibilita uma conexão, comunicação ou transferência de dados entre dois sistemas computacionais. É a gramática que define as formas léxica, sintática e semântica da comunicação. Essa gramática pode ser implementada via hardware, software ou ambos.

*World Wide Web* (que em português significa "Rede de alcance mundial"; também conhecida como Web e www) é um sistema de documentos em hipermídia que são interligados e executados na Internet. Os documentos podem estar na forma de vídeos, sons, hipertextos e figuras. Para visualizar a informação, pode-se usar um programa de computador chamado navegador para descarregar informações (chamadas "documentos" ou "páginas") de servidores web (ou "sítios") e mostrá-los na tela do usuário. O usuário pode então seguir as hiperligações na página para outros documentos ou mesmo enviar informações de volta para o servidor para interagir com ele. O ato de seguir hiperligações é, comumente, chamado de "navegar" ou "surfar" na Web.

*Hypertext Transfer Protocol* (HTTP), que significa Protocolo de Transferência de Hipertexto, é um protocolo de comunicação (na camada de aplicação segundo o Modelo OSI) utilizado para transferir dados por intranets e pela *World Wide Web*. Normalmente, este protocolo utiliza a porta 80 e é usado para a comunicação de sítios web, comunicando na linguagem HTML.

*HyperText Transfer Protocol Secure* (HTTPS), é uma implementação do protocolo HTTP sobre uma camada SSL ou do TLS. Essa camada adicional permite que os dados sejam transmitidos através de uma conexão criptografada e que se verifique a autenticidade do servidor e do cliente através de certificados digitais. A porta TCP usada por norma para o protocolo HTTPS é a 443. O protocolo HTTPS é utilizado, em regra, quando se deseja evitar que a informação transmitida entre o cliente e o servidor seja visualizada por terceiros, como, por exemplo, no caso de compras online. A existência na barra de tarefas (normalmente do lado direito) de um cadeado demonstra a certificação de página segura (SSL). Nas URLs dos sites o início ficaria 'https://'. Geralmente os navegadores mais atuais indicam um site seguro, geralmente através das barras de endereço que ficam verdes.

*Domain Name System* (DNS) Sistema de Nomes de Domínios é um sistema de gerenciamento de nomes hierárquico e distribuído operando segundo duas definições: Examinar e atualizar seu banco de dados. Resolver nomes de servidores em endereços de rede (IPs).

*Simple Mail Transfer Protocol* (SMTP) é o protocolo padrão para envio de e-mails através da Internet e é baseado em texto simples, onde um ou vários destinatários de uma mensagem são especificados (e, na maioria dos casos, validados) sendo, depois, a mensagem transferida.

*Post Office Protocol* (POP3) é um protocolo utilizado no acesso remoto a uma caixa de correio eletrônico e permite que todas as mensagens contidas numa caixa de correio eletrônico possam ser transferidas sequencialmente para um computador local.

*Internet Message Access Protocol* (IMAP) é um protocolo de gerenciamento de correio eletrônico superior em recursos ao POP3 - protocolo que a maioria dos provedores oferece aos seus assinantes. O mais interessante é que as mensagens ficam armazenadas no servidor e o internauta pode ter acesso a suas pastas e mensagens em qualquer computador, tanto por webmail como por cliente de correio eletrônico.

*File Transfer Protocol* (FTP) Protocolo de Transferência de Arquivos, é uma forma bastante rápida e versátil de transferir arquivos, sendo uma das mais usadas na internet.

Telnet é um protocolo cliente-servidor de comunicações usado para permitir a comunicação entre computadores ligados numa rede.

TCP/IP é um conjunto de protocolos de comunicação entre computadores em rede. Seu nome vem dos dois protocolos mais importantes do conjunto: o *Transmission Control Protocol* (TCP) Protocolo de Controle de Transmissão e o *Internet Protocol* (IP) Protocolo de Interconexão. O conjunto de protocolos pode ser visto como um modelo de camadas, onde cada camada é responsável por um grupo de tarefas, fornecendo um conjunto de serviços bem definidos para o protocolo da camada superior. As camadas mais altas estão logicamente mais perto do usuário (chamada camada de aplicação) e lidam com dados mais abstratos, confiando em protocolos de camadas mais baixas para tarefas de menor nível de abstração.

*Session Initiation Protocol* (SIP) é um protocolo de aplicação, que utiliza o modelo "requisição-resposta", similar ao HTTP, para iniciar sessões de comunicação interativa entre utilizadores. SIP é um protocolo de sinal para estabelecer chamadas e conferências através de redes via Protocolo IP.

*Simple Network Management Protocol* (SNMP) Protocolo Simples de Gerência de Rede é um protocolo de gerência típica de redes TCP/IP, da camada de aplicação, que facilita o intercâmbio de informação entre os dispositivos de rede, como placas e comutadores (em inglês: switches).

*Transmission Control Protocol* (TCP) é um dos protocolos sob os quais assenta o núcleo da Internet nos dias de hoje. A versatilidade e robustez deste protocolo tornaram-nas adequadas a redes globais, já que este verifica se os dados são enviados de forma correta, na sequência apropriada e sem erros, pela rede.

*User Datagram Protocol* (UDP) é um protocolo simples da camada de transporte. Ele permite que a aplicação escreva um datagrama encapsulado num pacote IPv4 ou IPv6, e então enviado ao destino. Mas não há qualquer tipo de garantia que o pacote irá chegar ou não. O protocolo UDP não é confiável.

*Internet Protocol* (IP) Protocolo de Internet é um protocolo usado entre duas ou mais máquinas em rede para encaminhamento dos dados. Os dados numa rede IP são enviados em blocos referidos como pacotes ou datagramas que na versão 4 possui o tamanho de 32 bits.

IPv6 é a versão mais atual, do protocolo IP. Ele está sendo implantado gradativamente na Internet e deve funcionar lado a lado com o IPv4, numa situação tecnicamente chamada de "pilha dupla" ou "*dual stack*", por algum tempo. Em longo prazo, o IPv6 tem como objetivo substituir o IPv4, que só suporta cerca de 4 bilhões ( $4 \times 10^9$ ) de endereços, contra cerca de  $3.4 \times 10^{38}$  endereços do novo protocolo.

*Address Resolution Protocol* (ARP) é um protocolo usado para encontrar um endereço da camada de enlace de rede (Ethernet, por exemplo) a partir do endereço da camada de rede (como um endereço IP). O emissor difunde em broadcast um pacote ARP contendo o endereço IP de outro host e espera uma resposta com um endereço MAC respectivo.

*Reverse Address Resolution Protocol* (RARP) ou Protocolo de Resolução Reversa de Endereços associa um endereço MAC conhecido a um endereço IP. Permite que os dispositivos de rede encapsulem os dados antes de enviá-los à rede.

*Internet Control Message Protocol* (ICMP) é um protocolo integrante do Protocolo IP e é utilizado para fornecer relatórios de erros à fonte original. Qualquer computador que utilize IP precisa aceitar as mensagens ICMP e alterar o seu comportamento de acordo com o erro relatado. Situações de Erro: Um pacote IP não consegue chegar; O Gateway não consegue retransmitir; e O Roteador indica uma rota melhor.

*IP Security Protocol* mais conhecido pela sua sigla, IPSec é uma extensão do protocolo IP que visa a ser o método padrão para o fornecimento de privacidade do usuário (aumentando a confiabilidade, integridade e autenticidade das informações).

*Real Time Protocol* (RTP) é um protocolo de redes utilizado em aplicações de tempo real como, por exemplo, entrega de dados áudio ponto-a-ponto, como Voz sobre IP.

*Dynamic Host Configuration Protocol* (DHCP) é um protocolo de serviço TCP/IP que oferece configuração dinâmica de terminais, com concessão de endereços IP de host e outros parâmetros de configuração para clientes de rede. Resumidamente, o DHCP opera da seguinte forma: Um cliente envia um pacote UDP em broadcast (destinado a todas as máquinas) com um pedido DHCP. Os servidores DHCP que capturarem este pacote irão responder (se o cliente se enquadrar numa série de critérios) com um pacote com configurações onde constará, pelo menos, um endereço IP, uma máscara de rede e outros dados opcionais, como o gateway, servidores de DNS, etc. O DHCP usa

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

um modelo cliente-servidor, no qual o servidor DHCP mantém o gerenciamento centralizado dos endereços IP usados na rede.

*Point-to-Point Protocol* (PPP) protocolo ponto-a-ponto, foi desenvolvido com o objetivo de transportar todo o tráfego entre 2 dispositivos de rede através de uma conexão física única. Embora seja um protocolo, o PPP encontra-se na lista de interfaces. Na prática, a interface PPP é implementada através de conexões físicas do tipo RS-232 ou modems. Atualmente é possível usar conexões PPP até sobre Ethernet (PPPoE).

### 1.3 Servidor Web

Um servidor Web se trata de um conjunto de hardware e software que tem várias funções e uma delas é responder as solicitações dos clientes, também chamado de submissões. O ordenador ou cliente envia uma submissão (*post*) que segue para a nuvem (Internet) até chegar ao servidor Web da aplicação em uso. Esse por sua vez trata a requisição e responde ao cliente. Lembrando que esse tráfego é realizado sobre o protocolo TCP/IP e a linguagem de comunicação é o HTTP ou HTTPS.

Então podemos dizer que um servidor web é um software, como exemplos têm o *Internet Information Server* (IIS) e o Tomcat entre muitos outros, que rodam sobre um hardware robusto o suficiente para atender diversos pedidos vindos de clientes via Internet. Esses clientes são navegadores (browsers) que interpretam página escrita em HTML linguagem aceita pelo protocolo HTTP.

Todo o processo tem início com o estabelecimento de comunicação chamado de conexão entre o ponto (host) A que é o servidor Web e o ponto (host) B que é o cliente. No cyberspace, ou melhor, o espaço cibernético, as conexões podem ocorrer a qualquer momento, assim os servidores web precisam estar disponíveis sempre. Com uma conexão estabelecida (*established*) o pedido de um cliente pode ser processado e conforme as restrições de segurança uma resposta é devolvida.

O Tomcat é um servidor web Java, sendo mais específico, serve para armazenar servlets, isto são programas escritos em Java para servidores Web. Servidor foi desenvolvido pela *Apache Software Foundation* e é *opensource*, aceito oficialmente pela Sun (hoje pertencente a Oracle) como servidor Java. Sendo uma referência para as tecnologias Java Servlet e Java Server Pages (JSP). Sendo assim quase um servidor de aplicação Web. Como servidor Web possui ferramentas de configuração e gerenciamento.

O *Internet Information Services*, que antes era chamado de *Internet Information Server* foi desenvolvido pela Microsoft e é distribuído juntamente com o Sistema Operacional Windows. Uma de suas funções é gerar páginas HTML dinâmicas a partir de solicitações dos clientes feitas em HTML ou em *Active Server Pages* (ASP e/ou ASPX), mas pode atender outras tecnologias através de *plugins*. Outra função desse servidor é interpretar Web Services chamados de ASMX.

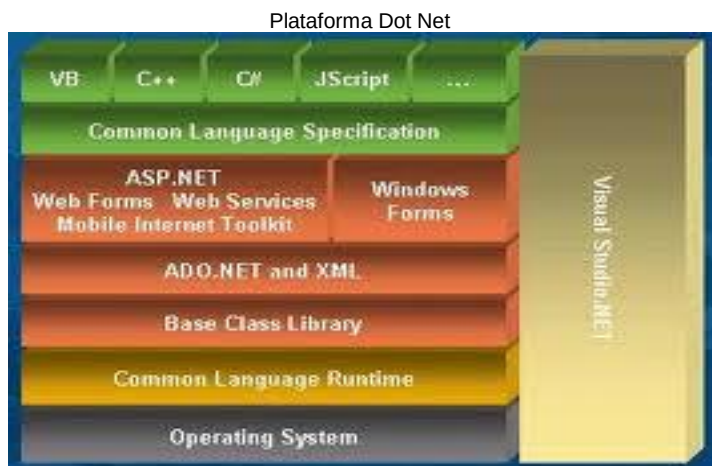
### 1.4 Aplicação Web

Existem várias possíveis possibilidades de construção de aplicações Web, mas irei citar duas, pois nos dias atuais são amplamente usadas. As aplicações desenvolvidas na plataforma Java e as desenvolvidas na plataforma Dot Net.

A plataforma Java, basicamente funciona da seguinte forma. Escreve-se um programa, o chamado código fonte em Java, uma linguagem orientada a objetos, desenvolvida pela Sun Microsystems que recentemente foi adquirida pela Oracle. Depois de escrito o programa deve-se compilar e um compilador disponível é o JAVAC. O resultado dessa compilação é um código objeto intermediário chamado de bytecode, que pode ser interpretado por um middleware chamado de Máquina Virtual Java, que na sigla em inglês é JVM. Esse software é capaz de interpretar o Byte Code para um Sistema Operacional específico, por exemplo, Windows, Linux, MacOS, entre tantos outros - com essa arquitetura a linguagem Java permite uma característica muito desejada que seja a Portabilidade. Assim escrever um software não ficou mais atrelado a um hardware.

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

A outra plataforma amplamente utilizada é a da Microsoft que também faz uso do recurso de Máquina Virtual, essa chamada de Dot Net Framework. A grande diferença é que essa máquina trabalha basicamente para o sistema operacional Microsoft Windows. A portabilidade não se dá em relação aos sistemas operacionais e sim em relação às linguagens de programação. Existe um gama de linguagens construídas para essa plataforma, podemos citar: VisualBasic.Net, Delphi.Net, Python.Net, Ruby.Net, PHP.Net, VisualFoxPro.Net, C#, F#, entre outras. Possibilitando que a curva de aprendizagem de construção de uma aplicação Web seja bem menor. Além de possuir uma integração de vários recursos na IDE utilizada para o seu desenvolvimento que é o Visual Studio.



Fonte: Internet.

### 1.5 Web 2.0

A Web 2.0 foi criada pela em meados de 2004 com o intuito de fomentar e implementar uma segunda geração de comunidades e serviços que seriam criados exclusivamente para a plataforma Web e não adaptados a ela como muitos sistemas estavam sendo.

Assim, surgiram conceitos novos como Wiki's, folksonomia, redes sociais, etc. Apesar de essa terminologia ter a conotação de nova versão, trata-se de se usar a infra existente para desenvolver exclusivamente para ela - a Internet - com foco nos usuários e desenvolvedores, buscando a construção de ambientes mais participativos - sistemas colaborativos.

Segundo Tim O'Reilly a definição para ela é: "Web 2.0 é a mudança para uma internet como plataforma, e um entendimento das regras para obter sucesso nesta nova plataforma. Entre outras, a regra mais importante é desenvolver aplicativos que aproveitem os efeitos de rede para se tornarem melhores quanto mais são usados pelas pessoas, aproveitando a inteligência coletiva". Para entender melhor a Web 2.0 segue essa apresentação da [www.lidec.futuro.usp.br](http://www.lidec.futuro.usp.br):

- <http://www.youtube.com/watch?v=iCkkjA8T-Fs&feature=related>
- <http://www.youtube.com/watch?v=4M2zBNeDkql&feature=related>

Então podemos dizer que a Web 2.0 é baseada em arquiteturas de participação, ou melhor, dizendo sistemas colaborativos ou inteligência coletiva colaborativa.

Basicamente podemos dizer que a Web 2.0 os usuários é que criam o conteúdo, ficando a forma a critério das organizações que desenvolvem os sites.

Exemplos de sistemas colaborativos:

- MySpace;
- Flickr;
- YouTube;
- Wikipédia;
- Digg;
- reddit;
- RSSfeeds;
- Facebook;
- Bebo;
- LinkedIn; e
- Netflix.

Terminologias colaborativas:

- Bookmarking;
- Tagging;
- RRS;
- Avatar; e
- Cauda-longa.}

### 1.6 Conceitos da Web 2.0

São estruturas sociais compostas por pessoas e/ou organizações, conectadas por um ou vários tipos de relações, que partilham valores e objetivos comuns. A característica principal da conexão se dá através da identidade entre essas pessoas.

Os tipos de identidade podem ser:

- Redes de relacionamentos (facebook, myspace, twitter, ...);
- Redes profissionais (linkedin, plaxo, ...);
- Redes comunitárias (redes sociais em bairros ou cidades); e
- Redes políticas (como as pessoas alcançam os seus objetivos).

Uma característica comum entre os diversos tipos de rede social é o compartilhamento de informações, conhecimentos, interesses e esforços em busca de objetivos comuns. A intensificação da formação das redes sociais, nesse sentido, reflete um processo de fortalecimento da sociedade civil, em um contexto de maior participação democrática e mobilização social. Veja o exemplo do que acontece no mundo árabe.

Podemos dizer que as áreas de atuação das redes sociais são:

- Antropologia;
- Biologia;
- Comunicação;
- Economia;
- Geografia;
- Ciências da Informação; e
- Psicologia Social.

Mais uma informação, a empresa Compete publica uma lista com as maiores redes sociais do mundo com seus acessos por mês, fonte [www.seopositionne.com](http://www.seopositionne.com). Para mais detalhes social-network. Outro site interessante para obter informações dos Top10 é o <http://top-10-list.org/>.

### 1.7 Conceitos, Siglas, Serviços e Software da Web

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

Alguns conceitos importantes de serem conhecidos no mundo da Internet são citados a seguir:

- AdWords, Links com anúncios relacionados às palavras-chave utilizadas no motor de busca.
- AdSense, propagandas que geram lucros pela quantidade de cliques.
- Blogs, sites que disponibilizam espaço para publicação de ideias que podem ser comentadas.
- Bookmarking Social, compartilhamento de bookmarks na Web.
- Busca Horizontal, serviços de busca generalistas.
- Busca Localizada, serviços de busca que utilizam CEP ou IP para retornar conteúdo mais significativo.
- Busca Vertical, serviços de busca especialistas.
- Findability (encontrabilidade), caracteriza a facilidade de encontrar conteúdo.
- Feeds, sites que enviam conteúdo atualizado para grupo interesse.
- Gadgets, pequenos aplicativos que podem ser incorporados em sites.
- Geotagging, serviços de localização de conteúdo.
- Inteligência Coletiva, conceito que trata da colaboração que resulta em idéias inteligentes.
- Mashups, combinação de vários sites com a finalidade de se criar um novo site.
- Mídia Social, disponibilização de texto, imagem e som.
- PageRank, considera o número de Links de uma página.
- Redes de Blogs, coleção de blogs de vários editores.
- Redes de Conteúdo, sites que filtram informações de várias maneiras diminuindo a poluição de informação.
- Redes Sociais, focada nos relacionamentos interpessoais.
- RIA, aplicações Web com GUI semelhantes aos Desktops.
- SaaS, aplicação executada em um servidor Web.
- Scalability (escalabilidade), caracteriza o crescimento dos recursos utilizados.
- Mecanismos de Busca, ferramentas disponibilizadas para encontrar conteúdo.
- Tagging, rotulagem de conteúdo.
- Web Services, serviços remotos disponibilizados via métodos remotos.
- Webtop, aplicativos Web com ambiente semelhante aos Desktops.
- Widgets, componentes GUI tipo janelas, botões, menus, ícones, barras de rolagem, etc.. para sites.
- Wiki's, disponibilização de definições.

Também seguem algumas siglas muito usadas atualmente:

- AJAX – Asynchronous JavaScript and XML.
- CPA – Custo por Ação.
- CPM – Custo por Impressões.
- CPC - Custo por Clique.
- REST – Representational State Transfer.
- RIA – Rich Internet Application.
- SaaS – Software as a Service ou Software como um Serviço.
- SEM – Search Engine Marketing ou Monetização do Mecanismo de Busca.
- SEO – Search Engine Optimization ou Otimização do Mecanismo de Busca.
- SERP – Search Engine Results Page.

E a seguir seguem os serviços mais utilizados na Internet:

- <http://www.bing.com.br> serviço de busca generalista da Microsoft.
- <http://br.ask.com> serviço de busca generalista da InterActiveCorp.
- <http://www.kayak.com> serviço de busca especialista em turismo.
- <http://www.expedia.com> serviço de busca especialista em turismo.
- <http://www.zillow.com> serviço de busca especialista em imóveis.
- <http://www.trulia.com> serviço de busca especialista em imóveis.
- <http://www.indeed.com.br> serviço de busca especialista em empregos.
- <http://www.monster.com.br> serviço de busca especialista em empregos.
- <http://www.shopzilla.com> serviço de busca especialista em compras.
- <http://www.mysimon.com> serviço de busca especialista em compras.
- <http://technorati.com> serviço de busca especializa de mídia social.

<http://www.alvarofpinheiro.eti.br>

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

- <http://www.about.com> serviço de conteúdo diversificado.
- <http://www.deitel.com> serviço de conteúdo tecnológico.
- <http://www.ehow.com> serviço de conteúdo de como fazer qualquer coisa.
- <http://www.hsw.uol.com.br> serviço de conteúdo de como tudo funciona.
- <http://www.livetips.biz> serviço de conteúdo sobre dicas de como viver.
- <https://www.wesabe.com/groups> serviço de inteligência coletiva sobre finanças.
- <http://digg.com> serviço de mídia social.
- <http://www.flickr.com> serviço de mídia social.
- <http://picasa.google.com> serviço de mídia social.
- <http://www.lastfm.com.br> serviço de rádio na Web.
- <http://www.pandora.com> serviço de rádio na Web.
- <http://pt.wikipedia.org> serviço de wiki generalista.
- <http://www.wikia.com/Wikia> serviço wiki sobre arte.
- <http://recife.pt.craigslist.org> serviço de classificados.
- <http://pt-br.wordpress.com> serviço de blog.
- <http://www.blogger.com> serviço de blog.
- <http://twitter.com> serviço de microblogging.
- <http://www.friendster.com> serviço de rede social.
- <http://br.myspace.com> serviço de rede social.
- <http://pt-br.facebook.com> serviço de rede social.
- <http://www.linkedin.com> serviço de rede social.
- <https://www.xing.com> serviço de rede social.
- <http://secondlife.com> serviço de rede social com mundo virtual.
- <http://www.gaiaonline.com> serviço de rede social com mundo virtual.
- <http://www.habbo.com.br> serviço de rede social com mundo virtual.
- <http://www.there.com/info/homepage> serviço de rede social com mundo virtual.
- <http://www.youtube.com> serviço de mídia social.
- <http://www.joost.com> serviço de TV na Web.
- <http://www.apple.com/br/itunes> serviço de venda músicas.
- <http://www.delicious.com> serviço de bookmarking.
- <http://www.stumbleupon.com> serviço de bookmarking.
- <http://www.laszlo systems.com> serviço de Webtop.
- <http://www.eyeos.org> serviço de Webtop.
- <http://www.startforce.com> serviço de Webtop.
- <http://workspace.officelive.com/pt-BR> serviço SaaS.
- <http://www.officelive.com> serviço SaaS.
- <http://basecamphq.com> serviço SaaS.
- <http://campfirenow.com> serviço SaaS de bate-papo em grupo.
- <http://tadalist.com> serviço SaaS de lista de tarefas.
- <http://www.housingmaps.com> serviço baseado em mashups.
- <http://www.programmableweb.com> serviços que disponibiliza mashups.
- <http://www.widgipedia.com> serviço que disponibiliza widgets.
- <http://aws.amazon.com> serviço que disponibiliza Web Services.
- <http://www.bittorrent.com/intl/pt/> serviço de compartilhamento de arquivos.
- <http://www.limewire.com/pt> serviço de compartilhamento de arquivos.
- <http://www.kazaa.com> serviço de compartilhamento de arquivos.
- <http://www.shareaza.com/> serviço de compartilhamento de arquivos.
- <http://sourceforge.net> distribuição de projetos open source.
- <http://freshmeat.net> distribuição de projetos open source.
- <http://www.tucows.com> distribuição de projetos open source.
- <https://www.paypal.com.br/br> serviço de pagamento Web.
- <http://www.netvibes.com/pt-br> serviço de página inicial.
- <http://www.google.com.br/ig> serviço de página inicial.
- <http://www.pageflakes.com> serviço de página inicial.
- <http://www.pageflakes.com> serviço de página inicial.
- <http://www.linkshare.com> serviço de afiliação.
- <http://www.register.com> serviço de registro de domínio.
- <http://www.skype.com> serviço de telefonia via Web.

<http://www.alvarofpinheiro.eti.br>

Finalizando, segue uma lista de software bastante utilizada:

- MediaWiki <http://www.mediawiki.org/wiki/MediaWiki> permite criar livremente seu Wiki.
- Dojo <http://dojotoolkit.org> toolkit para JavaScript.
- Flex <http://www.adobe.com/products/flex> framework de desenvolvimento RIA.
- JavaFX <http://javafx.com/> framework de desenvolvimento RIA.
- Silverlight <http://www.silverlight.net> framework de desenvolvimento RIA.
- Ruby on Rails <http://www.rubyonrails.pro.br> framework de desenvolvimento RIA.
- Script.aculo.us <http://script.aculo.us> framework de desenvolvimento RIA.
- JavaServer Faces <http://javaserverfaces.java.net> framework de desenvolvimento RIA.
- ASP.NET Ajax <http://www.asp.net/ajax/ajaxcontroltoolkit/samples> toolkit AJAX
- Adobe Integrated Runtime <http://www.adobe.com/products/air/> framework RIA.
- Google Gears <http://gears.google.com/> uso de aplicações Web off-line.

### 1.8 Futuro da Web

Folksonomia é a forma mais moderna de ontologia (estudo do ser humano), baseada na expressão que identifica a maneira de indexar informações ou mais especificadamente é à taxonomia (classificação) de pessoas, isto é, a indexação de multimídias sociais usando a linguagem natural como índice (tagging), permitindo que cada usuário da rede classifique o que lhe interessa com um ou mais palavras-chaves (tags). E por meio dessas é possível compartilhar e recuperar informações na rede. A grande vantagem do uso das tags (folksonomia) é a capacidade de divulgação dos conteúdos dos sites colaborativos, onde os próprios usuários é que se encarregam de fazer a classificações do conteúdo do site, minimizando muito o custo que uma empresa teria na contratação de mão de obra responsável por essa atividade, o que inviabilizaria muitos negócios, pois o tornaria financeiramente inviável, devido à grande quantidade de tempo e pessoal necessário para o mesmo. É importante também destacar que ao liberar a classificação de informação para o público, garante-se que o conteúdo será naturalmente relacionado a palavras que os usuários acreditam ter ligação com os mesmos. Não se corre o risco de que, por exemplo, o administrador do site, ao não conhecer bem um novo conteúdo publicado por um usuário, classifique-o de forma equivocada. Resumindo, na folksonomia, quem classifica o conteúdo são as próprias pessoas interessadas no mesmo. Exemplos de sites que fazem uso da folksonomia são o flickr e o YouTube, todos os sites Web 2.0 que trazem essa nova característica da Web 3.0.

*Web Ontology Language* (WOL) é uma linguagem que serve para definir e instanciar ontologias na Web, incluindo descrições de classes e suas respectivas propriedades e seus relacionamentos. Esse tipo de linguagem faz uso de várias tecnologias como XML e RDF.

Web Semântica é a forma mais atual da Web e que permite computadores e pessoas trabalharem colaborativamente, isto é, ligando significados de palavras (sentido) aos conteúdos publicados na Internet de modo que seja perceptível tanto pelo humano como pelo computador. Disponibilizando assim um novo formato de conteúdo para a Web que tem significado para computadores e pessoas. O objetivo principal da Web semântica não é fazer com que máquinas entendam a linguagem humana, mas sim, utilizar tecnologias e linguagens que tornem a informação legível para as máquinas, através de padrões definidos pela W3C, o uso da linguagem *extensible Markup Language* (XML), das tecnologias *Resource Description Framework* (RDF), das arquiteturas de meta-dados, de ontologias e dos agentes computacionais. Assim, o uso de serviços Web que garantam a interoperabilidade e cooperação se tornam mais eficientes e eficazes.

Microformatos é um conjunto de formatos abertos projetados para adicionar semântica em qualquer documento XML, especialmente HTML e XHTML. Isso é feito usando atributos do HTML específicos: class, rel, rev. As aplicações, como buscadores, podem extrair informações específicas de páginas que fazem uso de microformatos, como informações de contato, evento, licença, entre outras. Segue um exemplo, que informa que o link é seguro: `<a href="http://www.google.com/" rel="friend">Seguro</a>` Segue também algumas referências de microformatos: hCard - para

informação de contato; hCalendar - para eventos; hReview - para revisões; XFN - para relações sociais; e rel-license - para licenças.

*Resource Description Framework* (RDF) é uma linguagem para representar informação na Internet, nas quais os arquivos RDF são modelos ou fontes de dados, também conhecidos como metadados. Essa tecnologia é recomendada pela W3C, tendo como principal objetivo criar um modelo simples de dados, com uma semântica formal, usando XML. São 3 os componentes básicos de um arquivo RDF: recurso - qualquer coisa que pode conter um URL, incluindo as páginas da web, assim como elementos de um documento XML; propriedade - um recurso que tenha um determinado nome e possa ser utilizado como uma propriedade; e a indicação - consiste na combinação de um recurso, de uma propriedade, e de um valor.

RSS é um subconjunto de "dialetos" XML que servem para agregar conteúdo, podendo ser acessado por programas ou sites. A abreviatura do RSS é usada para se referir aos seguintes padrões: *Rich Site Summary* (RSS 0.91), *RDF Site Summary* (RSS 0.9 e 1.0) e *Really Simple Syndication* (RSS 2.0). A tecnologia do RSS permite aos usuários da internet se inscrever em sites que fornecem "feeds" e esses são tipicamente sites que mudam ou atualizam o seu conteúdo regularmente. Para isso, são utilizados *Feeds* RSS que recebem estas atualizações, desta maneira o utilizador pode permanecer informado de diversas atualizações em diversos sites sem precisar visitá-los um a um.

*eXtensible Hypertext Markup Language* (XHTML) é a reformulação da linguagem de marcação HTML, baseada em XML. Combina as tags de marcação HTML com regras da XML. Este processo de padronização tem em vista a exibição de páginas Web em diversos dispositivos (televisão, palm, celular, etc). Sua intenção é melhorar a acessibilidade. O XHTML consegue ser interpretado por qualquer dispositivo, independentemente da plataforma utilizada, pois as marcações possuem sentido semântico para as máquinas. O HTML não consegue esta implementação. No entanto, não existem muitas diferenças entre o HTML e o XHTML.

*eXtensible Markup Language* (XML) é uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais. Sendo capaz de descrever diversos tipos de dados. Seu propósito principal é a facilidade de compartilhamento de informações através da Internet.

Agora que temos a introdução do que é Web vamos conceituar no próximo capítulo a plataforma DOT NET.

### 2 Introdução à Dot Net

Neste capítulo serão introduzidos os conceitos que definem a tecnologia DOT NET para que o desenvolvedor possa conhecer um pouco os bastidores desse ambiente.

De acordo com a Microsoft o .NET é a “solução da Microsoft para Web Services, a próxima geração de software que conecta nosso mundo de informações, dispositivos e pessoas de uma maneira unificada e personalizada”.

Então a visão da Microsoft para essa plataforma é: Possibilitar o desenvolvimento mais rápido e mais fácil; Distribuir de forma simplificada de aplicações; Permitir o acesso a serviços e dados a qualquer momento e em qualquer lugar; e Fomentar aplicações colaborativas.

#### 2.1 Arquitetura

Para isso são disponibilizados os seguintes componentes: O *Common Language Runtime* (CLR) que é um ambiente de tempo de execução que fornece os serviços necessários a todas as aplicações .NET, sendo algum desses serviços a compilação, o gerenciamento de memória (*garbage collection*), o sistema comum de tipos, que permite o trabalho com qualquer linguagem escrita para .NET; O *Intermediate Language* (MSIL) que se trata de uma linguagem intermediária genérica comum; e o Just In Time (JIT) que é o compilador que transforma código intermediário em código nativo; Outro componente é o *Common Type System* (CTS) que trata do conjunto padronizado de tipos e operações dentro do CLR que suporta a implementação completa da maioria das linguagens de programação; O *Common Language Specification* (CLS) define um subconjunto de regras do CTS em torno de tipos e convenções para atribuição de nomes e outros atributos de implementação de linguagem que permitem uma portabilidade entre diferentes linguagens; O *Framework Class Library* (FCL) é uma biblioteca orientada a objetos composta de classes, interfaces, tipos e outros elementos para programação de aplicações .NET; Também existem os componentes de Bibliotecas de Classe; e o conjunto de *Assemblies*, como o ADO.NET, ASP.NET, VCL.NET.

Uma característica marcante desta plataforma é a facilidade de se adaptar as linguagens que dela fazem parte, pois o foco de aprendizagem se resume aos símbolos léxicos tradicionais das linguagens já que a grande parte dos comportamentos é padrão para qualquer linguagem DOT NET devido aos componentes acima relatados.

#### 2.2 Recursos

Veja o exemplo a seguir onde são exemplificados os símbolos padrões de três linguagens amplamente utilizadas nessa plataforma, os quais são, Delphi.NET, C# e VB.NET. Será possível perceber que os símbolos léxicos (*tokens*) são distintos de linguagem para linguagem, porém se formos usar métodos, como por exemplo, o de conversão para string, será o mesmo para todas as linguagens, exemplo ToString(). Essa é uma das grandes vantagens dessa plataforma.

Figura 2.2 Análise Léxica

| Operador                   | Delphi | C# | VB.NET |
|----------------------------|--------|----|--------|
| Atribuição                 | :=     | =  | =      |
| Comparação                 | =      | == | =      |
| Diferença                  | <>     | != | <>     |
| Menor que                  | <      | <  | <      |
| Maior que                  | >      | >  | >      |
| Menor ou igual a           | <=     | <= | <=     |
| Maior ou igual a           | >=     | >= | >=     |
| Lógico and                 | and    | && | and    |
| Lógico or                  | or     |    | or     |
| Lógico not                 | not    | !  | not    |
| Lógico xor                 | xor    | ^  | xor    |
| Adição                     | +      | +  | +      |
| Subtração                  | -      | -  | -      |
| Multiplicação              | *      | *  | *      |
| Divisão de Ponto Flutuante | /      | /  | /      |
| Divisão de Inteiro         | div    | /  | \      |
| Módulo                     | mod    | %  | Mod    |
| Potência exponencial       |        |    | ^      |

Fonte: Próprio Autor

No DOT NET tudo é um tipo por valor capaz de sofrer uma conversão implícita para uma classe, isso recebe o nome de boxing e unboxing, isto é, inteiro, string, ponto flutuante e todo o resto não é implementado como primitivas do compilador, eles são mapeados para tipos por valores fornecidos pelo DOT NET Framework. Assim para se saber o tamanho em bytes de um tipo ou variável usa-se o método `SizeOf()`.

Essa característica torna a curva de aprendizagem da linguagem muito baixa, já que o necessário é conhecer as primitivas básicas que praticamente é o que se difere, pois os tipos e os métodos são padronizados, facilitando a vida de quem desenvolve nesse ambiente, pois se já conhece alguma outra linguagem DOT NET já conhece boa parte das outras linguagens dessa plataforma. Veja a seguir os tipos comuns desse ambiente representado na coluna CLR.

Figura 2.3 Tipos

| Tipo                          | Delphi              | C#      | CLR             |
|-------------------------------|---------------------|---------|-----------------|
| Inteiro de 8 bits c/sinal     | Shortint            | sbyte   | System.SByte    |
| Inteiro de 8 bits s/sinal     | Byte                | byte    | System.Byte     |
| Inteiro de 16 bits c/sinal    | Smallint            | short   | System.Int16    |
| Inteiro de 16 bits s/sinal    | Word                | ushort  | System.UInt16   |
| Inteiro de 32 bits c/sinal    | Integer             | int     | System.Int32    |
| Inteiro de 32 bits s/sinal    | Cardinal            | uint    | System.UInt32   |
| Inteiro de 64 bits c/sinal    | Int64               | long    | System.Int64    |
| Inteiro de 64 bits s/sinal    | UInt64              | ulong   | System.UInt64   |
| Flutuante de precisão simples | Single, Float       | Float   | System.Single   |
| Flutuante de precisão dupla   | Double              | double  | System.Double   |
| Decimal fixo                  |                     | decimal | System.Decimal  |
| Decimal fixo Delphi           | Currency            |         |                 |
| Data/hora                     | Datetime            |         | System.DateTime |
| Variante                      | Variant, olevariant |         |                 |
| Caracter de 1 byte            | AnsiChar            |         |                 |
| Caracter de 2 bytes           | Char, WideChar      | char    | System.Char     |
| String fixo                   | ShortString         |         |                 |
| String de 1 byte dinâmico     | AnsiString          |         |                 |
| String de 2 bytes dinâmico    | String, WideString  | string  | System.String   |
| Booleano                      | Boolean             | bool    | System.Boolean  |

Fonte: Próprio Autor

Agora vamos entender o que é ASP.NET ou *Active Server Page Dot Net*.

## 2.3 Introdução ao ASPX

O que é ASP.NET ou *Active Server Page Dot Net*. São páginas de servidor ativas, isto é, são recursos de programação por scripts no servidor, que se pode usar para criar páginas dinâmicas, interativas e de alta performance.

As páginas ASP, os scripts que rodam no servidor e não no cliente. É o próprio servidor ou melhor o *Internet Information Server* (IIS) que transforma os scripts em HTML padrão, fazendo com que qualquer browser do mercado seja capaz de acessar um site que usa ASPc

## 2.4 Recursos do ASPX

Entre os recursos que podem ser implementados via ASP, podemos citar: A programação em JavaScript; O acesso a banco de dados; As sessões ou melhor dizendo, as persistências de informações no servidor.

O ASP surgiu juntamente com o lançamento do *Internet Information Server* versão 3. Esta é uma solução Microsoft, que exige que o seu servidor rode em um sistema operacional da Microsoft. E são os seguintes servidores que suportam o uso de páginas ASP: *Microsoft Internet Information Server*; *Microsoft Peer Web Services*; *Microsoft Personal Web Server*. A grande vantagem, porém, é que existe esta exigência apenas do lado do servidor. No lado do cliente, é possível utilizar qualquer browser, gerando a independência do navegador.

O ASP executa páginas complexas no servidor gerando páginas simples para enviar os resultados para o cliente (browser). Essas páginas "complexas" podem manipular bancos de dados, permitindo visualizar, atualizar e adicionar dados nos servidores SQL. Outra característica das páginas web que rodam no servidor são a segurança do código fonte, como o servidor retorna somente o resultado

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

HTML, o código fonte, isto é, a lógica fica protegida. Além disso, o ASP pode utilizar comandos JavaScript e HTML.

O ASP suporta várias linguagens script, por exemplo, suporte nativo para JScript, que é o JavaScript e VBScript, ambos da Microsoft. Possibilidade de baixar plug-ins ActiveX que possibilita suporte para outras linguagens como REXX, PERL, Python, etc.

Em relação aos browsers, o ASP suporta todos, isto é possível pelo fato de que as páginas ASP serem processadas no servidor e convertidas em HTML, o que o cliente (browser) interpreta.

Mais uma característica excelente das páginas ASP é a capacidade de guardar os estados dos objetos, isto é, os valores dos campos. As aplicações ASP podem armazenar dados que é mantido durante toda uma sessão, desta forma, um usuário pode fornecer seu nome somente uma vez em uma página e as demais páginas podem obter este dado (estado) automaticamente.

Outro aspecto é a segurança fornecida pelas páginas ASP, sendo esse aspecto parte integrante do servidor Web da Microsoft o *Internet Information Server* (IIS), que tem sua segurança integrada com o Windows. Sendo assim, é possível facilmente restringir o acesso as páginas ASP usando os esquemas de autenticação do IIS que consiste de senha básica da Web ou senha do Windows ou ainda certificados de cliente.

As páginas Web usam basicamente os WebForms, isto é, os formulários Web, que são o caminho natural para os usuários enviarem informações para o ASP. Os formulários podem ser HTML ou ASP dependendo das capacidades dinâmicas desejadas. O Formulário deve passar variáveis para o arquivo ASP para processamento de dados. Esses formulários podem passar dados tanto na forma GET como na forma POST, o desenvolvedor escolhe a que desejar e ambas as formas são simples de codificar.

O ASP possui vários componentes e um dos mais utilizados é o componente TextBox que quando convertido para HTML se torna o `<INPUT NAME="UltimoNome">`. Este comando criará um input box com o tamanho default e o browser passará para o arquivo output em ASP a variável com o nome UltimoNome. No capítulo desenvolvendo aplicação Web serão mostrados os componentes com detalhes, mas antes será realizada uma introdução aos conceitos do paradigma orientado a objetos para que se inicie com a linguagem C#.

### 3 Dicas do ASPX e CSS

Bom, aqui vamos começar com as dicas de código da linguagem de marcação ASPX e CSS.

ASP.NET como dito inicialmente é a linguagem de marcação da Microsoft para o desenvolvimento de aplicações Web sendo a sucessora do ASP. Permite através da integração com a linguagem de programação .NET criar páginas dinâmicas. Essa linguagem é suportada pelo Framework .NET herdando todas as características deste, por isso permitindo o uso de várias linguagens de programação como C#, F#, Visual Basic .NET, entre muitas outras.

É possível desenvolver aplicações ASP.NET utilizando somente um bloco de notas e o compilador .NET, porém o ambiente de desenvolvimento mais comum das aplicações ASP.NET é com a IDE chamada de Visual Studio .NET a qual possui diversas características que simplificam o trabalho do desenvolvedor, tais como componentes visuais e funcionais.

Uma característica marcante desta plataforma é que uma aplicação ASP.NET pode reutilizar código de escrito em qualquer linguagem .NET. Uma página ASP.NET escrita em VB.NET pode chamar componentes escritos em C# ou Web Services escritos em C++, pois as aplicações ASP.NET são pré-compiladas antes da execução.

As aplicações Web ASP.NET fazem uso do Framework .NET e do servidor IIS, mas existe o projeto mod\_aspdotnet que permite a utilização do servidor Apache HTTP Server e também o projeto Mono é um esforço para permitir que aplicações ASP.NET possam executar na plataforma Linux.

CSS significa *Cascading Style Sheets* onde se define os estilos das páginas web para efeitos de transição, imagem, entre muitos outros. A principal função do CSS é retirar as imagens de *background* e disponibilizar efeitos de animações de forma simples, mas isso depende do uso de *browsers* atualizados e com suporte à essa linguagem, encontrados no Google Chrome, Opera, Internet Explorer 9, Safari e Mozilla Firefox.

#### 3.1 Dica: Adicionar arquivo CSS a uma página ASPX.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <link href="arquivo.css" rel="stylesheet" type="text/css" />
</head>
```

#### 3.2 Dica: Adicionar um ícone a uma página ASPX.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <link rel="shortcut icon" href="http://algun/site/som/arquivo.ico"
type="image/x-icon" />
</head>
```

#### 3.3 Dica: Adicionar título a uma página ASPX.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
    <title>Titulo</title>
</head>
```

### 3.4 Dica: Escrever uma função JavaScript para abrir uma URL numa página ASPX.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Titulo</title>
</head>
    <script language="javascript" type="text/javascript">
        function abrir(URL) {
            var width = 150;
            var height = 250;
            var left = 99;
            var top = 99;
            window.open(URL, 'janela', 'width=' + width + ', height=' +
height + ', top=' + top + ', left=' + left + ', scrollbars=yes, status=no,
toolbar=no, location=no, directories=no, menubar=no, resizable=no,
fullscreen=no');
        }
    </script>
</head>
```

### 3.5 Dica: Chamar uma função JavaScript em uma evento ASPX.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Titulo</title>
</head>
</head>
<body onunload="finaliza();">
    <form id="form" runat="server">
        <div style="text-align: center">
            </div>
    </form>
</body>
```

### 3.6 Dica: Carregar estilos de páginas para arredondar bordas de objetos ASPX.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Titulo</title>
</head>
    <script language="javascript" type="text/javascript">
</script>
    <style type="text/css">
        .styleCaixaCentrolArredondadoTopo
        {
            -webkit-border-top-left-radius: 12px;
            -webkit-border-top-right-radius: 12px;
            text-align: center;
        }
    </style>
</head>
```

```

        vertical-align: top;
        border-style: solid;
        border-width: 1px;
        border-color: #468DC1;
    }
    .styleCaixaCentro1ArredondadoBaixo
    {
        -webkit-border-bottom-left-radius: 12px;
        -webkit-border-bottom-right-radius: 12px;
        text-align: center;
        vertical-align: top;
        border-style: solid;
        border-width: 1px;
        border-color: #468DC1;
    }
</style>
</head>

```

### 3.7 Dica: Carregar estilos de páginas para alinhar objetos.

```

<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Titulo</title>
</head>
    <script language="javascript" type="text/javascript">
</script>
    <style type="text/css">
        .styleAlinhaEsquerda
        {
            text-align: left;
            vertical-align: middle;
        }
        .styleAlinhaCentro
        {
            text-align: center;
            vertical-align: middle;
        }
        .styleAlinhaDireita
        {
            text-align: right;
            vertical-align: middle;
        }
    </style>
</head>

```

### 3.8 Dica: Carregar estilos de páginas para modificar objetos ASPX.

```

<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Titulo</title>
</head>
    <style type="text/css">
        .styleTexto
        {

```

```

        -moz-border-radius: 10px;
        -webkit-border-radius: 10px;
        font-family: Tahoma;
        font-size: 10px;
        background-color: White;
        color: Black;
        text-align: left;
        vertical-align: Top;
    }
    .styleBotao
    {
        -moz-border-radius: 8px;
        -webkit-border-radius: 8px;
        background-color: #1B5476;
        font-family: Tahoma;
        font-size: 12px;
        font-weight: bold;
        color: White;
        cursor: pointer;
    }
</style>
</head>

```

### 3.9 Dica: Criar menu suspenso com CSS.

```

<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Titulo</title>
</head>
    <style>
        a, a:hover {text-decoration: none;}
        ul#navWrapper {border: 0 black dashed;margin-left: -39px;float:
left;}
        ul#navWrapper li {border: 0 red dashed;float: left;list-style:
none;margin-right: 0.75em;background-color: #DDD;padding: 0
0.25em;border-radius: 4px;box-shadow: 3px 3px 6px 1px #333;}
        ul#navWrapper li li {border: 0 blue dashed;float: none;margin-
left: -44px;margin-top: 3px;}
        ul#navWrapper li li:first-child {margin-top: 4px;}
        ul#navWrapper ul {display: none;position: absolute;background-
color: #FFF;background-color: rgba(255, 255, 255, 0);}
        ul#navWrapper li:hover ul {display: block;}
        ul#navWrapper a {font-weight: bold;}
        ul#navWrapper li:hover {background-color: #8C8D61;}
        div#Banner {border: 1px black solid;border-radius: 4px;clear:
both;height: 4em;background-color: rgb(85, 126, 185);background-
image: -webkit-radial-gradient(100% 0%, circle cover,
rgb(104,24,136) 0%, rgb(85,126,185) 100%);background-image: radial-
gradient(circle at 100% 0%, rgb(104,24,136) 0%, rgb(85,126,185)
100%);}
    </style>
</head>
<body>
    <form id="form" runat="server">
        <div style="text-align: center">
            <ul id="Menu">

```

```
<!-- Menu 1 -->
<li>
  <a href="pag1.aspx" aria-haspopup="true">Menu1</a>
  <ul>
    <li><a href="pag1a.aspx">Item1</a></li>
    <li><a href="pag1b.aspx">Item2</a></li>
  </ul>
</li>
<!-- Menu B -->
<li>
  <a href="pageB.html" aria-haspopup="true">Menu2</a>
  <ul>
    <li><a href="pag2a.aspx">Item3</a></li>
    <li><a href="pag2b.aspx">Item4</a></li>
  </ul>
</li>
</ul>
</div>
</form>
</body>
```

### 3.10 Dica: Criar estilo CSS com bordas arredondadas.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>

<style type="text/css">
  .styleX
  {
    border-radius: 25px;
    background: url(paper.gif);
    background-position: left top;
    background-repeat: repeat;
    padding: 20px;
    width: 200px;
    height: 150px;
  }
</style>
</head>
```

### 3.11 Dica: Criar estilo CSS com bordas com imagem.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>

<style type="text/css">
  .styleX
  {
    border: 10px solid transparent;
    padding: 15px;
    border-image: url(border.png) 30 round;
  }
</style>
</head>
```

**3.12 Dica: Criar estilo CSS com background.**

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>

<style type="text/css">
    .styleX
    {
        background-image: url(img_flwr.gif), url(paper.gif);
        background-position: right bottom, left top;
        background-repeat: no-repeat, repeat;
    }
</style>

</head>
```

**3.13 Dica: Criar estilo CSS com definição de cores.**

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>

<style type="text/css">
    .styleX
    {
        background-color: rgba(255, 0, 0, 0.3);} /* vermelho */
        background-color: rgba(0, 255, 0, 0.3);} /* verde */
        background-color: rgba(0, 0, 255, 0.3);} /* azul */
    }
</style>

</head>
```

**3.14 Dica: Criar estilo CSS com difinição de cores gradiente.**

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>

<style type="text/css">
    .styleX
    {
        background: linear-gradient(red, yellow);
    }
</style>

</head>
```

**3.15 Dica: Criar estilo CSS com sombreamento.**

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>

<style type="text/css">
    .styleX
```

```

{
text-shadow: 2px 2px 5px red;
}
</style>
</head>

```

### 3.16 Dica: Criar estilo CSS com ou sem quebra de palavra.

```

<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<style type="text/css">
.styleSemQuebra
{
word-break: keep-all;
}
.styleComQuebra
{
word-break: break-all;
}
</style>
</head>

```

### 3.17 Dica: Criar estilo CSS com definição de fonte.

```

<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<style type="text/css">
.styleX
{
font-family: myFirstFont;
}
</style>
</head>

```

### 3.18 Dica: Criar estilo CSS com rotação de um elemento.

```

<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<style type="text/css">
.styleX
{
/*
matrix(n,n,n,n,n,n) Define translação 2D usando uma matriz de 6
valores
translate(x,y) Define translação 2D movendo o elemento ao longo do
eixo X e Y
translateX(n) Define translação 2D movendo o elemento ao longo do
eixo X

```

```
        translateY(n) Define translação 2D movendo o elemento ao longo do
eixo Y
        scale(x,y) Define translação 2D trocando a altura e tamanho do
elemento
        scaleX(n) Define translação 2D trocando o tamanho do elemento
        scaleY(n) Define translação 2D trocando a altura do elemento
        rotate(angle) Define translação 2D rotacionando o elemento em
angulação
        skew(x-angle,y-angle) Define translação 2D trocando o elemento no
eixo X e Y
        skewX(angle) Define translação 2D trocando o elemento no eixo X
        skewY(angle) Define translação 2D trocando o elemento no eixo Y
        */
        transform: rotate(20deg);
    }
</style>
</head>
```

### 3.19 Dica: Criar estilo CSS com efeito de transição.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<style type="text/css">
    .styleX
    {
        transition-property: width;
        transition-duration: 2s;
        transition-timing-function: linear;
        transition-delay: 1s;
    }
</style>
</head>
```

### 3.20 Dica: Criar estilo CSS com efeito de animação.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<style type="text/css">
    .styleX
    {
        width: 100px;
        height: 100px;
        position: relative;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
        animation-delay: 2s;
    }
</style>
</head>
```

**3.21 Dica: Criar estilo CSS com efeito de transparência.**

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<style type="text/css">
    .styleX
    {
        -webkit-filter: grayscale(100%);
    }
</style>
</head>
```

**3.22 Dica: Criar estilo CSS com definição de duração de um efeito.**

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<style type="text/css">
    .styleX
    {
        transition-duration: 0.4s;
        background-color: #4CAF50;
        color: white;
    }
</style>
</head>
```

**3.23 Dica: Criar estilo CSS com definição de múltiplas colunas.**

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<style type="text/css">
    .styleX
    {
        -webkit-column-count: 3;
    }
</style>
</head>
```

**3.24 Dica: Criar estilo CSS com efeito de alinhamento container.**

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>
<style type="text/css">
    .styleX
    {
        width: 70px;
    }
</style>
</head>
```

```
        height: 300px;
        border: 1px solid #c3c3c3;
        display: -webkit-flex;
        display: flex;
        -webkit-flex-wrap: wrap;
        flex-wrap: wrap;
        -webkit-align-content: center;
        align-content: center;
    }
</style>
</head>
```

### 3.25 Dica: Criar estilo CSS com efeito de alinhamento de itens de um container.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>

    <style type="text/css">
        .styleX
        {
            display: flex;
            align-items: center;
        }
    </style>

</head>
```

### 3.26 Dica: Criar estilo CSS com efeito de alinhamento de itens de um container.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
</head>

    <style type="text/css">
        .styleX
        {
            display: flex;
            align-items: center;
        }
    </style>

</head>
```

### 4 Dicas do JavaScript

JavaScript é uma linguagem de programação interpretada que foi concebida para os browsers pudessem executar lógica de programação no lado do cliente e assim interagissem com o usuário sem a necessidade de submeter ao servidor a lógica. É atualmente a principal linguagem de programação *client-side* para os navegadores Web. Ela foi concebida como uma linguagem script no paradigma orientação a objetos, com tipagem fraca.

Seguem as dicas da linguagem JavaScript:

#### 4.1 Dica: Retornar as dimensões (altura e largura) de uma página cliente.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Titulo</title>
    <script language="JavaScript" type="text/javascript">
        var alturaTela;
        var larguraTela;
        function Dimensoes() {
            if (screen) {
                alturaTela = screen.height;
                larguraTela = screen.width;
            }
            else {
                alturaTela = escape("Não identificado")
                larguraTela = escape("Não identificado")
            }
            document.getElementById('TextBoxDimensoes').value = alturaTela
+ ";" + larguraTela;
            // computador a=1080 l=1920
            // celular a=640 l=360
            if (alturaTela < 700) {
                window.location = "http://www.site.com/pagina.aspx";
            }
            return (0);
        }
    </script>
</head>
```

#### 4.2 Dica: Adicionar arquivo de som a uma página ASPX.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Titulo</title>
    <script language="javascript" type="text/javascript">
        function som() {
            embed = document.createElement("embed");
            embed.setAttribute("src",
"http://algun/site/som/arquivo.wav");
            embed.setAttribute("hidden", true);
            embed.setAttribute("autostart", true);
            document.body.appendChild(embed);
        }
    </script>
</head>
```

```
</script>
</head>
```

#### 4.3 Dica: Adicionar uma função de alerta popup página ASPX.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Titulo</title>
    <script language="javascript" type="text/javascript">
        function ExibeAlerta(msg) {
            window.alert(msg);
            return false;
        }
    </script>
</head>
```

#### 4.4 Dica: Adicionar uma função de impressão numa página ASPX.

```
<%@ Master Language="C#" CodeFile="Default.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Titulo</title>
    <script language="javascript">
        function Imprimir(identificador) {
            var imprime = document.getElementById(identificador);
            var janela = window.open('', '',
'letf=0,top=0,width=1,height=1,toolbar=0,scrollbars=0,status=0');
            janela.document.write(imprime.innerHTML);
            janela.document.close();
            janela.focus();
            janela.print();
            janela.close();
            imprime.innerHTML = strOldOne;
        }
    </script>
</head>
```

#### 4.5 Dica: Como resolver mensagem de erro no uso do Timer com o UpdatePanel.

```
<asp:ScriptManager
    ID="ScriptManager" runat="server" AsyncPostBackTimeout="0">
</asp:ScriptManager>
// Inserir script após o ScriptManager
<script type="text/javascript" language="javascript">

Sys.WebForms.PageRequestManager.getInstance().add_endRequest(EndRequestHan
dler);
function EndRequestHandler(sender, args){
    if (args.get_error() != undefined){
        args.set_errorHandled(true);
    }
}
</script>
```

## 5 Dicas do CSharp

CSharp é uma linguagem de programação parte compilada (bytecode) e parte interpretada (máquina virtual Common Language Runtime), é fortemente tipada e baseada nos paradigmas de programação imperativa, funcional, declarativa, orientada a objetos. Foi desenvolvida e é mantida pela Microsoft como parte da plataforma .NET. A sua sintaxe orientada a objetos é baseada no C++ contendo forte influência das linguagens Object Pascal (Delphi) e Java.

Seguem as dicas da linguagem CSharp:

### 5.1 Dica: Rotina para Capitalizar a primeira letra de cada nome de uma cadeia de caracteres.

```
public partial class _Default : System.Web.UI.Page
{
    protected static string CapitalizarString(string pTexto)
    {
        String sRet = "";
        try
        {
            pTexto = pTexto.ToLower();
            System.Collections.ArrayList alAux =
Geral.StringToArray(pTexto, " ", 1);
            String sAux = "";
            for (int i = 0; i <= alAux.Count - 1; i++)
            {
                if (alAux[i].ToString() != "")
                {
                    sAux = Geral.Capitalizar(alAux[i].ToString());
                    if (((sAux.IndexOf("Da") != -1) ||
(sAux.IndexOf("De") != -1) || (sAux.IndexOf("Di") != -1) ||
(sAux.IndexOf("Do") != -1) || (sAux.IndexOf("Du") != -1)) && (sAux.Length
== 2)))
                    {
                        sAux = sAux.ToLower();
                    }
                    else if (((sAux.IndexOf("Ao") != -1)) && (sAux.Length
== 2)))
                    {
                        sAux = sAux.ToLower();
                    }
                    else if (((sAux.IndexOf("A") != -1) ||
(sAux.IndexOf("E") != -1) || (sAux.IndexOf("I") != -1) ||
(sAux.IndexOf("O") != -1) || (sAux.IndexOf("U") != -1)) && (sAux.Length ==
1)))
                    {
                        sAux = sAux.ToLower();
                    }
                }
                if (sRet == "")
                {
                    sRet = sRet + sAux;
                }
                else
                {
                    sRet = sRet + " " + sAux;
                }
            }
        }
    }
}
```

```
        catch
        {
        }
        return (sRet);
    }
}
```

### **5.2 Dica: Rotina para enviar e-mail pelo SMTP do Gmail.**

```
public partial class _Default : System.Web.UI.Page
{
    public static void GEMail(String pRemetente, String pSenha, String
pPara, String pComCopia, String pTitulo, String pCorpo)
    {
        try
        {
            System.Net.Mail.SmtpClient cliente = new
System.Net.Mail.SmtpClient("smtp.gmail.com", 587);
            cliente.EnableSsl = true;
            System.Net.Mail.MailAddress remetente = new
System.Net.Mail.MailAddress(pRemetente, pTitulo);
            System.Net.Mail.MailAddress destinatario = new
System.Net.Mail.MailAddress(pPara, pComCopia);
            System.Net.Mail.MailMessage mensagem = new
System.Net.Mail.MailMessage(remetente, destinatario);
            mensagem.Body = pCorpo;
            mensagem.Subject = pTitulo;
            System.Net.NetworkCredential credenciais = new
System.Net.NetworkCredential(pRemetente, pSenha, "");
            cliente.Credentials = credenciais;
            cliente.Send(mensagem);
        }
        catch (Exception ex)
        {
            throw (ex);
        }
    }
}
```

### **5.3 Dica: Importar um arquivo armazenando seu binário no banco de dados.**

```
public partial class _Default : System.Web.UI.Page
{
    protected void Importar(FileUpload FileImportar, String
ExtensoesPermitidas, String ComandosSQL, String ID)
    {
        String sNomeArquivo = "";
        String sArquivoCaminho = "";
        String sExtensao = "";
        String sFormato = "";
        long nTamanho = 0;
        try
        {
            sArquivoCaminho = FileImportar.PostedFile.FileName.Trim();
            sNomeArquivo = Path.GetFileName(sArquivoCaminho);
            if (sNomeArquivo == "")
            {
                Geral.Alerta(this, "Informe o arquivo para importar!",
"Alerta");
            }
        }
    }
}
```

```
else
{
    sExtensao = Path.GetExtension(sArquivoCaminho).ToLower();
    if (sExtensao.IndexOf(ExtensoesPermitidas) == -1)
    {
        Geral.Alerta(this, "Tipo de arquivo não permitido",
"Alerta");
    }
    else
    {
        nTamanho = FileImportar.PostedFile.InputStream.Length;
        switch (sExtensao)
        {
            case ".doc":
                sFormato = "application/vnd.ms-word";
                break;
            case ".docx":
                sFormato = "application/vnd.ms-word";
                break;
            case ".xls":
                sFormato = "application/vnd.ms-excel";
                break;
            case ".xlsx":
                sFormato = "application/vnd.ms-excel";
                break;
            case ".bmp":
                sFormato = "image/bmp";
                break;
            case ".jpg":
                sFormato = "image/jpg";
                break;
            case ".png":
                sFormato = "image/png";
                break;
            case ".gif":
                sFormato = "image/gif";
                break;
            case ".pdf":
                sFormato = "application/pdf";
                break;
            case ".txt":
                sFormato = "application/txt";
                break;
        }
        Stream oStream = FileImportar.PostedFile.InputStream;
        BinaryReader oBinario = new BinaryReader(oStream);
        Byte[] bytes =
oBinario.ReadBytes((Int32)oStream.Length);
        SqlConnection oSqlCnx = new SqlConnection();
        SqlCommand oSqlCommand = new SqlCommand(ComandoSQL);
        oSqlCommand.Parameters.Add("@ARQUIVO",
SqlDbType.VarChar).Value = sNomeArquivo;
        oSqlCommand.Parameters.Add("@FORMATO",
SqlDbType.VarChar).Value = sFormato;
        oSqlCommand.Parameters.Add("@TAMANHO",
SqlDbType.Int).Value = nTamanho;
        oSqlCommand.Parameters.Add("@DADOS",
SqlDbType.Binary).Value = bytes;
```

```
oSqlCommand.Parameters.Add("@ID", SqlDbType.Int).Value =
Convert.ToInt32(ID);
String sSqlCnx =
ConfigurationManager.AppSettings["BancoConexao"].ToString();
oSqlCnx.ConnectionString = sSqlCnx;
oSqlCommand.CommandType = CommandType.Text;
oSqlCommand.Connection = oSqlCnx;
oSqlCnx.Open();
oSqlCommand.ExecuteNonQuery();
oSqlCnx.Close();
oSqlCnx.Dispose();
    }
}
}
catch (Exception ex)
{
}
}
}
```

### 5.4 Dica: Definir tema (skin) de página na pré-inicialização.

```
public partial class _Default : System.Web.UI.Page
{
    void Page_PreInit(object sender, System.EventArgs args)
    {
        try
        {
            if (Session["SkinFile"].ToString().Trim() != "")
            {
                Page.Theme = Session["SkinFile"].ToString();
            }
        }
        catch
        {
        }
    }
}
```

### 5.5 Dica: Imprimir dados do buffer de impressão

```
public partial class _Default : System.Web.UI.Page
{
    void Page_PreInit(object sender, System.EventArgs args)
    {
        string sGrupo = "";
        try
        {
            sGrupo = RepositorioBase.DadosConsultar("1", "SELECT
MIN(NU_BDI_GRUPO) FROM BUFFERIMPRESSAO").Rows[0][0].ToString();
            if (sGrupo == "0")
            {
                sGrupo = "1";
            }
        }
        catch
        {
            sGrupo = "1";
        }
        string sLinha = "";
    }
}
```

```

try
{
    if (sGrupo != "")
    {
        DataTable dtLinha = new DataTable();
        dtLinha = RepositorioBase.DadosConsultar("1", "SELECT
VC_BDI_LINHA FROM BUFFERIMPRESSAO WHERE NU_BDI_GRUPO = " + sGrupo);
        if (dtLinha.Rows.Count != 0)
        {
            for (int nLinha = 0; nLinha <= dtLinha.Rows.Count - 1;
nLinha++)
            {
                try
                {
                    sLinha = dtLinha.Rows[nLinha][0].ToString();
                    ArrayList alDados = new ArrayList();
                    alDados = Geral.StringToArray(sLinha, ";", 1);
                    System.Net.Mail.SmtpClient cliente = new
System.Net.Mail.SmtpClient("smtp.gmail.com", 587);
                    cliente.EnableSsl = true;
                    System.Net.Mail.MailAddress remetente = new
System.Net.Mail.MailAddress(email, assunto);
                    System.Net.Mail.MailAddress destinatario = new
System.Net.Mail.MailAddress(alDados[0].ToString().Trim(),
alDados[0].ToString().Trim());
                    System.Net.Mail.MailMessage mensagem = new
System.Net.Mail.MailMessage(remetente, destinatario);
                    mensagem.Body = alDados[2].ToString();
                    mensagem.Subject =
alDados[1].ToString().Trim();
                    System.Net.NetworkCredential credenciais = new
System.Net.NetworkCredential(email, senha, "");
                    cliente.Credentials = credenciais;
                    cliente.Send(mensagem);
                }
                catch
                {
                }
            }
            RepositorioBase.DadosManter("1", "DELETE FROM
BUFFERIMPRESSAO WHERE NU_BDI_GRUPO = " + sGrupo);
        }
    }
    catch
    {
    }
}
}

```

### 5.6 Dica: Definir estilo dos objetos na carga da página.

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {

```

```

        Button1.Attributes.Add("onMouseOver",
"this.style.backgroundColor='#99CCCC';this.style.cursor='hand';this.style.
textDecoration='underline'");
        Button1.Attributes.Add("onMouseOut",
"this.style.backgroundColor='';this.style.cursor='point';this.style.textDe
coration='none'");
    }
    catch
    {
    }
}
}

```

### 5.7 Dica: Obter dados do navegador (Browser) na carga da página.

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            System.Web.HttpBrowserCapabilities browser = Request.Browser;
            String[] DadosNavegador = new String[17];
            DadosNavegador[0] = browser.Type;
            DadosNavegador[1] = browser.Browser;
            DadosNavegador[2] = browser.Version;
            DadosNavegador[3] = browser.MajorVersion.ToString();
            DadosNavegador[4] = browser.MinorVersion.ToString();
            DadosNavegador[5] = browser.Beta.ToString();
            DadosNavegador[6] = browser.Crawler.ToString();
            DadosNavegador[7] = browser.AOL.ToString();
            DadosNavegador[8] = browser.Win16.ToString();
            DadosNavegador[9] = browser.Win32.ToString();
            DadosNavegador[10] = browser.Frames.ToString();
            DadosNavegador[11] = browser.Tables.ToString();
            DadosNavegador[12] = browser.Cookies.ToString();
            DadosNavegador[13] = browser.VBScript.ToString();
            DadosNavegador[14] = browser.EcmaScriptVersion.ToString();
            DadosNavegador[15] = browser.JavaApplets.ToString();
            DadosNavegador[16] = browser.ActiveXControls.ToString();
            Session["Navegador"] = "Type:" + DadosNavegador[0] +
";Browser:" + DadosNavegador[1] + ";Version:" + browser.Version;
            HyperLinkVersao.ToolTip = Session["Navegador"].ToString();
            if (DadosNavegador[0].IndexOf("Desktop") != -1) // Chrome
            {
            }
            else if (DadosNavegador[0].IndexOf("IE") != -1) // IE
            {
            }
            else if (DadosNavegador[0].IndexOf("Firefox") != -1) //
Mozilla Firefox
            {
            }
            else //// IE modo não compatibilidade e outros navegadores
            {
            }
        }
        catch
        {
        }
    }
}

```

```

    }
}

```

### 5.8 Dica: Obter usuário logado na rede autenticada windows.

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        int nPos = 0;
        Session["Usuario"] = "Usuário";
        string sUsuario = this.Context.User.Identity.Name;
        if (sUsuario.Length != 0)
        {
            nPos = sUsuario.IndexOf("\\");
            if (nPos == -1)
            {
                nPos = sUsuario.IndexOf("/");
            }
            else
            {
                nPos = nPos + 1;
            }
            if (nPos == -1)
            {
                Session["Usuario"] = sUsuario;
            }
            else
            {
                Session["Usuario"] = sUsuario.Substring(nPos);
            }
        }
    }
}

```

### 5.9 Dica: Configurar uma janela popup.

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            try
            {
                popupwin.Visible = false;
                String sSQLPopup = "SELECT
ID_POP,VC_POP_TITULO,TX_POP_MENSAGEM,NU_POP_TAMANHO,NU_POP_ALTURA,NU_POP_E
SQUERDA,NU_POP_DIREITA,NU_POP_MOSTRARAPOS,NU_POP_ESCONDERAPOS,NU_POP_VELOC
IDADEDOPOPUP,NU_POP_WINDOWSCROLL,VC_POP_DOCKMODE,VC_POP_COR,NU_POP_ATIVO
FROM POPUPS WHERE (NU_POP_ATIVO=1) OR ('" +
DateTime.Now.Date.ToString("yyyy/MM/dd") + "' BETWEEN DT_POP_INICIO AND
DT_POP_TERMINO) ORDER BY ID_POP DESC";
                DataTable dtPopup = RepositorioBase.DadosConsultar("1",
sSQLPopup);
                if (dtPopup.Rows.Count != 0)
                {
                    for (int nPopup = 0; nPopup <= dtPopup.Rows.Count - 1;
nPopup++)

```

```

        {
            (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).Title =
dtPopup.Rows[nPopup]["VC_POP_TITULO"].ToString();
            (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).Message =
dtPopup.Rows[nPopup]["TX_POP_MENSAGEM"].ToString();
            (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).Width =
Convert.ToInt32(dtPopup.Rows[nPopup]["NU_POP_TAMANHO"].ToString());
            (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).Height =
Convert.ToInt32(dtPopup.Rows[nPopup]["NU_POP_ALTURA"].ToString());
            (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).OffsetX =
Convert.ToInt32(dtPopup.Rows[nPopup]["NU_POP_ESQUERDA"].ToString());
            (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).OffsetY =
Convert.ToInt32(dtPopup.Rows[nPopup]["NU_POP_DIREITA"].ToString());
            (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).ShowAfter =
Convert.ToInt32(dtPopup.Rows[nPopup]["NU_POP_MOSTRARAPÓS"].ToString());
            (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).HideAfter =
Convert.ToInt32(dtPopup.Rows[nPopup]["NU_POP_ESCONDERAPÓS"].ToString());
            (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).PopupSpeed =
Convert.ToInt32(dtPopup.Rows[nPopup]["NU_POP_VELOCIDADEDOPOPUP"].ToString(
));
            (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).WindowScroll =
(dtPopup.Rows[nPopup]["NU_POP_WINDOWSCROLL"].ToString() == "1");
            if
(dtPopup.Rows[nPopup]["VC_POP_DOCKMODE"].ToString() == "BottomLeft")
            {
                (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).DockMode =
EekSoft.Web.PopupDocking.BottomLeft;
            }
            else
            {
                (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).DockMode =
EekSoft.Web.PopupDocking.BottomRight;
            }
            if (dtPopup.Rows[nPopup]["VC_POP_COR"].ToString()
== "Azul")
            {
                (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).ColorStyle =
EekSoft.Web.PopupColorStyle.Blue;
            }
            else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Verde")
            {
                (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EekSoft.Web.PopupWin).ColorStyle =
EekSoft.Web.PopupColorStyle.Green;
            }
        }
    
```

```
                else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Vermelho")
                {
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).ColorStyle =
EeekSoft.Web.PopupColorStyle.Red;
                }
                else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Violeta")
                {
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).ColorStyle =
EeekSoft.Web.PopupColorStyle.Violet;
                }
                else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Amarelo")
                {
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).ColorStyle =
EeekSoft.Web.PopupColorStyle.Custom;
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).Shadow =
System.Drawing.Color.Yellow;
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).GradientDark =
System.Drawing.Color.Yellow;
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).TextColor =
System.Drawing.Color.Black;
                }
                else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Preto")
                {
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).ColorStyle =
EeekSoft.Web.PopupColorStyle.Custom;
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).Shadow =
System.Drawing.Color.Black;
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).GradientDark =
System.Drawing.Color.Black;
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).TextColor =
System.Drawing.Color.White;
                }
                else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Marrom")
                {
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).ColorStyle =
EeekSoft.Web.PopupColorStyle.Custom;
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).Shadow =
System.Drawing.Color.Brown;
                    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).GradientDark =
System.Drawing.Color.Brown;
```

```
(this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).TextColor =
System.Drawing.Color.Black;
    }
    else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Cinza")
    {
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).ColorStyle =
EeekSoft.Web.PopupColorStyle.Custom;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).Shadow =
System.Drawing.Color.Gray;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).GradientDark =
System.Drawing.Color.Gray;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).TextColor =
System.Drawing.Color.Black;
    }
    else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Indigo")
    {
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).ColorStyle =
EeekSoft.Web.PopupColorStyle.Custom;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).Shadow =
System.Drawing.Color.Indigo;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).GradientDark =
System.Drawing.Color.Indigo;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).TextColor =
System.Drawing.Color.Black;
    }
    else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Magenta")
    {
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).ColorStyle =
EeekSoft.Web.PopupColorStyle.Custom;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).Shadow =
System.Drawing.Color.Magenta;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).GradientDark =
System.Drawing.Color.Magenta;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).TextColor =
System.Drawing.Color.Black;
    }
    else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Laranja")
    {
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).ColorStyle =
EeekSoft.Web.PopupColorStyle.Custom;
```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
(this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).Shadow =
System.Drawing.Color.Orange;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).GradientDark =
System.Drawing.Color.Orange;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).TextColor =
System.Drawing.Color.Black;
    }
    else if
(dtPopup.Rows[nPopup]["VC_POP_COR"].ToString() == "Prata")
    {
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).ColorStyle =
EeekSoft.Web.PopupColorStyle.Custom;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).Shadow =
System.Drawing.Color.Silver;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).GradientDark =
System.Drawing.Color.Silver;
        (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).TextColor =
System.Drawing.Color.Black;
    }
    (this.Controls[0].FindControl("popupwin" +
nPopup.ToString()) as EeekSoft.Web.PopupWin).Visible = true;
}
}
catch
{
}
}
```

### 5.10 Dica: Enviar e-mails num determinado horário.

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            StringBuilder sb = new StringBuilder();
            sb.Append("SELECT ");
            sb.Append("ID_ELT AS ID, ");
            sb.Append("VC_ELT_DE AS DE, ");
            sb.Append("VC_ELT_PARA AS PARA, ");
            sb.Append("VC_ELT_CC AS CC, ");
            sb.Append("VC_ELT_TITULO AS TITULO, ");
            sb.Append("VC_ELT_TEXTO AS TEXTO, ");
            sb.Append("VC_ELT_ANEXO AS ANEXO, ");
            sb.Append("VC_ELT_IMAGEM AS IMAGEM, ");
            sb.Append("DH_ELT_GRAVADO AS GRAVADO, ");
            sb.Append("NU_ELT_ENVIADO AS ENVIADO, ");
            sb.Append("DH_ELT_PROGRAMACAO AS DAT P, ");
```

```

sb.Append("SUBSTRING(CONVERT(VARCHAR(11),DH_ELT_PROGRAMACAO,114),1,2) AS
HOR_P, ");

sb.Append("SUBSTRING(CONVERT(VARCHAR(11),DH_ELT_PROGRAMACAO,114),4,2) AS
MIN_P, ");

        sb.Append("GETDATE() AS DAT_C, ");
        sb.Append("SUBSTRING(CONVERT(VARCHAR(11),GETDATE(),114),1,2)
AS HOR_C, ");
        sb.Append("SUBSTRING(CONVERT(VARCHAR(11),GETDATE(),114),4,2)
AS MIN_C ");
        sb.Append("FROM EMAILSLOTE ");
        sb.Append("WHERE DH_ELT_PROGRAMACAO<=GETDATE() ");
        sb.Append("AND NU_ELT_ENVIADO=0 ");
        sb.Append("AND
SUBSTRING(CONVERT(VARCHAR(11),DH_ELT_PROGRAMACAO,114),1,2)>=SUBSTRING(CONV
ERT(VARCHAR(11),GETDATE(),114),1,2) ");
        sb.Append("AND
SUBSTRING(CONVERT(VARCHAR(11),GETDATE(),114),4,2) BETWEEN
SUBSTRING(CONVERT(VARCHAR(11),DH_ELT_PROGRAMACAO,114),4,2) AND '60'");
        DataTable dtAux = RepositorioBase.DadosConsultar("1",
sb.ToString());
        if (dtAux.Rows.Count > 0)
        {
            for (int i = 0; i <= dtAux.Rows.Count - 1; i++)
            {
                try
                {
                    Geral.Email("S", dtAux.Rows[i]["DE"].ToString(),
dtAux.Rows[i]["PARA"].ToString(), dtAux.Rows[i]["CC"].ToString(),
dtAux.Rows[i]["TITULO"].ToString(), dtAux.Rows[i]["TEXTO"].ToString(),
"html", "alta", dtAux.Rows[i]["ANEXO"].ToString(),
dtAux.Rows[i]["IMAGEM"].ToString(), true);
                    RepositorioBase.DadosManter("1", "UPDATE
PEMAILSLOTE SET NU_ELT_ENVIADO=1 WHERE ID_ELT=" +
dtAux.Rows[i]["ID"].ToString());
                    break;
                }
                catch
                {
                }
            }
        }
    }
}
}

```

### 5.11 Dica: Gerar um registro de acesso ao sistema armazenando numa tabela de LOGs.

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            try
            {
                RepositorioBase.DadosManter("1", "INSERT INTO LOG
(VC_LOG_USUARIO,VC_LOG_DATAHORA,VC_LOG_MODULO,VC_LOG_OPERACAO) VALUES ('"

```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
+ Session["Usuario"].ToString() + "','" + DateTime.Now.ToString() + "','"
+ Session["Modulo"].ToString() + "','" + Session["Operacao"].ToString() +
"')");
    }
    catch
    {
    }
}
}
```

### 5.12 Dica: Exibir data com mês por extenso.

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            LabelData.Text =
            Geral.Semana(DateTime.Now.DayOfWeek.ToString()) + ", " +
            DateTime.Now.Day.ToString() + " de " + Geral.Mes(DateTime.Now.Month) + "
            de " + DateTime.Now.Year.ToString();
        }
    }
}
```

### 5.13 Dica: Obter configurações do banco de dados do arquivo web.config.

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Session["BancoIP"] =
            ConfigurationManager.AppSettings["BancoIP"];
            Session["BancoTipo"] =
            ConfigurationManager.AppSettings["BancoTipo"];
            Session["BancoConexao"] =
            ConfigurationManager.AppSettings["BancoConexao"];
        }
    }
}
```

### 5.14 Dica: Executar um JavaScript para setar o focus num objeto ASPX no código CSharp.

```
public partial class _Default : System.Web.UI.Page
{
    private void SetFocus(Control ctrl)
    {
        string focusScript = "<script language='javascript'>" +
        "document.getElementById('" + ctrl.ClientID + "').focus();</script>";
        Page.RegisterStartupScript("FocusScript", focusScript);
    }
}
```

### 5.15 Dica: Baixar (download) de arquivo no cliente armazenado no servidor.

```
public partial class _Default : System.Web.UI.Page
{
```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
public void Download(string pArquivo)
{
    string sArquivo = Request.ServerVariables["APPL_PHYSICAL_PATH"] +
@"App_Data\" + pArquivo;
    System.IO.FileInfo arquivo = new System.IO.FileInfo(sArquivo);
    Response.Clear();
    Response.AddHeader("Content-Disposition", "attachment; filename="
+ arquivo.Name);
    Response.AddHeader("Content-Length", arquivo.Length.ToString());
    Response.ContentType = "application/octet-stream";
    Response.WriteFile(arquivo.FullName);
    Response.End();
    Response.Redirect("Default.aspx");
}
}
```

### 5.16 Dica: Carregar configurações de estilo de uma arquivo CSS via CSharp.

```
public partial class _Default : System.Web.UI.Page
{
    public void ArquivoCSS(string pCor)
    {
        System.Web.UI.HtmlControls.HtmlHead header =
(System.Web.UI.HtmlControls.HtmlHead) Page.Header;
        System.Web.UI.HtmlControls.HtmlLink link = new
System.Web.UI.HtmlControls.HtmlLink();
        link.Attributes.Add("href", Page.ResolveClientUrl("Arquivo.css"));
        link.Attributes.Add("type", "text/css");
        link.Attributes.Add("rel", "stylesheet");
        header.Controls.Add(link);
    }
}
```

### 5.17 Dica: Realizar compartilhamento de informações nas redes sociais.

```
public partial class _Default : System.Web.UI.Page
{
    protected void ImageButtonFacebook_Click(object sender,
ImageClickEventArgs e)
    {
        try
        {
            if (LabelImprensaLink.Text.Trim() != "")
            {
                String sLink = LabelImprensaLink.Text.Trim().Replace(" ",
"_");
                String sCompartilhamento =
"https://www.facebook.com/login.php?next=http://www.facebook.com/sharer.ph
p?u=" + sLink + "&ret=login";
                sCompartilhamento =
System.Text.RegularExpressions.Regex.Replace(sCompartilhamento,
@"<(.|\n)*?>", String.Empty);
                sCompartilhamento = Server.HtmlDecode(sCompartilhamento);
                Response.Redirect(sCompartilhamento);
            }
            else
            {
                Geral.Alerta(this, "Não existe link para
compartilhamento", "Alerta");
            }
        }
    }
}
```

```
    }
    catch
    {
        Geral.Alerta(this, "Compartilhamento negado - Acesso ao
facebook bloqueado", "Alerta");
    }
}

protected void ImageButtonTwitter_Click(object sender,
ImageClickEventArgs e)
{
    try
    {
        if (LabelImprensaLink.Text.Trim() != "")
        {
            String sLink = LabelImprensaLink.Text.Trim().Replace(" ",
"_");
            String sCompartilhamento =
"https://twitter.com/intent/tweet?url=" + sLink +
"&title=Título&text=Clique no link para ler a notícia.";
            sCompartilhamento =
System.Text.RegularExpressions.Regex.Replace(sCompartilhamento,
@"<(.|\n)*?>", String.Empty);
            sCompartilhamento = Server.HtmlDecode(sCompartilhamento);
            Response.Redirect(sCompartilhamento);
        }
        else
        {
            Geral.Alerta(this, "Não existe link para
compartilhamento", "Alerta");
        }
    }
    catch
    {
        Geral.Alerta(this, "Compartilhamento negado - Acesso ao
twitter bloqueado", "Alerta");
    }
}

protected void ImageButtonGoogleFavoritos_Click(object sender,
ImageClickEventArgs e)
{
    try
    {
        if (LabelImprensaLink.Text.Trim() != "")
        {
            String sLink = LabelImprensaLink.Text.Trim().Replace(" ",
"_");
            String sCompartilhamento =
"https://www.google.com/bookmarks/mark?op=add&bkmk=" + sLink +
"&title=Título&annotation=Clique no link para ler a notícia.";
            sCompartilhamento =
System.Text.RegularExpressions.Regex.Replace(sCompartilhamento,
@"<(.|\n)*?>", String.Empty);
            sCompartilhamento = Server.HtmlDecode(sCompartilhamento);
            Response.Redirect(sCompartilhamento);
        }
        else
        {
            Geral.Alerta(this, "Não existe link para
compartilhamento", "Alerta");
        }
    }
}
```

```
    }
}
catch
{
    Geral.Alerta(this, "Compartilhamento negado - Acesso ao
google+ bloqueado", "Alerta");
}
}
protected void ImageButtonGoogleplus_Click(object sender,
ImageClickEventArgs e)
{
    try
    {
        try
        {
            String sLink = LabelImprensaLink.Text.Trim().Replace(" ",
"_");
            String sCompartilhamento =
"https://plus.google.com/share?url=" + sLink + "&gpsrc=gplp0";
            sCompartilhamento =
System.Text.RegularExpressions.Regex.Replace(sCompartilhamento,
@"<(.|\n)*?>", String.Empty);
            sCompartilhamento = Server.HtmlDecode(sCompartilhamento);
            Response.Redirect(sCompartilhamento);
        }
        catch
        {
            Geral.Alerta(this, "Compartilhamento negado - Acesso ao
google+ bloqueado", "Alerta");
        }
    }
    catch
    {
    }
}
protected void ImageButtonLinkedin_Click(object sender,
ImageClickEventArgs e)
{
    try
    {
        if (LabelImprensaLink.Text.Trim() != "")
        {
            String sLink = LabelImprensaLink.Text.Trim().Replace(" ",
"_");
            String sCompartilhamento =
"https://www.linkedin.com/shareArticle?mini=true&url=" + sLink +
"&title=Título :: Clique no link para ler a noticia.";
            sCompartilhamento =
System.Text.RegularExpressions.Regex.Replace(sCompartilhamento,
@"<(.|\n)*?>", String.Empty);
            sCompartilhamento = Server.HtmlDecode(sCompartilhamento);
            Response.Redirect(sCompartilhamento);
        }
        else
        {
            Geral.Alerta(this, "Não existe link para
compartilhamento", "Alerta");
        }
    }
}
```

```
    }
    catch
    {
        Geral.Alerta(this, "Compartilhamento negado - Acesso ao
linkedin bloqueado", "Alerta");
    }
}

protected void ImageButtonGmail_Click(object sender,
ImageClickEventArgs e)
{
    try
    {
        if (LabelImprensaLink.Text.Trim() != "")
        {
            String sLink = LabelImprensaLink.Text.Trim();
            String sCompartilhamento =
"https://mail.google.com/mail/u/0/?view=cm&fs=1&to&su=Título&body=Clique
no link para ler a notícia :: " + sLink; // +"&ui=2&tf=1";
            sCompartilhamento =
System.Text.RegularExpressions.Regex.Replace(sCompartilhamento,
@"<(.|\n)*?>", String.Empty);
            sCompartilhamento = Server.HtmlDecode(sCompartilhamento);
            Response.Redirect(sCompartilhamento);
        }
        else
        {
            Geral.Alerta(this, "Não existe link para
compartilhamento", "Alerta");
        }
    }
    catch
    {
        Geral.Alerta(this, "Compartilhamento negado - Acesso ao gmail
bloqueado", "Alerta");
    }
}

#endregion
}
```

### 5.18 Dica: Abrir uma janela popup em JavaScript via código CSharp.

```
public partial class _Default : System.Web.UI.Page
{
    protected void LinkButtonInformativo_Click(object sender, EventArgs e)
    {
        try
        {
            ScriptManager.RegisterStartupScript(this, typeof(Page),
"Redirect", "window.open('PaginaBranca.aspx', '_blank',
'menubar=0,scrollbars=1,width=780,height=700,top=10');", true);
        }
        catch
        {
        }
    }
}
```

### 5.19 Dica: Varrer objeto de uma página ASPX no código CSharp.

```
public partial class _Default : System.Web.UI.Page
```

```
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        for (int i = 0; i <= this.Controls.Count - 1; i++)
        {
            try
            {
                sI = (i + 1).ToString();
                if ((this.Controls[0].FindControl("TextBox" + sI) as
                TextBox).Text.Trim() == "")
                {
                    (this.Controls[0].FindControl("TextBox" + sI) as
                    TextBox).Focus();
                    Geral.Alerta(this, "Campo de preenchimento
                    obrigatório", "Alerta");
                }
            }
            catch
            {
            }
        }
    }
}
```

### 5.20 Dica: Converter arquivo binário para string.

```
public partial class _Default : System.Web.UI.Page
{
    public static String ConverteBinarioParaString(String pArquivo)
    {
        try
        {
            FileStream fs = new FileStream(pArquivo, FileMode.Open,
            FileAccess.Read);
            byte[] filebytes = new byte[fs.Length];
            fs.Read(filebytes, 0, Convert.ToInt32(fs.Length));
            return "data:image/png;base64," +
            Convert.ToBase64String(filebytes, Base64FormattingOptions.None);
        }
        catch
        {
        }
    }
}
```

### 5.21 Dica: Enviar email pelo GMail.

```
public partial class _Default : System.Web.UI.Page
{
    public static void GEMail(string pPara, string pTitulo, string pCorpo,
    string pEmail, string pSenha)
    {
        try
        {
            System.Net.Mail.SmtpClient cliente = new
            System.Net.Mail.SmtpClient("smtp.gmail.com", 587);
            cliente.EnableSsl = true;
            System.Net.Mail.MailAddress remetente = new
            System.Net.Mail.MailAddress(pEmail, pTitulo);
```

```
        System.Net.Mail.MailAddress destinatario = new
System.Net.Mail.MailAddress(pPara, pPara);
        System.Net.Mail.MailMessage mensagem = new
System.Net.Mail.MailMessage(remetente, destinatario);
        mensagem.Body = pCorpo;
        mensagem.Subject = pTitulo;
        System.Net.NetworkCredential credenciais = new
System.Net.NetworkCredential(pEmail, pSenha, "");
        cliente.Credentials = credenciais;
        cliente.Send(mensagem);
    }
    catch (Exception ex)
    {
        throw (ex);
    }
}
}
```

### 5.22 Dica: Obter o string do SELECT de um GridView.

```
public partial class _Default : System.Web.UI.Page
{
    public static string SelectGrade(GridView pGridView, string pComando,
ArrayList al)
    {
        bool bPassou = false;
        string sSelect = "SELECT ";
        for (int col = 1; col <= pGridView.Columns.Count - 1; col++)
        {
            bPassou = false;
            if (pGridView.Columns[col].Visible)
            {
                if (pGridView.Columns[col].SortExpression != "")
                {
                    bPassou = true;
                    sSelect = sSelect +
pGridView.Columns[col].SortExpression;
                }
                if (pGridView.Columns[col].HeaderText != "")
                {
                    bPassou = true;
                    sSelect = sSelect + " AS '" +
pGridView.Columns[col].HeaderText;
                }
                if (bPassou == true)
                {
                    sSelect = sSelect + "', ";
                }
            }
        }
        sSelect = sSelect.Substring(0, sSelect.Length - 2);
        string sFrom = pComando.Substring(pComando.IndexOf("FROM"));
        if (sFrom.IndexOf("WHERE") != -1)
        {
            sFrom = sFrom.Substring(0, sFrom.IndexOf("WHERE"));
        }
        else if (pComando.IndexOf("ORDER BY") != -1)
        {
            sFrom = sFrom.Substring(0, sFrom.IndexOf("ORDER BY"));
        }
    }
}
```

```

    }
    sSelect = sSelect + " " + sFrom;
    if (al[0] != "")
    {
        int ind = 0;
        string sParam = "";
        string sWhereOrder =
pComando.Substring(pComando.IndexOf("WHERE"));
        while (sWhereOrder.IndexOf("@") != -1)
        {
            sParam = sWhereOrder.Substring(sWhereOrder.IndexOf("@"));
            if (sParam.IndexOf(" ") != -1)
            {
                sParam = sParam.Substring(0, sParam.IndexOf(" "));
            }
            if (sParam.IndexOf("(") != -1)
            {
                sParam = sParam.Substring(0, sParam.IndexOf("("));
            }
            sWhereOrder = sWhereOrder.Replace(sParam, "'" + al[ind++]
+ "'");
        }
        sSelect = sSelect + " " + sWhereOrder;
    }
    return (sSelect);
}
}
}

```

### 5.23 Dica: Obter o string do SELECT de um DetailView.

```

public partial class _Default : System.Web.UI.Page
{
    public static string SelectDetalhe(DetailsView pDetailsView,
SqlDataSource pSqlDataSourceDetails, ArrayList al, string pSQLBase, string
pSQLCriterio, string pSQLOrdenacao)
    {
        bool bPassou = false;
        string sFrom =
pSqlDataSourceDetails.SelectCommand.Substring(pSqlDataSourceDetails.Select
Command.IndexOf(" FROM ") + 6);
        sFrom = sFrom.Substring(0, sFrom.IndexOf(" "));
        string sSelect = "SELECT ";
        for (int lin = 0; lin <= pDetailsView.Fields.Count - 1; lin++)
        {
            bPassou = false;
            if (pDetailsView.Fields[lin].Visible)
            {
                if (pDetailsView.Fields[lin].SortExpression.Trim() != "")
                {
                    bPassou = true;
                    try
                    {
                        if
(pDetailsView.Fields[lin].SortExpression.Trim().Substring(0, 3) == "ID_")
                        {
                            sSelect = sSelect + " " + sFrom + "." +
pDetailsView.Fields[lin].SortExpression;
                        }
                        else

```

```

        {
            sSelect = sSelect +
pDetailsView.Fields[lin].SortExpression;
        }
        catch
        {
            sSelect = sSelect +
pDetailsView.Fields[lin].SortExpression;
        }
        if (pDetailsView.Fields[lin].HeaderText.Trim() != "")
        {
            sSelect = sSelect + " AS '" +
pDetailsView.Fields[lin].HeaderText + "'";
        }
    }
    if (bPassou == true)
    {
        sSelect = sSelect + ", ";
    }
}

sSelect = sSelect.Substring(0, sSelect.Length - 2);
sSelect = sSelect + " " +
pSqlDataSourceDetails.SelectCommand.Substring(pSqlDataSourceDetails.SelectCommand.IndexOf("FROM"));
int ind = 0;
while (sSelect.IndexOf("@") != -1)
{
    string sParam = sSelect.Substring(sSelect.IndexOf("@"));
    if (sParam.IndexOf(" ") != -1)
    {
        sParam = sParam.Substring(0, sParam.IndexOf(" "));
    }
    if (sParam.IndexOf("(") != -1)
    {
        sParam = sParam.Substring(0, sParam.IndexOf("("));
    }
    sSelect = sSelect.Replace(sParam, "'" + al[ind++] + "'");
}
return (sSelect);
}
}

```

#### 5.24 Dica: Retornar o dia da semana de uma data.

```

public partial class _Default : System.Web.UI.Page
{
    public static string Semana(string pData, string pRetorno)
    {
        string sSemana = Convert.ToDateTime(pData).DayOfWeek.ToString();
        if (sSemana == "Sunday")
        {
            if (pRetorno == "")
            {
                return ("Domingo");
            }
            else
            {

```

```
        return ("Dom");
    }
}
else if (sSemana == "Monday")
{
    if (pRetorno == "")
    {
        return ("Segunda");
    }
    else
    {
        return ("Seg");
    }
}
else if (sSemana == "Tuesday")
{
    if (pRetorno == "")
    {
        return ("Terça");
    }
    else
    {
        return ("Ter");
    }
}
else if (sSemana == "Wednesday")
{
    if (pRetorno == "")
    {
        return ("Quarta");
    }
    else
    {
        return ("Qua");
    }
}
else if (sSemana == "Thursday")
{
    if (pRetorno == "")
    {
        return ("Quinta");
    }
    else
    {
        return ("Qui");
    }
}
else if (sSemana == "Friday")
{
    if (pRetorno == "")
    {
        return ("Sexta");
    }
    else
    {
        return ("Sex");
    }
}
else
```

```
{
    if (pRetorno == "")
    {
        return ("Sábado");
    }
    else
    {
        return ("Sab");
    }
}
}
```

### 5.25 Dica: Retornar o mês por extenso.

public partial class \_Default : System.Web.UI.Page

```
{
    public static string Mes(int pMes)
    {
        if (pMes == 1)
        {
            return ("Janeiro");
        }
        else if (pMes == 2)
        {
            return ("Fevereiro");
        }
        else if (pMes == 3)
        {
            return ("Março");
        }
        else if (pMes == 4)
        {
            return ("Abril");
        }
        else if (pMes == 5)
        {
            return ("Maio");
        }
        else if (pMes == 6)
        {
            return ("Junho");
        }
        else if (pMes == 7)
        {
            return ("Julho");
        }
        else if (pMes == 8)
        {
            return ("Agosto");
        }
        else if (pMes == 9)
        {
            return ("Setembro");
        }
        else if (pMes == 10)
        {
            return ("Outubro");
        }
    }
}
```

```
else if (pMes == 11)
{
    return ("Novembro");
}
else
{
    return ("Dezembro");
}
}
```

**5.26 Dica: Exibir uma saudação baseado no horário.**

```
public partial class _Default : System.Web.UI.Page
{
    public static string Saudacao()
    {
        if (DateTime.Now.Hour < 12)
        {
            return ("Bom dia,");
        }
        else if (DateTime.Now.Hour < 18)
        {
            return ("Boa tarde,");
        }
        else
        {
            return ("Boa noite,");
        }
    }
}
```

**5.27 Dica: Transformar a primeiro letra em maiúscula e as demais em minúsculas.**

```
public partial class _Default : System.Web.UI.Page
{
    public static string Capitalizar(string pTexto)
    {
        return (pTexto.Substring(0, 1).ToUpper() +
pTexto.Substring(1).ToLower());
    }
}
```

**5.28 Dica: Exportar dados de uma gridview para formato EXCEL ou HTML ou JPG.**

```
public partial class _Default : System.Web.UI.Page
{
    public static void Exportar(String pArquivo, GridView gv, String
pTipo)
    {
        HttpContext.Current.Response.Clear();
        if (pTipo.ToLower() == "xls")
        {
            HttpContext.Current.Response.AddHeader("content-disposition",
string.Format("attachment; filename={0}", pArquivo + "." + pTipo));
            HttpContext.Current.Response.ContentType = "application/ms-
excel";
        }
        else if (pTipo.ToLower() == "html")
        {

```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
HttpContext.Current.Response.AddHeader("content-disposition",
string.Format("attachment; filename={0}", pArquivo + "." + pTipo));
HttpContext.Current.Response.ContentType = "text/HTML";
}
else
{
    HttpContext.Current.Response.AddHeader("content-disposition",
string.Format("attachment; filename={0}", pArquivo + ".jpg"));
    HttpContext.Current.Response.ContentType = "image/JPEG";
}
using (StringWriter sw = new StringWriter())
{
    using (HtmlTextWriter htw = new HtmlTextWriter(sw))
    {
        // Criar tabela para conter as linhas da gridview
        Table table = new Table();
        // Incluir configurações
        table.GridLines = gv.GridLines;
        // Adicionar o cabeçalho da linha na tabela
        if (gv.HeaderRow != null)
        {
            PreparaControleParaExportar(gv.HeaderRow);
            table.Rows.Add(gv.HeaderRow);
        }
        // Adiciona cada linha de dados na tabela
        foreach (GridViewRow row in gv.Rows)
        {
            PreparaControleParaExportar(row);
            table.Rows.Add(row);
        }
        // Adiciona o rodapé na tabela
        if (gv.FooterRow != null)
        {
            PreparaControleParaExportar(gv.FooterRow);
            table.Rows.Add(gv.FooterRow);
        }
        // Renderizar a tabela para dentro do htmlwriter
        table.RenderControl(htw);
        // Renderizar o htmlwriter para dentro do response
        HttpContext.Current.Response.Write(sw.ToString());
        HttpContext.Current.Response.End();
    }
}
}
```

### 5.29 Dica: Preparar os diversos tipos de controles para serem exportados.

```
public partial class _Default : System.Web.UI.Page
{
    private static void PreparaControleParaExportar(Control control)
    {
        for (int i = 0; i < control.Controls.Count; i++)
        {
            Control current = control.Controls[i];
            if (current is LinkButton)
            {
                control.Controls.Remove(current);
            }
        }
    }
}
```

```
                control.Controls.AddAt(i, new LiteralControl((current as
LinkButton).Text));
            }
            else if (current is ImageButton)
            {
                control.Controls.Remove(current);
                control.Controls.AddAt(i, new LiteralControl((current as
ImageButton).AlternateText));
            }
            else if (current is HyperLink)
            {
                control.Controls.Remove(current);
                control.Controls.AddAt(i, new LiteralControl((current as
HyperLink).Text));
            }
            else if (current is DropDownList)
            {
                control.Controls.Remove(current);
                control.Controls.AddAt(i, new LiteralControl((current as
DropDownList).SelectedItem.Text));
            }
            else if (current is CheckBox)
            {
                control.Controls.Remove(current);
                control.Controls.AddAt(i, new LiteralControl((current as
CheckBox).Checked ? "True" : "False"));
            }
            if (current.HasControls())
            {
                PreparaControleParaExportar(current);
            }
        }
    }
}
```

### 5.30 Dica: Enviar email formatado em HTML.

```
public partial class _Default : System.Web.UI.Page
{
    public static bool EMail(string pUsuarioEmailReceber, string pFrom,
string pTo, string pCC, string pSubject, string pBody, string pBodyFormat,
string pPriority, string pAnexo, string pImagem, bool pCabecalho)
    {
        String sExtensao = "";
        try
        {
            bool bEnvia = true;
            try
            {
                bEnvia = RepositorioUsuarios.getReceberEmail(pTo);
            }
            catch
            {
            }
            if (bEnvia)
            {
                // Tratamento de anexo e imagem
                if (pAnexo != "")
                {

```

```

        sExtensao =
pAnexo.Substring(pAnexo.IndexOf(".")).ToLower().Trim();
        if ((sExtensao == ".jpg") || (sExtensao == ".png") ||
(sExtensao == ".bmp") || (sExtensao == ".gif"))
        {
            pImagem = pAnexo;
            pAnexo = "";
        }
    }
    else if (pImagem != "")
    {
        sExtensao =
pImagem.Substring(pImagem.IndexOf(".")).ToLower().Trim();
        if ((sExtensao == ".jpg") || (sExtensao == ".png") ||
(sExtensao == ".bmp") || (sExtensao == ".gif"))
        {
        }
        else
        {
            pAnexo = pImagem;
            pImagem = "";
        }
    }
    else
    {
        pAnexo = "";
        pImagem = "";
    }
    MailMessage mailMsg = new MailMessage();
    mailMsg.To.Add(pTo);
    if (pCC != "")
    {
        mailMsg.CC.Add(pCC);
    }
    MailAddress mailAddress = new MailAddress(pFrom);
    mailMsg.From = mailAddress;
    mailMsg.Subject = pSubject;
    StringBuilder sFile = new StringBuilder();
    if (pBodyFormat == "html")
    {
        String sEMail = "(e-mail)";
        String sEmailLogo =
System.Web.HttpContext.Current.Server.MapPath("") +
"\\App_Themes\\Layout\\img-logo.png";
        String sEMailSistema = "Sistema";
        String sEMailData = "Data: " +
DateTime.Now.Day.ToString("00") + "/" + DateTime.Now.Month.ToString("00")
+ "/" + DateTime.Now.Year.ToString("0000");
        String sEMailHora = "Hora: " +
DateTime.Now.Hour.ToString("00") + ":" +
DateTime.Now.Minute.ToString("00");
        String sEMailPagina = "Página: " + "1/1";
        String sEMailTitulo = pSubject;
        String sEMailSubtitulo = "";
        String sEMailCorpo = pBody.Replace("\n", "<BR>");
        mailMsg.IsBodyHtml = true;
        pBody = pBody.Replace("\n", "<BR>");
    }

```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```

        sFile.Append("<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML
1.0 Transitional//EN' 'http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd'>");
        sFile.Append("<html
xmlns='http://www.w3.org/1999/xhtml'>");
        sFile.Append("<head><title>");
        sFile.Append(sEmail);
        sFile.Append("</title></head>");
        sFile.Append("<body>");
        sFile.Append("        <form name='formEMail' method='post'
action='EMail.aspx' id='formEMail'>");
        sFile.Append("<div>");
        sFile.Append("<input type='hidden' name='__VIEWSTATE'
id='__VIEWSTATE'
value='/wEPDwUKLTy0NjY1OTE5M2RkeDYgVtB6e4OxFOsR354HIssDIXk=' />");
        sFile.Append("</div>");
        sFile.Append("        <table id='tableEMail'
style='vertical-align: top; text-align: center; width: 100%; font-family:
Tahoma; font-size: 12px; color: #006793; font-weight: bold;'>");
        sFile.Append("                <tr style='line-height: 2px;
background-color: #006793;'>");
        sFile.Append("                        <td colspan='3'>");
        sFile.Append("                                </td>");
        sFile.Append("                                </tr>");
        sFile.Append("                                <tr>");
        sFile.Append("                                        <td style='width: 33%'>");
        sFile.Append("                                                <img src='\" +
MakeImageSrcData(System.Web.HttpContext.Current.Server.MapPath(\"App_Themes
/Layout/img-logo.png\")) + \"' />");
        sFile.Append("                                </td>");
        sFile.Append("                                </tr>");
        sFile.Append("                                <tr>");
        sFile.Append("                                        <td style='width: 33%'>");
        sFile.Append("                                                <br />");
        sFile.Append("                                                <span
id='LabelProcuradoria'>PROCURADORIA GERAL DO ESTADO DE
PERNAMBUCO</span>");
        sFile.Append("                                <br />");
        sFile.Append("                                </td>");
        sFile.Append("                                </tr>");
        sFile.Append("                                <tr>");
        sFile.Append("                                        <td style='width: 33%'>");
        sFile.Append("                                                <span id='LabelData'>\" +
sEmailData + \"</span>");
        sFile.Append("                                </td>");
        sFile.Append("                                </tr>");
        sFile.Append("                                <tr>");
        sFile.Append("                                        <td style='width: 33%'>");
        sFile.Append("                                                <span id='LabelHora'>\" +
sEmailHora + \"</span>");
        sFile.Append("                                </td>");
        sFile.Append("                                <tr>");
        sFile.Append("                                <tr>");
        sFile.Append("                                <td colspan='3'>");
        sFile.Append("                                        <span id='LabelTitulo'>\"
+ sEmailTitulo + \"</span>");
        sFile.Append("                                </td>");
        sFile.Append("                                </tr>");
        sFile.Append("                                <tr>");

```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```

        sFile.Append("                <td colspan='3'>");
        sFile.Append("                <span
id='LabelSubtitulo'>" + sEmailSubtitulo + "</span>");
        sFile.Append("                </td>");
        sFile.Append("            </tr>");
        sFile.Append("            <tr>");
        sFile.Append("                <td colspan='3'
style='vertical-align: top; text-align: left; width: 100%; font-family:
Tahoma; font-size: 12px; font-weight: normal;'>");
        sFile.Append("                <br />");
        sFile.Append("                <br />");
        sFile.Append("                <span>" + sEmailCorpo +
"</span>");
        sFile.Append("                <br />");
        sFile.Append("            </td>");
        sFile.Append("        </tr>");
        sFile.Append("    <tr style='line-height: 2px;
background-color: #006793;'>");
        sFile.Append("        <td colspan='3'>");
        sFile.Append("        </td>");
        sFile.Append("    </tr>");
        sFile.Append("    </table>");
        sFile.Append("    </form>");
        sFile.Append("</body>");
        sFile.Append("</html>");
        if (pImagem != "")
        {
            pBody = pBody.Replace("<html>", "");
            pBody = pBody.Replace("<body>", "");
            pBody = pBody.Replace("</html>", "");
            pBody = pBody.Replace("</body>", "");
            pBody = "<html><body>" + pBody + "<BR><img
src='cid:site' /></body></html>";
            AlternateView av =
AlternateView.CreateAlternateViewFromString(pBody, null,
System.Net.Mime.MediaTypeNames.Text.Html);
            if (pImagem.IndexOf("\\") != -1)
            {
                LinkedResource lr = new
LinkedResource(pImagem);
                lr.ContentId = "site";
                av.LinkedResources.Add(lr);
            }
            else
            {
                LinkedResource lr = new
LinkedResource(System.Web.HttpContext.Current.Server.MapPath(pImagem));
                lr.ContentId = "site";
                av.LinkedResources.Add(lr);
            }
            mailMsg.AlternateViews.Add(av);
        }
    }
    else
    {
        mailMsg.IsBodyHtml = false;
        if (pCabecalho)
        {
            sFile.Append("Título\n");

```

```

        sFile.Append("Subtitulo\n");
        sFile.Append("Site\n");
        sFile.Append("Intranet\n");
    }
}
mailMsg.Body = sFile.ToString();
if (pAnexo != "")
{
    if (pAnexo != null)
    {
        try
        {
            string attachmentPath = pAnexo;
            System.Net.Mail.Attachment inline = null;
            if (sExtensao == ".pdf")
            {
                inline = new
System.Net.Mail.Attachment(attachmentPath,
MediaTypeNames.Application.Pdf);
            }
            else if (sExtensao == ".rtf")
            {
                inline = new
System.Net.Mail.Attachment(attachmentPath,
MediaTypeNames.Application.Rtf);
            }
            else if ((sExtensao == ".zip") || (sExtensao
== ".rar"))
            {
                inline = new
System.Net.Mail.Attachment(attachmentPath,
MediaTypeNames.Application.Zip);
            }
            else
            {
                inline = new
System.Net.Mail.Attachment(attachmentPath,
MediaTypeNames.Application.Octet);
            }
            inline.ContentDisposition.Inline = true;
            inline.ContentDisposition.DispositionType =
DispositionTypes.Inline;
            if (sExtensao == ".pdf")
            {
                inline.ContentType.MediaType =
MediaTypeNames.Application.Pdf;
            }
            else if (sExtensao == ".rtf")
            {
                inline.ContentType.MediaType =
MediaTypeNames.Application.Rtf;
            }
            else if ((sExtensao == ".zip") || (sExtensao
== ".rar"))
            {
                inline.ContentType.MediaType =
MediaTypeNames.Application.Zip;
            }
            else

```

```

        {
            inline.ContentType.MediaType =
MediaTypesNames.Application.Octet;
        }
        inline.ContentType.Name =
Path.GetFileName(attachmentPath);
        mailMsg.Attachments.Add(inline);
    }
    catch
    {
        mailMsg.Body = mailMsg.Body + "<Br><Br>Falha
na incorporação do arquivo anexado!";
    }
}
}
if (pPriority == "alta")
{
    mailMsg.Priority = MailPriority.High;
}
else if (pPriority == "baixa")
{
    mailMsg.Priority = MailPriority.Low;
}
else if (pPriority == "normal")
{
    mailMsg.Priority = MailPriority.Normal;
}
try
{
    SmtpClient smtpClient = new SmtpClient("endereço",
porta);

    smtpClient.EnableSsl = false;
    smtpClient.Send(mailMsg);
}
catch
{
    try
    {
        SmtpClient smtpClient = new
SmtpClient("smtp.gmail.com", 587); //465 ou 587
        smtpClient.EnableSsl = true;
        smtpClient.UseDefaultCredentials = true;
        smtpClient.Send(mailMsg);
    }
    catch (Exception x)
    {
        throw (x);
    }
}
}
return (true);
}
catch (Exception x)
{
    throw (x);
}
}
}

```

**5.31 Dica: Ativar recurso de som.**

```
public partial class _Default : System.Web.UI.Page
{
    public static void Som(int pTipo)
    {
        Thread.Sleep(1);
        if (pTipo == 1)
        {
            Console.Beep(800, 3);
        }
        else
        {
            Console.Beep(1000, 3);
        }
    }
}
```

**5.32 Dica: Alerta sonoro em javascript.**

```
public partial class _Default : System.Web.UI.Page
{
    public static void AlertaSonoro(System.Web.UI.Page senderPage, string
alertSonoro, string alertKey)
    {
        string strScript = "<script> language=JavaScript>function
TocarSom(alertSonoro) {var thissound=
eval('document.'+soundobj);thissound.Play();}</script>";
        if (!(senderPage.IsStartupScriptRegistered(alertKey)))
        {
            senderPage.RegisterStartupScript(alertKey, strScript);
        }
    }
}
```

**5.33 Dica: Exibir janela de alerta (popup) em javascript.**

```
public partial class _Default : System.Web.UI.Page
{
    public static void Alerta(System.Web.UI.Page senderPage, string
alertMsg, string alertKey)
    {
        string strScript = "<script language=JavaScript>alert('" +
alertMsg + "')</script>";
        if (!(senderPage.IsStartupScriptRegistered(alertKey)))
        {
            senderPage.RegisterStartupScript(alertKey, strScript);
        }
    }
}
```

**5.34 Dica: Converter String para ArrayList usando caracter especial para quebra.**

```
public partial class _Default : System.Web.UI.Page
{
    public static ArrayList StringToArray(string pString, string
pCaracterQuebra, int pRetorno)
    {
        int nRetorno = 0;
        int nInicio = 0;
        int nQtdeChar = 0;
        string sAux = "";
```

```
if (pRetorno == 0)
{
    nRetorno = 2;
}
else
{
    nRetorno = pRetorno;
}
ArrayList alTmp = new ArrayList();
for (int I = 0; I <= pString.Length - 1; I++)
{
    nQtdeChar++;
    if (pString.Substring(I, 1) == pCaracterQuebra)
    {
        sAux = pString.Substring(nInicio, nQtdeChar-nRetorno);
        nQtdeChar = 0;
        nInicio = I+1;
        alTmp.Add(sAux);
    }
}
sAux = pString.Substring(nInicio, nQtdeChar);
alTmp.Add(sAux);
return (alTmp);
}
}
```

### 5.35 Dica: Mudar a cor de célula de uma gridview.

```
public partial class _Default : System.Web.UI.Page
{
    public static Color GridViewRowDataBound(object sender,
GridViewRowEventArgs e)
    {
        Color cor = System.Drawing.Color.Gray;
        try
        {
            if ((e.Row.RowType == DataControlRowType.DataRow))
            {
                switch (e.Row.Cells[2].Text)
                {
                    case "A":
                    {
                        cor = System.Drawing.Color.Azure;
                        break;
                    }
                    case "B":
                    {
                        cor = System.Drawing.Color.CadetBlue;
                        break;
                    }
                    case "C":
                    {
                        cor = System.Drawing.Color.LightSkyBlue;
                        break;
                    }
                    case "D":
                    {
                        cor = System.Drawing.Color.CornflowerBlue;
                        break;
                    }
                }
            }
        }
    }
}
```

```

        }
    }
}

catch
{
}

return (cor);
}
}
}

```

### 5.36 Dica: Converter HTML para String.

```

public partial class _Default : System.Web.UI.Page
{
    public static String ConvertHTML2TXT(String html)
    {
        String txt = "";
        Regex reg = new Regex("<[^>]+>", RegexOptions.IgnoreCase);
        txt = reg.Replace(html, "");
        txt = Regex.Replace(txt, @"^s*$\n", string.Empty,
RegexOptions.Multiline);
        txt = Regex.Replace(txt, "<[^>]*>", string.Empty);
        return (txt);
    }
}

```

### 5.37 Dica: Retornar um DataTable de uma SELECT.

```

public partial class _Default : System.Web.UI.Page
{
    public static DataTable DadosConsultar(string pSql)
    {
        try
        {
            DataTable oData = new DataTable();
            SqlConnection oCnx = new
SqlConnection(ConfigurationManager.AppSettings["ConnectionString"]);
            SqlCommand oCmd = new SqlCommand(pSql, oCnx);
            SqlDataAdapter oAdapter = new SqlDataAdapter(oCmd);
            oCnx.Open();
            oAdapter.Fill(oData);
            oCnx.Close();
            return (oData);
        }
        catch (Exception x)
        {
            throw (x);
        }
    }
}

```

### 5.38 Dica: Executar um comando SQL.

```

public partial class _Default : System.Web.UI.Page
{
    public static void DadosManter(string pSql)
    {
        try
        {

```

```
        SqlConnection oCnx = new
SqlConnection(ConfigurationManager.AppSettings["ConnectionString"]);
        SqlCommand oCmd = new SqlCommand(pSql, oCnx);
        SqlDataAdapter oAdapter = new SqlDataAdapter(oCmd);
        oCnx.Open();
        SqlTransaction oTrans = oCnx.BeginTransaction();
        oCmd.Transaction = oTrans;
        oCmd.ExecuteNonQuery();
        oTrans.Commit();
        oCnx.Close();
    }
    catch (Exception x)
    {
        throw x;
    }
}
```

### 5.39 Dica: Adicionar meta tags a uma página ASPX.

```
public partial class _Default : System.Web.UI.Page
{
    private void adicionarTAGS()
    {
        try
        {
            if (Request.Params["param"] != null)
            {
                HtmlMeta tag = new HtmlMeta();
                tag.Attributes.Add("property", "og:image");
                tag.Content = Session["imagem"].ToString();
                Page.Header.Controls.Add(tag);

                HtmlMeta tag2 = new HtmlMeta();
                tag2.Attributes.Add("property", "og:url");
                tag2.Content = Session["url"].ToString();
                Page.Header.Controls.Add(tag2);

                HtmlMeta tag3 = new HtmlMeta();
                tag3.Attributes.Add("property", "og:description");
                tag3.Content = Session["descricao"].ToString();
                Page.Header.Controls.Add(tag3);

                HtmlMeta tag4 = new HtmlMeta();
                tagTitle.Attributes.Add("property", "og:title");
                tag4.Content = Session["titulo"].ToString();
                Page.Header.Controls.Add(tagTitle);

                HtmlMeta tag5 = new HtmlMeta();
                tag5.Attributes.Add("itemprop", "image");
                tag5.Content = Session["imagem"].ToString();
                Page.Header.Controls.Add(tag5);
            }
        }
        catch
        {
        }
    }
}
```

**5.40 Dica: Inserir código antes e depois do código herdado.**

```
public partial class _Default : System.Web.UI.Page
{
    protected override void OnLoad(EventArgs e)
    {
        // Escrever aqui o código a ser executado antes do código herdado
do evento
        base.OnLoad(e);
        // Escrever aqui o código a ser executado depois do código herdado
do evento
    }
    protected override void OnDataBinding(EventArgs e)
    {
        // Escrever aqui o código a ser executado antes do código herdado
do evento
        base.OnDataBinding(e);
        // Escrever aqui o código a ser executado depois do código herdado
do evento
    }
    protected override void OnInit(EventArgs e)
    {
        // Escrever aqui o código a ser executado antes do código herdado
do evento
        base.OnInit(e);
        // Escrever aqui o código a ser executado depois do código herdado
do evento
    }
    protected override void OnPreRender(EventArgs e)
    {
        // Escrever aqui o código a ser executado antes do código herdado
do evento
        base.OnPreRender(e);
        // Escrever aqui o código a ser executado depois do código herdado
do evento
    }
    protected override void OnUnload(EventArgs e)
    {
        // Escrever aqui o código a ser executado antes do código herdado
do evento
        base.OnUnload(e);
        // Escrever aqui o código a ser executado depois do código herdado
do evento
    }
}
```

**5.41 Dica: Obter dados do navegador.**

```
public partial class _Default : System.Web.UI.Page
{
    protected void NavegadorDados()
    {
        try
        {
            System.Web.HttpBrowserCapabilities browser = Request.Browser;
            String[] DadosNavegador = new String[17];
            DadosNavegador[0] = browser.Type;
            DadosNavegador[1] = browser.Browser;
            DadosNavegador[2] = browser.Version;
            DadosNavegador[3] = browser.MajorVersion.ToString();
        }
    }
}
```

```
DadosNavegador[4] = browser.MinorVersion.ToString();
DadosNavegador[5] = browser.Beta.ToString();
DadosNavegador[6] = browser.Crawler.ToString();
DadosNavegador[7] = browser.AOL.ToString();
DadosNavegador[8] = browser.Win16.ToString();
DadosNavegador[9] = browser.Win32.ToString();
DadosNavegador[10] = browser.Frames.ToString();
DadosNavegador[11] = browser.Tables.ToString();
DadosNavegador[12] = browser.Cookies.ToString();
DadosNavegador[13] = browser.VBScript.ToString();
DadosNavegador[14] = browser.EcmaScriptVersion.ToString();
DadosNavegador[15] = browser.JavaApplets.ToString();
DadosNavegador[16] = browser.ActiveXControls.ToString();
Session["NavegadorTipo"] = "";
Session["Navegador"] = "Type:" + DadosNavegador[0] +
";Browser:" + DadosNavegador[0] + ";Version:" + browser.Version;
LabelVersao.ToolTip = Session["Navegador"].ToString();
if (DadosNavegador[0].IndexOf("Desktop") != -1) // Chrome
{
    Session["NavegadorTipo"] = "Chrome";
}
else if (DadosNavegador[0].IndexOf("IE") != -1) // IE
{
    Session["NavegadorTipo"] = "IE";
}
else if (DadosNavegador[0].IndexOf("Firefox") != -1) //
Mozilla Firefox
{
    Session["NavegadorTipo"] = "Firefox";
}
else // IE modo não compatibilidade e outros navegadores
{
    Session["NavegadorTipo"] = "IE";
}
}
catch
{
}
}
```

### 5.42 Dica: Limpar mensagens de erro da aplicação.

```
public partial class _Default : System.Web.UI.Page
{
    protected void Application_Error(object sender, EventArgs e)
    {
        Context.ClearError();
    }
}
```

### 5.43 Dica: Abrir uma página ASPX via javascript.

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            if (!IsPostBack)
```

```
        {
            ScriptManager.RegisterStartupScript(this, typeof(Page),
"Redirect", "window.open('Pagina.aspx', '_blank',
'menubar=0,scrollbars=1,width=440,height=370,left=" +
Session["GaleriaEsquerda"].ToString() + ",top=" +
Session["GaleriaTopo"].ToString() + ",resizable=no');", true);
        }
    }
    catch (Exception x)
    {
        throw x;
    }
}
```

### 5.44 Dica: Converter binário para uma imagem.

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            if (!IsPostBack)
            {
                Image1.ImageUrl = "data:image/jpeg;base64," +
Convert.ToBase64String((byte[])Session["Imagem"]);
            }
        }
        catch (Exception x)
        {
            throw x;
        }
    }
}
```

### 5.45 Dica: Modificar as características de uma gridview.

```
public partial class _Default : System.Web.UI.Page
{
    protected void GridView1_RowDataBound(object sender,
GridViewRowEventArgs e)
    {
        try
        {
            e.Row.Attributes["onmouseover"] = "BLOCKED
SCRIPTsetMOverColor(this);";
            e.Row.Attributes["onmouseout"] = "BLOCKED
SCRIPTsetmOutColor(this);";
            e.Row.Attributes["onclick"] =
ClientScript.GetPostBackClientHyperlink(this.GridView1, "Select$" +
e.Row.RowIndex.ToString());
            if (e.Row.RowType == DataControlRowType.DataRow)
            {
                e.Row.ToolTip = "Clique aqui para selecionar a linha";
                e.Row.Attributes.Add("onmouseover",
"this.style.cursor='pointer'");
                e.Row.Attributes.Add("onmouseout",
"this.style.cursor='hand'");
            }
        }
    }
}
```

```
                (e.Row.FindControl("LinkButton1") as
LinkButton).Attributes.Add("onmouseover",
"this.style.cursor='pointer';this.style.textDecoration='underline';this.st
yle.color='Blue'");
                (e.Row.FindControl("LinkButton1") as
LinkButton).Attributes.Add("onmouseout",
"this.style.cursor='hand';this.style.textDecoration='none';this.style.colo
r='#006793'");
            }
        catch (Exception x)
        {
            throw x;
        }
    }
}
```

### 5.46 Dica: Aplicar uma expressão regular em uma string.

```
public partial class _Default : System.Web.UI.Page
{
    protected String ExpressaoRegular(String pTexto)
    {
        Return (System.Text.RegularExpressions.Regex.Replace(pTexto,
@"<(.\|n)*?>", String.Empty));
    }
}
```

### 6 Dicas do SQL

O objetivo desse capítulo é introduzir nos conceitos de bancos de dados em especial banco de dados relacionais, focando na Linguagem de Consulta Estruturada que é uma linguagem do paradigma declarativo para banco de dados relacional baseadas na álgebra relacional.

O SQL surgiu na década de 70 na IBM baseado no modelo relacional de Codd. É o padrão para banco de dados relacionais e é mantido pela *American National Standards Institute* (ANSI) e *International Standards Organization* (ISO). Existindo várias versões revisadas do SQL ANSI/ISO, as quais são: 1986, 1987, 1992, 1999 e 2003. E além dessas padronizações, existem as extensões inseridas pelos desenvolvedores de banco de dados.

#### 6.1 Structure Query Language (SQL)

Iniciando o estudo do SQL temos as seguintes categorias:

- *Data Definition Language* (DDL): Create ..., Drop ..., Alter ...;
- *Data Manipulation Language* (DML): Insert ..., Update ..., Delete ...;
- *Data Query Language* (DQL): Select ...;
- *Data Transaction Language* (DTL): Savepoint ..., Rollback ..., Commit ...;
- *Data Control Language* (DCL): Grant ..., Revoke ...;

##### 6.1.1 Inserção de Metadados (Criação de Tabelas)

Comando que permite criar as estruturas de armazenamento de dados.

###### 6.1.1.1 Sintaxe

CREATE TABLE nome\_tabela [(definição\_create,...)] [table\_options] [select\_statement]

definição\_create:

```
nome_coluna tipo [NOT NULL | NULL] [DEFAULT valor_padrão]
[[PRIMARY] KEY]
| [CONSTRAINT [symbol]] PRIMARY KEY (index_col_name,...)
| INDEX [nome_indice] (index_nome_coluna,...)
| [CONSTRAINT [symbol]] UNIQUE [INDEX] [index_name] (index_col_name,...)
| [CONSTRAINT [symbol]] FOREIGN KEY [index_name] (index_col_name,...)
```

definição\_referência:

```
REFERENCES nome_tabela [(index_nome_coluna,...)]
[ON DELETE opção_referência]
[ON UPDATE opção_referência]
```

opção\_referência:

```
RESTRICT | CASCADE | SET NULL | SET DEFAULT
```

###### 6.1.1.2 Dicas

É possível criar uma tabela a partir de uma subconsulta;

CREATE TABLE tabela [coluna, ...] AS subconsulta.

###### 6.1.1.3 Práticas

Criar a seguintes tabelas: Tmp: Número (PK), número inteiro. Cargo: Número (PK), número inteiro, auto-incremento; Nome, literal variável(30), não nulo. Departamento: Número (PK), número inteiro; Nome, literal variável(30), não nulo; Fator, número decimal com precisão 3 e escala 2. Empregado: Número (PK), número inteiro; Nome, literal variável (50), não nulo; Departamento, (FK) numérico, atualização cascata, exclusão restrita.

```
CREATE TABLE CARGO (
  NUMERO INTEGER IDENTITY( 1, 1),
  NOME VARCHAR(30) NOT NULL,
  PRIMARY KEY (NUMERO)
```

<http://www.alvarofpinheiro.eti.br>

```
);  
  
CREATE TABLE DEPARTAMENTO (  
    NUMERO INTEGER,  
    NOME VARCHAR(30) NOT NULL,  
    FATOR DECIMAL(3,2),  
    PRIMARY KEY (NUMERO)  
);  
  
CREATE TABLE EMPREGADO (  
    NUMERO INTEGER,  
    NOME VARCHAR(50) NOT NULL,  
    DEPARTAMENTO INTEGER,  
    PRIMARY KEY (NUMERO),  
    FOREIGN KEY (DEPARTAMENTO) REFERENCES DEPARTAMENTO (NUMERO) ON UPDATE  
    CASCADE  
);
```

### 6.1.2 Inserção de Dados (Linhas | Registros | Ocorrências | Tuplas)

Comando que permite incluir os dados nas estruturas de armazenamento.

#### 6.1.2.1 Sintaxe

```
INSERT [INTO] nome_tabela [(nome_coluna,...)]  
VALUES ((expressão | DEFAULT),...),(...),...  
| [INTO] nome_tabela [(nome_coluna,...)]  
SELECT ...
```

#### 6.1.2.2 Dicas

Pode-se inserir linha a linha;  
Pode-se dependendo do BD inserir uma lista;  
Pode-se inserir linhas de uma tabela em outra;  
Não se usa a cláusula VALUE e sim SELECT.

#### 6.1.2.3 Práticas

Inserir esses dados nas tabela Cargo: Número, Nome; 1, Analista Trainee; 2, Analista Junior; 3, Analista Pleno; 4, Analista Sênior; 5, Analista Master:

```
INSERT INTO CARGO (NOME) VALUES ('ANALISTA TRAINEE');  
INSERT INTO CARGO (NOME) VALUES ('ANALISTA JUNIOR');  
INSERT INTO CARGO (NOME) VALUES ('ANALISTA PLENO');  
INSERT INTO CARGO (NOME) VALUES ('ANALISTA SÊNIOR');  
INSERT INTO CARGO (NOME) VALUES ('ANALISTA MASTER');  
INSERT INTO CARGO (NOME) VALUES ('CONSULTOR');
```

Departamento: Número, Nome, Fator; 10, Administração (Admin), 1.75; 20, Marketing (Mkt), 1.85; 50, Compras (Com), 2. 73; 60, Tecnologia (Tec), 1.55; 80, Vendas (Ven), 1.87; 90, Executivo (Exec), 1.92; 110, Contabilidade (Contab), 2.35; 190, Contratos (Ctt), 2.23:

```
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (10, 'ADMINISTRAÇÃO  
(ADMIN)', 1.75);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (20, 'MARKETING (MKT)',  
1.85);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (50, 'COMPRAS (COM)',  
2.73);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (60, 'TECNOLOGIA (TEC)',  
1.55);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (80, 'VENDAS (VEN)', 1.87);
```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (90, 'EXECUTIVO (EXEC)', 1.92);
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (110, 'CONTABILIDADE (CONTAB)', 2.35);
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (190, 'CONTRATOS (CTT)', 2.23);
```

Inserir dados na tabela Empregados: Número, Nome, Departamento; 100, Antônio Palmeira, 10; 101, José Figueira, 20; 102, Maria Mangueira, 10; 103, João Abacateiro, 50; 104, Carlos Castanheiro, 60; 107, Mônica Videira, 90; 124, Pedro Laranjeira, 80; 141, Tiago Pessegueiro, 110; 142, Renata Videira, 80; 143, Gabriel Macieira, 110:

```
INSERT INTO EMPREGADO VALUES (100, 'ANTÔNIO PALMEIRA', 10);
INSERT INTO EMPREGADO VALUES (101, 'JOSÉ FIGUEIRA', 20);
INSERT INTO EMPREGADO VALUES (102, 'MARIA MANGUEIRA', 10);
INSERT INTO EMPREGADO VALUES (103, 'JOÃO ABACATEIRO', 50);
INSERT INTO EMPREGADO VALUES (104, 'CARLOS CASTANHEIRO', 60);
INSERT INTO EMPREGADO VALUES (107, 'MÔNICA VIDEIRA', 90);
INSERT INTO EMPREGADO VALUES (124, 'PEDRO LARANJEIRA', 80);
INSERT INTO EMPREGADO VALUES (141, 'TIAGO PESSEQUEIRO', 110);
INSERT INTO EMPREGADO VALUES (142, 'RENATA VIDEIRA', 80);
INSERT INTO EMPREGADO VALUES (143, 'GABRIEL MACIEIRA', 110);
```

### 6.1.3 Alteração de Metadados (Modificar Estruturas)

Comando que permite modificar as estruturas de armazenamento.

#### 6.1.3.1 Sintaxe

ALTER TABLE nome\_tbl especificação\_alter [, especificação\_alter ...]  
especificação\_alter:

```
  ADD [COLUMN] definição_create [FIRST | AFTER nome_coluna]
| ADD [COLUMN] (definição_create, definição_create,...)
| ADD INDEX [nome_indice] (index_nome_col,...)
| ADD [CONSTRAINT [symbol]] PRIMARY KEY (index_col_name,...)
| ADD [CONSTRAINT [symbol]] UNIQUE [index_name] (index_col_name,...)
| ADD [CONSTRAINT [symbol]] FOREIGN KEY [index_name] (index_col_name,...)
  [definição_referencia]
| ALTER [COLUMN] nome_col {SET DEFAULT literal | DROP DEFAULT}
| CHANGE [COLUMN] nome_col_antigo definição_create
  [FIRST | AFTER nome_coluna]
| MODIFY [COLUMN] definição_create [FIRST | AFTER nome_coluna]
| DROP [COLUMN] nome_col
| DROP PRIMARY KEY
| DROP INDEX nome_indice
| RENAME [TO] nome_nova_tbl
| ORDER BY col
```

#### 6.1.3.2 Práticas

Alterar os dados da tabela Empregado adicionando as colunas: Primeironome, literal variável (25); Sobrenome, literal variável (25); Departamento, (FK) número inteiro; Cargo, (FK) número inteiro, atualização cascata; Salário, número real; Comissão, número inteiro; Nascimento, data; Apelido, literal fixo(10), não nulo; Gerente, número inteiro:

```
ALTER TABLE EMPREGADO
ADD PRIMEIRONOME VARCHAR(25),
SOBRENOME VARCHAR(25),
CARGO INTEGER,
SALARIO REAL,
```

<http://www.alvarofpinheiro.eti.br>

```
COMISSAO INTEGER,  
NASCIMENTO DATE,  
APELIDO VARCHAR(10),  
GERENTE INTEGER;
```

```
ALTER TABLE EMPREGADO  
ADD FOREIGN KEY (CARGO) REFERENCES CARGO (NUMERO) ON UPDATE CASCADE,  
FOREIGN KEY (GERENTE) REFERENCES EMPREGADO (NUMERO);
```

```
ALTER TABLE EMPREGADO  
DROP COLUMN NOME;
```

#### **6.1.4 Alteração de Dados (Atualização)**

```
UPDATE nome_tabela  
    SET nome_coluna1=expr1 [, nome_coluna2=expr2 ...]  
    [WHERE definição]  
    [ORDER BY ...]
```

##### **6.1.4.1 Dicas**

Pode-se alterar linha;  
Pode-se alterar listas;  
Pode-se alterar subconsultas;  
Pode-se alterar linhas com base em outra tabela.

##### **6.1.4.2 Práticas**

Atualizar as linhas de Empregado com os valores: Número, Primeironome, Sobrenome, Cargo, Departamento, Gerente, Salário, Comissão, Nascimento, Apelido; 100, Antônio, Palmeira, 1, 10, nulo, 1000, nulo, 01/07/80, Tonho; 101, José, Figueira, 1, 20, 100, 1500, nulo, 18/05/70, Zé; 102, Maria, Mangueira, 2, 10, 100, 2000, 3, 22/07/80, Ma; 103, João, Abacateiro, 3, 50, 100, 1500, 2, 03/03/71, Jo; 104, Carlos, Castanheiro, nulo, 60, 100, 2500, 3, 01/01/65, Ca; 107, Mônica, Videira, 1, 90, nulo, 1000, 2, 15/02/60, Mo; 124, Pedro, Laranjeira, 2, 80, 107, 1100, nulo, 30/07/83, Pedroca; 141, Tiago, Pessegueiro, 3, 80, 107, 2000, 2, 21/06/78, Ti; 142, Renata, Videira, 4, 80, 107, 1500, nulo, 02/12/72, Rê; 143, Gabriel, Macieira, nulo, 110, 107, 2000, 2, 11/11/81, Biel:

```
UPDATE EMPREGADO SET  
CARGO=1, GERENTE=NULL, SALARIO=1000, COMISSAO=NULL,  
NASCIMENTO='01/07/80', APELIDO='TONHO' WHERE NUMERO=100;  
UPDATE EMPREGADO SET  
CARGO=1, GERENTE=100, SALARIO=1500, COMISSAO=NULL,  
NASCIMENTO='18/05/70', APELIDO='ZÉ' WHERE NUMERO=101;  
UPDATE EMPREGADO SET  
CARGO=2, GERENTE=100, SALARIO=2000, COMISSAO=3,  
NASCIMENTO='22/07/80', APELIDO='MA' WHERE NUMERO=102;  
UPDATE EMPREGADO SET  
CARGO=3, GERENTE=100, SALARIO=1500, COMISSAO=2,  
NASCIMENTO='03/03/71', APELIDO='JO' WHERE NUMERO=103;  
UPDATE EMPREGADO SET  
CARGO=NULL, GERENTE=100, SALARIO=2500, COMISSAO=3,  
NASCIMENTO='01/01/65', APELIDO='CA' WHERE NUMERO=104;  
UPDATE EMPREGADO SET  
CARGO=1, GERENTE=NULL, SALARIO=1000, COMISSAO=2,  
NASCIMENTO='15/02/60', APELIDO='MO' WHERE NUMERO=107;  
UPDATE EMPREGADO SET  
CARGO=2, GERENTE=107, SALARIO=1100, COMISSAO=NULL,  
NASCIMENTO='30/07/83', APELIDO='PEDROCA' WHERE NUMERO=124;  
UPDATE EMPREGADO SET  
CARGO=3, GERENTE=107, SALARIO=2000, COMISSAO=2,  
NASCIMENTO='21/06/78', APELIDO='TI' WHERE NUMERO=141;
```

<http://www.alvarofpinheiro.eti.br>

```
UPDATE EMPREGADO SET  
CARGO=4, GERENTE=107, SALARIO=1500, COMISSAO=NULL,  
NASCIMENTO='02/12/72', APELIDO='RE' WHERE NUMERO=142;  
UPDATE EMPREGADO SET  
CARGO=NULL, GERENTE=107, SALARIO=2000, COMISSAO=2,  
NASCIMENTO='11/11/81', APELIDO='BIEL' WHERE NUMERO=143;
```

### 6.1.5 Algumas Funções

Permitem aumentar a capacidade de execução de consultas dos bancos de dados.

#### 6.1.5.1 Sintaxe

SQLServer | Postgre | MySQL | Firebird

**SUBSTRING**(*String* texto, posicao\_inicial, tamanho) - retorna uma *String* com o comprimento definido em "tamanho" extraída da *String* "texto", a partir da "posicao\_inicial";

Oracle: SUBSTR

**CHARINDEX**( *String* pesquisado, campo, posicao\_inicial) – retorna a posição do *String* pesquisado dentro do campo informado.

#### 6.1.5.2 Práticas

Fazendo uso das funções subString e charindex separar o conteúdo do campo Nome nos campos Primeironome e Sobrenome. Depois apagar o campo Nome ficando: Número, Primeironome, Sobrenome, Departamento; 100, Antônio, Palmeira, 10; 101, José, Figueira, 20; 102, Maria, Mangueira, 10; 103, João, Abacateiro, 50; 104, Carlos, Castanheiro, 60; 107, Mônica, Videira, 90; 124, Pedro, Laranjeira, 80; 141, Tiago, Pessegueiro, 80; 142, Renata, Videira, 80; 143, Gabriel, Macieira, 110:

```
UPDATE EMPREGADO SET  
PRIMEIRONOME = SUBSTRING(NOME,1,CHARINDEX(' ',NOME)-1),  
SOBRENOME = SUBSTRING(NOME,CHARINDEX(' ',nome)+1,100);
```

### 6.1.6 Índices

Permite criar estruturas de indexação que auxiliam as estruturas de armazenamento aumentando a performance no acesso aos dados.

#### 6.1.6.1 Sintaxe

```
CREATE [UNIQUE] INDEX nome_indice  
ON nome_tabela (index_col_name,...)  
index_col_name:  
col_name [(length)] [ASC | DESC]
```

Trata-se de uma referência associada a uma chave, que tem a finalidade de otimizar a pesquisa, permitindo assim a localização rápida de um registro quando efetuada uma consulta. Trata-se de uma estrutura auxiliar associado a uma tabela, que possui ponteiros para os dados armazenados em colunas específicas. Fazendo uma analogia, o banco de dados usa o índice de maneira semelhante ao índice remissivo de um livro, verificando um determinado assunto no índice e depois localizando a sua posição.

#### 6.1.6.2 Práticas

Criar índices por: Primeironome e Sobrenome; Sobrenome; Salário; Comissão; Cargo; Departamento; Apelido (Único):

```
CREATE INDEX EMPREGADOIX1 ON EMPREGADO (PRIMEIRONOME);  
CREATE INDEX EMPREGADOIX2 ON EMPREGADO (SOBRENOME);  
CREATE INDEX EMPREGADOIX3 ON EMPREGADO (SALARIO);  
CREATE INDEX EMPREGADOIX4 ON EMPREGADO (COMISSAO);  
CREATE INDEX EMPREGADOIX5 ON EMPREGADO (CARGO);
```

```
CREATE INDEX EMPREGADOIX6 ON EMPREGADO (DEPARTAMENTO);  
CREATE UNIQUE INDEX EMPREGADOIX7 ON EMPREGADO (APELIDO);
```

### 6.1.7 Dicionário de Dados

Trata-se de uma coleção de metadados que contêm definições e representações de elementos de dados. Os Sistemas Gerenciadores de Bancos de Dados (SGBD's) fazem uso dos dicionários de dados, permitindo apenas consulta por parte dos usuários, e possuem as seguintes informações: Definição dos elementos de dados; Perfis de usuários, papéis e privilégios; Descrição de objetos; Integridade de restrições; Informação de verificação; e Alocações de espaço.

#### 6.1.7.1 Práticas

Visualizar a estrutura das tabelas Empregados e Departamentos:

```
SELECT * FROM SYSOBJECTS O INNER JOIN SYSCOLUMNS C ON O.ID=C.ID WHERE O.NAME  
= 'DEPARTAMENTO';
```

### 6.1.8 Consulta de Dados

Comando que permite extrair os dados das estruturas de armazenamento como desejado pelo usuários.

#### 6.1.8.1 Sintaxe

```
SELECT  
[ ALL | DISTINCT ]  
[ TOP n ]  
[ table_name. | table_alias. | view_name. ]  
*  
  
[ column_name [ [ AS ] column_alias ]  
| expression [ [ AS ] column_alias ]  
[ ..., n ]  
[ INTO new_table ]  
FROM table  
[ AS table_alias ]  
[ ( column_alias_1, [ column_alias_2, ..., n ] ) ]  
|  
view_name [ AS table_alias ]  
[ INNER JOIN | LEFT [ OUTER ] JOIN | RIGHT [ OUTER ] JOIN table [ ON search_conditions ]  
|  
[ WHERE search_conditions  
|  
WHERE column_name =* column_name  
]  
[ GROUP BY { group_by_expression, [ ..., n ] } ]  
[ HAVING search_conditions ]  
[ UNION query ]  
[ ORDER BY { order_by_expression [ ASC | DESC ], [ ..., n ] } ]
```

#### 6.1.8.2 Dicas

Operador de concatenação (||);

Em alguns BD se usa (+) para concatenação;

Concatena colunas ou *Strings* de caracteres a outras colunas.

#### 6.1.8.3 Práticas

Visualizar os campos: Número e Nome (Primeironome || ' ' || Sobrenome):

```
SELECT NUMERO, PRIMEIRONOME + ' ' + SOBRENOME AS NOME FROM EMPREGADO;
```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

Visualizar os registros dos empregados do departamento 20 ou 80 dos campos: Número; Primeironome; Sobrenome; e Departamento:

```
SELECT NUMERO, PRIMEIRONOME, SOBRENOME, DEPARTAMENTO  
FROM EMPREGADO  
WHERE DEPARTAMENTO = 20 OR DEPARTAMENTO = 80;
```

Visualizar os registros dos empregados juntamente com o nome do departamento: Número (Empregados); Primeironome (Empregados); Sobrenome (Empregados); Departamento (Empregados); e Nome (Departamentos).

```
SELECT E.NUMERO, E.PRIMEIRONOME, E.SOBRENOME, E.DEPARTAMENTO,  
D.NOME  
FROM EMPREGADO E INNER JOIN DEPARTAMENTO D  
ON E.DEPARTAMENTO = D.NUMERO;
```

### 6.1.9 Precedência

Use parênteses (...) para forçar a ordem em que as expressões serão avaliadas.

### 6.1.10 Comparação

Se um ou ambos os argumentos são NULL, o resultado da comparação é NULL; Se ambos os argumentos em uma comparação são *Strings*, eles são comparados como *Strings*; Se ambos os argumentos são inteiros, eles são comparados como inteiros; Valores hexadecimais são tratados como *Strings* binárias se não comparadas a um número; Se um dos argumentos é uma coluna *TIMESTAMP* ou *DATETIME* e o outro argumento é uma constante, a constante é convertida para um timestamp antes da comparação ser realizada; Por padrão, comparações de *String* são feitas de modo independente do caso, usando o conjunto de caracteres atual (ISO-8859-1 Latin1 por padrão); Comparação *Strings* com distinção de letras maiúsculas e minúsculas pode ser realizado com qualquer dos operadores, exceto o *LIKE*; Espaços em branco no fim da *String*, espaços, tabs e quebra de linha serão ignorados.

### 6.1.11 Projeção

Consiste na escolha das colunas que serão exibidas.

### 6.1.12 Seleção

Consiste na escolha das linhas que serão exibidas.

### 6.1.13 Junção

Consiste em relacionar dados de duas ou mais tabelas ou mesmo relacionar uma tabela com ela mesma podendo ser uma correlação ou autojunção.

### 6.1.14 Consulta Básica

Consiste em usar os argumentos básicos do comando Select que são o *SELECT* e *FROM*.

#### 6.1.14.1 Dicas

As instruções SQL não fazem distinção entre maiúsculas e minúsculas;

As instruções SQL podem estar em uma ou mais linhas;

As palavras-chave não podem ser abreviadas ou quebradas;

Normalmente as cláusulas são colocadas em linhas separadas;

Os recuos são utilizados para dar mais legibilidade.

#### 6.1.14.2 Práticas

Visualizar todos os registros e todos os campos da tabela departamentos.

```
SELECT * FROM DEPARTAMENTO;
```

### 6.1.15 Apelido (Alias)

Consiste em atribuir outro nome a um metadado mas que não irá afetar a estrutura física é apenas uma alteração lógica.

#### 6.1.15.1 Práticas

Visualizar todos os registros nos campos Número, Nome (Primeironome + ' ' + Sobrenome) e após a consulta ser realizada observar o nome do campo para a expressão Nome. Finalize repetindo a consulta colocando o apelido Nome do Funcionário.

```
SELECT NUMERO, PRIMEIRONOME + ' ' + SOBRENOME FROM EMPREGADO;  
SELECT NUMERO, PRIMEIRONOME + ' ' + SOBRENOME AS NOME FROM EMPREGADO;
```

### 6.1.16 Campos Calculados

São campos lógicos baseados em operações aritméticas fazendo uso dos operadores e funções.

Operadores aritméticos:

- Adicionar: +
- Subtrair: -
- Multiplicar: \*
- Dividir: / “retorna o quociente”

Funções matemáticas:

- Potenciação: POWER(número, potência)
- Radiciação: SQRT(número)
- Divisão: % | MOD “retorna o resto”

Precedência de Operadores:

- A multiplicação e a divisão têm prioridade sobre a adição e subtração;
- Os operadores com a mesma prioridade são avaliados da esquerda para a direita;
- Os parênteses são usados para forçar a avaliação priorizada e para esclarecer as instruções.

#### 6.1.16.1 Práticas

Visualizar o Sobrenome, o salário, e o líquido dos empregados, considerando que todos possuem um desconto de 10% de INSS.

```
SELECT SOBRENOME, SALARIO, SALARIO - (SALARIO*0.10) AS LIQUIDO  
FROM EMPREGADO;
```

### 6.1.17 Valores Nulos

São campos que não possuem nenhum tipo de dado o que chamamos de valor nulo.

#### 6.1.17.1 Dicas

Nulo é uma valor que não está disponível;  
Nulo é um valor desconhecido;  
Nulo não é a mesma coisa que zero;  
Nulo não é a mesma coisa que espaço em branco;  
Para SQLServer usar ISNULL(campo, valor);

#### 6.1.17.2 Práticas

Visualizar o Sobrenome e o Salário Bruto (Salário + Comissão). Obs: A Comissão representa um percentual aplicado sobre o salário.

```
SELECT SOBRENOME, SALARIO + (SALARIO * COMISSAO / 100) AS BRUTO  
FROM EMPREGADO;
```

Visualizar o Sobrenome e o Salário Bruto (Salário + Comissão). Obs: Quando a Comissão for nula deve-se substituir por zero.

```
SELECT SOBRENOME, SALARIO + (SALARIO * ISNULL(COMISSAO,0) / 100) AS BRUTO FROM  
EMPREGADO;
```

#### 6.1.18 Duplicidade

Consiste em eliminar na consulta, isto é, apenas na visualização dos dados, as duplicidades ou valores repetidos.

##### 6.1.18.1 Dicas

A exibição default das consultas é de todas as linhas, incluindo linhas duplicadas.

##### 6.1.18.2 Práticas

Visualizar o Sobrenome dos Empregados.

```
SELECT SOBRENOME FROM EMPREGADO;
```

Visualizar o Sobrenome dos Empregados sem duplicidade.

```
SELECT DISTINCT SOBRENOME FROM EMPREGADO;
```

#### 6.1.19 Condicionamento

São seleções de dados resultantes de condicionamentos lógicos realizados pela cláusula WHERE.

Operadores de Comparação:

- Igual a: =
- Maior que: >
- Maior que ou igual a: >=
- Menor que: <
- Menor que ou igual a: <=
- Diferente de: <>

Operadores Lógicos:

- Verdadeiro para ambas expressões verdadeiras: AND
- Verdadeiro bastando uma das expressões verdadeiras: OR
- Inverso da expressão: NOT

Regras de Precedência

- Operadores Aritméticos;
- Operador de Concatenação;
- Condições de Comparação;
- IS [NOT] NULL, [NOT] LIKE, [NOT] IN;
- [NOT] BETWEEN;
- Condição lógica NOT;
- Condição lógica AND;
- Condição lógica OR.

##### 6.1.19.1 Dicas

As *Strings* de caracteres e valores de data aparecem entre aspas simples;

Os valores de caractere fazem distinção entre maiúsculas e minúsculas;

Os valores de data fazem distinção entre formatos;

O formato de data default é DD-MON-YY.

### 6.1.19.2 Práticas

Visualizar o Número e Sobrenome de todos os empregados do departamento 90.

```
SELECT NUMERO, SOBRENOME  
FROM EMPREGADO  
WHERE DEPARTAMENTO = 90;
```

Visualizar o Número, Sobrenome e Nascimento de todos os empregados que nasceram em 22/07/80.

```
SELECT NUMERO, SOBRENOME, NASCIMENTO FROM EMPREGADO WHERE NASCIMENTO =  
'22/07/80';
```

Visualizar o Número, Sobrenome e Nascimento de todos os empregados que nasceram entre (inclusive) 01/01/70 e 31/12/80.

```
SELECT NUMERO, SOBRENOME, NASCIMENTO FROM EMPREGADO WHERE NASCIMENTO  
BETWEEN '01/01/70' AND '31/12/80';
```

Visualizar o Número, Sobrenome e Departamento de todos os empregados que pertencem aos departamentos: 10, 20, 50, 60, 80 ou 90.

```
SELECT NUMERO, SOBRENOME, DEPARTAMENTO  
FROM EMPREGADO  
WHERE DEPARTAMENTO IN (10, 20, 50, 60, 80, 90);
```

Visualizar o Número e Sobrenome de todos os empregados que possuem 'eira' no sobrenome.

```
SELECT NUMERO, SOBRENOME  
FROM EMPREGADO  
WHERE SOBRENOME LIKE '%EIRA%';
```

Visualizar o Número e Sobrenome de todos os empregados que não possuem comissão.

```
SELECT NUMERO, SOBRENOME  
FROM EMPREGADO  
WHERE COMISSAO IS NULL;
```

Visualizar o Número, Sobrenome e Salário de todos os empregados que possuem salário entre (inclusive) 1500 e 2000, dos departamentos 20, 50 e 80.

```
SELECT NUMERO, SOBRENOME, SALARIO, DEPARTAMENTO  
FROM EMPREGADO  
WHERE SALARIO BETWEEN 1500 AND 2000  
AND DEPARTAMENTO IN (20, 50, 80);
```

### 6.1.20 Ordenação

Permite realizar classificação, isto é, uma ordenação dos dados que serão exibidos.

#### 6.1.20.1 Sintaxe

```
[ORDER BY {coluna, expr} [ASC|DESC]];
```

#### 6.1.20.2 Dicas

É possível utilizar o alias na cláusula de ordenação;

É possível utilizar a posição da coluna na ordenação;

Pode-se classificar por colunas que não estejam no select;

<http://www.alvarofpinheiro.eti.br>

Pode-se classificar por N colunas.

### 6.1.20.3 Práticas

Visualizar o Sobrenome e o Salário dos Empregados ordenados por Salário em ordem decrescente.

```
SELECT SOBRENOME, SALARIO  
FROM EMPREGADO  
ORDER BY SALARIO DESC;
```

### 6.1.21 Aninhamento

São comandos ou funções que se apresentam uma dentro da outra.

#### 6.1.21.1 Dicas

Funções de uma única linha podem ser aninhadas em qualquer nível;  
As funções aninhadas são avaliadas do nível mais interno para o mais externo.

#### 6.1.21.2 Práticas

Exibir o tamanho do Apelido concatenado com o Sobrenome com os brancos e depois sem os brancos; Exibir o tamanho do Apelido concatenado com o Sobrenome com os brancos

```
SELECT LEN(APELIDO + ' ' + SOBRENOME) FROM EMPREGADO;
```

### 6.1.22 Funções

Funções são recursos avançados do SQL e podem ser utilizadas para efetuar o seguinte: Executar cálculos; Modificar itens de dados individuais; Manipular a saída para grupos de linhas; Formatar datas e números para exibição; Converter tipos de dados de coluna; Algumas funções utilizam argumentos; Sempre retornam um valor.

#### 6.1.22.1 Tipos

Funções de uma Única Linha - operam somente linhas únicas e retornam um resultado por linha;  
Funções de Várias Linhas - podem manipular grupos de linhas para fornecer um resultado e são mais conhecidas como funções de grupo.

#### 6.1.22.2 Dicas

Servem para manipular itens de dados;  
Aceitam um ou mais argumentos;  
Retornam um valor para cada linha retornada;  
São do tipo: Caractere, Numéricas, Data, Conversão e Gerais;  
Podem ser usadas em cláusulas: SELECT, WHERE e ORDER BY.

### 6.1.23 Funções Caractere

LOWER – converte para minúsculas;  
UPPER – converte para maiúsculas;  
SUBSTRING – retorna o subString de uma String;  
LENGTH – retorna o número de caracteres;  
CHARINDEX – retorna a posição de um String;  
REPLACE - substitui uma String por outra String;  
RTRIM | LTRIM | TRIM – retira brancos.

#### 6.1.23.1 Práticas

Exibir todos os Primeiros nomes em minúsculas.

```
SELECT LOWER(PRIMEIRONOME) AS PRIMEIRONOME FROM EMPREGADO;
```

Exibir todos os Sobrenomes em maiúsculas.

```
SELECT UPPER(SOBRENOME) AS PRIMEIRONOME FROM EMPREGADO;
```

Exibir os 3 primeiros caracteres do Sobrenome.

```
SELECT SUBSTRING(SOBRENOME,1,3) FROM EMPREGADO;
```

Exibir o tamanho do Sobrenome.

```
SELECT LEN(SOBRENOME) FROM EMPREGADO;
```

Exibir a posição no '(' no nome do departamento.

```
SELECT CHARINDEX('(', NOME) FROM DEPARTAMENTO;
```

### 6.1.24 Funções Numérica

ROUND – arredonda a coluna;

TRUNC – trunca a coluna;

#### 6.1.24.1 Práticas

Exibir o Fator.

```
SELECT FATOR FROM DEPARTAMENTO;
```

Exibir o Fator arredondando para 1 decimal.

```
SELECT ROUND(FATOR,1) FROM DEPARTAMENTO;
```

Exibir o Fator truncando para 1 decimal.

```
--SELECT ROUND(FATOR, 1, 2) FROM DEPARTAMENTO;
```

### 6.1.25 MOD | %

Retorna o resto da divisão.

#### 6.1.25.1 Dicas

Para exibir dados que não estão inseridos em tabelas do banco de dados pode-se fazer uso de uma tabela fictícia usada para exibir resultados chamada de DUAL.

#### 6.1.25.2 Práticas

Visualizar o quociente da divisão de 5 por 2.

```
SELECT 5 / 2 FROM TMP;
```

Visualizar o resto da divisão de 5 por 2.

```
SELECT 5 % 2 FROM TMP;
```

### 6.1.26 Funções Data

Retornam dados no formato de datas.

#### 6.1.26.1 Sintaxe

```
GETDATE | SYSDATE
```

#### 6.1.26.2 Dicas

Os Bancos de Dados (BD) armazenam as datas em formato de número inteiro: século, ano, mês, dia, horas, minutos e segundos;

O formato de exibição default depende do BD, mas no geral é utilizado a exibição DD-MMM-AA;

Pode-se adicionar ou subtrair (aritmética com datas)

Operação; Resultado; Descrição  
Data+Número; Data; Adiciona dias a datas  
Data-Número; Data; Subtrai dias a datas  
Data-Data; Dias; Subtrai uma data de outra  
Data+Número/24; Data; Adiciona hora a datas

### 6.1.26.3 Práticas

Exiba e verifique o retorno: Exibir a data corrente; Exibir Data+Número; Exibir Data-Número; Exibir Data-Data; Exibir Data+Número/24.

```
SELECT GETDATE() FROM TMP;  
SELECT GETDATE() + 1 FROM TMP;  
SELECT GETDATE() - 1 FROM TMP;  
SELECT GETDATE() - '13/11/67' FROM TMP;  
SELECT GETDATE() + 10/24 FROM TMP;  
SELECT YEAR(GETDATE()) FROM TMP;  
SELECT MONTH(GETDATE()) FROM TMP;  
SELECT DAY(GETDATE()) FROM TMP;
```

### 6.1.27 Função Conversão

Conversão de dados entre diversos tipos.

#### 6.1.27.1 Sintaxe

CAST

#### 6.1.27.2 Práticas

Exibir a conversão do salário para *String*.

```
SELECT CAST(SALARIO AS CHAR(10)) FROM EMPREGADO;
```

### 6.1.28 Função de Busca e Troca

Recupera e altera dados.

#### 6.1.28.1 Sintaxe

REPLACE

#### 6.1.28.2 Práticas

Trocar no nome de departamento: Admin para Adm; Exec para Exe; e Contab para Con.

```
SELECT  
REPLACE(NOME, '(ADMIN)', '(ADM)'),  
REPLACE(NOME, '(EXEC)', '(EXE)'),  
REPLACE(NOME, '(CONTAB)', '(CON)')  
FROM DEPARTAMENTO;
```

### 6.1.29 Função de Condicionamento

Condiciona a exibição dos dados permitindo o uso de expressões condicionais.

#### 6.1.29.1 Sintaxe

```
CASE expr WHEN cond THEN retorno  
[WHEN condN THEN retornoN]  
[ELSE retorno]  
END
```

#### 6.1.29.2 Práticas

Visualizar Número, Sobrenome, Salário e Departamento dos empregados;

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

Visualizar Número, Sobrenome, Salário e Departamento dos empregados aplicando 10% de aumento nos salários dos depart. 10 e 20, 15% dos depart. 80 e 90, e 5% nos demais.

```
SELECT NUMERO, SOBRENOME,  
CASE  
WHEN DEPARTAMENTO IN (10,20) THEN SALARIO + (SALARIO*0.10)  
WHEN DEPARTAMENTO IN (80,90) THEN SALARIO + (SALARIO*0.15)  
ELSE SALARIO + (SALARIO*0.5)  
END AS SALARIO,  
DEPARTAMENTO  
FROM EMPREGADO;
```

### 6.1.30 Produto Cartesiano

Um produto cartesiano será formado quando uma condição de junção for omitida e como consequência todas as linhas da primeira tabela serão unidas a todas as linhas da segunda tabela.

#### 6.1.30.1 Práticas

Visualizar Número, Sobrenome e Nome de Departamento dos empregados não colocando a junção das tabelas e observe o resultado.

```
SELECT E.NUMERO, E.SOBRENOME, D.NOME FROM EMPREGADO E, DEPARTAMENTO D;  
SELECT E.NUMERO, E.SOBRENOME, D.NOME FROM EMPREGADO E CROSS JOIN  
DEPARTAMENTO D;
```

### 6.1.31 Junção

Quando forem necessários dados de mais de uma tabela no banco de dados, será usada uma condição de junção. As linhas de uma tabela podem ser unidas a linhas de outra tabela de acordo com os valores comuns existentes nas colunas correspondentes, ou seja, em geral colunas de chave primária e chave estrangeira.

#### 6.1.31.1 Tipos de Junção

- Equijunção
- Não-equijunção
- Junção externa
- Autojunção

#### 6.1.31.2 Dicas

Quando criar uma instrução SELECT que una tabelas, coloque o nome da tabela antes do nome da coluna para fins de clareza e para melhorar o acesso ao banco de dados;

Caso apareça o mesmo nome de coluna em mais de uma tabela, o nome da coluna deve vir antecedido do nome da tabela (AMBIGUIDADE);

Para unir N tabelas, é necessário um mínimo de N-1 condições de junção. Por exemplo, para unir 4 tabelas, é necessário um mínimo de 3 junções. Essa regra pode não se aplicar se sua tabela tiver uma chave primária concatenada. Nesse caso, será necessário mais de uma coluna para identificar exclusivamente cada linha;

Para melhorar a sintaxe deve-se apelidar as tabelas.

### 6.1.32 Equijunção

Junção por igualdade.

#### 6.1.32.1 Dicas

Equijunção também são chamadas de junções simples ou internas;

Observe que nas junções simples ou internas se não existe uma relação a tupla não é gerada.

Tupla: Cada linha formada por uma lista ordenada de colunas representa um registro. Os registros não precisam conter informações em todas as colunas, podendo assumir valores nulos quando assim se fizer necessário.

### 6.1.32.2 Práticas

Exibir o Sobrenome do empregado e o Nome do departamento de um empregado.

```
SELECT E.SOBRENOME, D.NOME  
FROM EMPREGADO E  
INNER JOIN DEPARTAMENTO D  
ON E.DEPARTAMENTO = D.NUMERO;
```

Determine o nome do cargo e do departamento de um empregado.

```
SELECT E.SOBRENOME, C.NOME, D.NOME  
FROM EMPREGADO E  
INNER JOIN CARGO C  
ON E.CARGO = C.NUMERO  
INNER JOIN DEPARTAMENTO D  
ON E.DEPARTAMENTO = D.NUMERO;
```

### 6.1.33 Não-equijunção

São junções que não usam condições de igualdade.

#### 6.1.33.1 Práticas

Exibir o Número, Sobrenome, nome do cargos de cada empregados que podem ser galgados na empresa.

```
SELECT E.NUMERO, E.SOBRENOME, C.NOME  
FROM EMPREGADO E  
INNER JOIN CARGO C  
ON C.NUMERO > E.CARGO;
```

### 6.1.34 Junção Externa

Quando se deseja exibir dados que não possuem uma relação comum entre tabelas.

#### 6.1.34.1 Dicas

Em banco como Oracle pode-se utilizar a extensão (+) e em SQL Server (\*);

Só pode ser utilizado de um lado da expressão;

Não se pode utilizar o operador IN e nem o OR.

#### 6.1.34.2 Práticas

Exibir o Número, Sobrenome, Apelido, nome do cargo de todos os empregados, mesmo daqueles que ainda não possuem cargos atribuídos.

```
SELECT E.NUMERO, E.SOBRENOME, E.APELIDO, C.NOME  
FROM EMPREGADO E  
RIGHT JOIN CARGO C  
ON C.NUMERO = E.CARGO;
```

### 6.1.35 Autojunção

Quando se une uma tabela a ela mesma.

#### 6.1.35.1 Práticas

Exibir o Número, Primeironome dos empregados e o Número e o Primeironome dos seus respectivos gerentes.

```
SELECT E.NUMERO, E.PRIMEIRONOME, G.NUMERO, G.PRIMEIRONOME  
FROM EMPREGADO E LEFT JOIN EMPREGADO G  
ON E.GERENTE = G.NUMERO;
```

### 6.1.36 Funções de Grupo

Operam um conjunto de linhas, as funções de grupo operam em conjuntos de linhas para fornecer um resultado por grupo. Esses conjuntos podem ser a tabela inteira ou dividida em grupos.

#### 6.1.36.1 Dicas

Funções de grupo são mais conhecidas como funções agregadas: AVG, COUNT, MAX, MIN e SUM; ALL faz com que a função considere todos os valores, inclusive as duplicidades; DISTINCT faz com que a função considere somente valores não duplicados; Todas as funções de grupos ignoram valores nulos, para garantir a aplicabilidade para todos os registros use (\*).

#### 6.1.36.2 Práticas

Exibir a média de salários pago pelo empregador.

```
SELECT AVG(SALARIO) FROM EMPREGADO;
```

Exibir o somatório dos salários.

```
SELECT SUM(SALARIO) FROM EMPREGADO;
```

Exibir a quantidade de empregados.

```
SELECT COUNT(SALARIO) FROM EMPREGADO;
```

Exibir o maior salário.

```
SELECT MAX(SALARIO) FROM EMPREGADO;
```

Exibir o menor salário.

```
SELECT MIN(SALARIO) FROM EMPREGADO;
```

### 6.1.37 Grupos de Dados

Usa-se o GROUP BY para dividir as linhas de uma tabela em grupos.

#### 6.1.37.1 Práticas

Exibir a média de salários pagos por Departamento.

```
SELECT D.NOME, AVG(E.SALARIO) AS MEDIAS  
FROM DEPARTAMENTO D INNER JOIN EMPREGADO E  
ON D.NUMERO = E.DEPARTAMENTO  
GROUP BY D.NOME;
```

Exibir o somatório dos salários pagos por Cargo.

```
SELECT C.NOME, SUM(E.SALARIO) AS SOMATORIO  
FROM CARGO C INNER JOIN EMPREGADO E  
ON C.NUMERO = E.CARGO  
GROUP BY C.NOME;
```

Exibir a quantidade de empregados por gestor.

```
SELECT G.SOBRENOME, COUNT(E.NUMERO) AS QTDE
```

```
FROM EMPREGADO G INNER JOIN EMPREGADO E
ON G.NUMERO = E.GERENTE
GROUP BY G.SOBRENOME;
```

Exibir o maior salário por gestor.

```
SELECT G.SOBRENOME, MAX(E.SALARIO) AS MAIORSALARIO
FROM EMPREGADO G INNER JOIN EMPREGADO E
ON G.NUMERO = E.GERENTE
GROUP BY G.SOBRENOME;
```

Exibir o menor salário por gestor.

```
SELECT G.SOBRENOME, MIN(E.SALARIO) AS MENORSALARIO
FROM EMPREGADO G INNER JOIN EMPREGADO E
ON G.NUMERO = E.GERENTE
GROUP BY G.SOBRENOME;
```

### 6.1.38 Grupo de Dados com Filtro

Usa-se o HAVING para filtrar as linhas de uma tabela em grupos.

#### 6.1.38.1 Práticas

Exibir a média de salários pagos por Departamento cujo a média seja maior ou igual a 2000.

```
SELECT D.NOME, AVG(E.SALARIO) AS MEDIAS
FROM DEPARTAMENTO D INNER JOIN EMPREGADO E
ON D.NUMERO = E.DEPARTAMENTO
GROUP BY D.NOME
HAVING AVG(E.SALARIO) >= 2000;
```

Exibir o somatório dos salários pagos por Cargo cujo o somatório seja menor que 1500.

```
SELECT D.NOME, SUM(E.SALARIO) AS MEDIAS
FROM DEPARTAMENTO D INNER JOIN EMPREGADO E
ON D.NUMERO = E.DEPARTAMENTO
GROUP BY D.NOME
HAVING AVG(E.SALARIO) < 1500;
```

### 6.1.39 ROLLUP

Usa-se o ROLLUP para produzir linhas superagregadas pela referência cruzada de colunas. O agrupamento ROLLUP produz um conjunto de resultados que contém as linhas agrupadas normais e os valores de subtotais.

#### 6.1.39.1 Práticas

Exibir o somatório de Salário agrupado por Departamento e Cargo dos empregados com Número de Departamento menor que 60, exibindo os subtotais de Departamento e Cargo.

```
SELECT CAST(DEPARTAMENTO AS CHAR), CAST(CARGO AS CHAR), SUM(SALARIO)
FROM EMPREGADO
WHERE DEPARTAMENTO < 60
GROUP BY DEPARTAMENTO, CARGO
UNION
SELECT "", 'TOTAL', COALESCE(SUM(SALARIO), 0)
FROM EMPREGADO;
```

```
SELECT DEPARTAMENTO, CARGO, SUM(SALARIO)
FROM EMPREGADO
```

```
WHERE DEPARTAMENTO < 60  
GROUP BY ROLLUP (DEPARTAMENTO, CARGO);
```

#### **6.1.40 CUBE**

Usa-se o CUBE para produzir um conjunto de resultados que contém as linhas de ROLLUP e as linhas de tabulação cruzada. Pode-se usá-lo para produzir valores de tabulação cruzada com uma instrução SELECT simples.

##### **6.1.40.1 Práticas**

Exibir o somatório de Salário agrupado por Departamento e Cargo dos empregados com Número de Departamento menor que 60, exibindo a referência cruzada de Departamento e Cargo.

```
SELECT DEPARTAMENTO, CARGO, SUM(SALARIO)  
FROM EMPREGADO  
WHERE DEPARTAMENTO < 60  
GROUP BY CUBE (DEPARTAMENTO, CARGO);
```

#### **6.1.41 Ordenação de Dados**

Usa-se o ORDER BY para classificar as linhas de uma seleção.

##### **6.1.41.1 Práticas**

Exibir a média de salários pagos por Departamento em ordem decrescente.

```
SELECT D.NOME, AVG(E.SALARIO) AS MEDIAS  
FROM DEPARTAMENTO D INNER JOIN EMPREGADO E  
ON D.NUMERO = E.DEPARTAMENTO  
GROUP BY D.NOME  
ORDER BY AVG(E.SALARIO) DESC;
```

Exibir o somatório dos salários pagos por Cargo em ordem ascendente.

```
SELECT C.NOME, SUM(E.SALARIO) AS MEDIAS  
FROM CARGO C INNER JOIN EMPREGADO E  
ON C.NUMERO = E.CARGO  
GROUP BY C.NOME  
ORDER BY 1 ASC;
```

```
SELECT D.NOME, AVG(E.SALARIO) AS MEDIAS  
FROM DEPARTAMENTO D INNER JOIN EMPREGADO E  
ON D.NUMERO = E.DEPARTAMENTO  
GROUP BY D.NOME  
ORDER BY AVG(E.SALARIO) DESC;
```

```
SELECT C.NOME, SUM(E.SALARIO) AS MEDIAS  
FROM CARGO C INNER JOIN EMPREGADO E  
ON C.NUMERO = E.CARGO  
GROUP BY C.NOME  
ORDER BY 1 ASC;
```

#### **6.1.42 Aninhamento de Grupo de Dados**

As funções de grupo podem ser aninhadas até uma profundidade de dois.

##### **6.1.42.1 Práticas**

Exibir o salário médio máximo pago por Departamento.

```
SELECT MAX(AVG(E.SALARIO)) AS MEDIAS  
FROM EMPREGADO E
```

GROUP BY E.DEPARTAMENTO;

### **6.1.43 Subconsulta**

Consiste em executar uma consulta (externa) a partir de outra consulta (interna).

#### **6.1.43.1 Dicas**

Existem operadores que retornam apenas uma linha (=, <>, >, >=, <, <=) e o que retorna várias linhas (IN);

Coloque subconsultas entre parênteses;

A quantidade de níveis de subconsultas dependem do banco de dados.

#### **6.1.43.2 Práticas**

Quais os empregados possuem salário maior que o de Pedroca?

```
SELECT APELIDO, SALARIO
FROM EMPREGADO
WHERE SALARIO >
(SELECT SALARIO FROM EMPREGADO WHERE APELIDO='PEDROCA');
```

Qual o nome (Primeironome + Sobrenome) dos empregados possuem o menor salário?

```
SELECT PRIMEIRONOME + ' ' + SOBRENOME AS NOME, SALARIO FROM EMPREGADO
WHERE SALARIO = (SELECT MIN(SALARIO) FROM EMPREGADO);
```

Exibir o menor Salário por Departamento onde o menor Salário seja o maior que o menor Salário do Departamento de código 50.

```
SELECT D.NOME, MIN(E.SALARIO)
FROM EMPREGADO E
INNER JOIN DEPARTAMENTO D
ON E.DEPARTAMENTO = D.NUMERO
GROUP BY D.NOME
HAVING MIN(E.SALARIO) >
(SELECT MIN(SALARIO) FROM EMPREGADO WHERE DEPARTAMENTO = 50);
```

O que há de errado com a instrução abaixo? A subconsulta retornou mais de 1 valor. Isso não é permitido quando a subconsulta segue um =, !=, <, <=, >, >= ou quando ela é usada como uma expressão.

```
SELECT Numero, Sobrenome
FROM Empregado
WHERE Salario = (SELECT MIN(Salario)
FROM Empregado
GROUP BY Departamento);
```

### **6.1.44 ANY**

Seleciona algum conjunto de dados.

#### **6.1.44.1 Práticas**

Exibir Número, Sobrenome, Salário dos empregados que possuírem algum salário menor que o menor salário do departamento de código 50.

```
SELECT NUMERO, SOBRENOME, SALARIO
FROM EMPREGADO
WHERE SALARIO < ANY
(SELECT MIN(SALARIO) FROM EMPREGADO WHERE DEPARTAMENTO = 50);
```

### 6.1.45 ALL

Seleciona todos os dados.

#### 6.1.45.1 Práticas

Exibir Número, Sobrenome, Salário dos todos os empregados que possuírem salário menor que o menor salário do departamento de código 50.

```
SELECT NUMERO, SOBRENOME, SALARIO
FROM EMPREGADO
WHERE SALARIO < ALL
(SELECT MIN(SALARIO) FROM EMPREGADO WHERE DEPARTAMENTO = 50);
```

### 6.1.46 Correlacionadas

Ocorre quando uma subconsulta faz referência a uma coluna de uma tabela referenciada na instrução mãe. Uma subconsulta correlacionada é avaliada uma vez para cada linha processada pela instrução mãe. A instrução mãe pode ser uma instrução SELECT, UPDATE ou DELETE.

#### 6.1.46.1 Práticas

Exibir todos os empregados (Sobrenome, Salário e Número do Departamento) que ganham acima do salário médio de seus respectivos departamentos.

```
SELECT EXT.SOBRENOME, EXT.SALARIO, EXT.DEPARTAMENTO
FROM EMPREGADO EXT
WHERE EXT.SALARIO >
(SELECT AVG(ITN.SALARIO)
FROM EMPREGADO ITN
WHERE ITN.DEPARTAMENTO = EXT.DEPARTAMENTO);
```

### 6.1.47 EXISTS

O operador EXISTS testa a existência de linhas no conjunto de resultados da subconsulta. Se o valor da linha de uma subconsulta for encontrado: A pesquisa na consulta interna não continuará; e A condição será indicada por um flag TRUE. Se o valor de linha de uma subconsulta não for encontrado: A condição será indicada por um flag FALSE; A pesquisa continuará na consulta interna.

#### Práticas

Exibir os empregados (Sobrenome, Salário e Número do Departamento) com no mínimo uma pessoa subordinada a eles.

```
SELECT EXT.SOBRENOME, EXT.SALARIO, EXT.DEPARTAMENTO
FROM EMPREGADO EXT
WHERE EXISTS
(SELECT 'X'
FROM EMPREGADO ITN
WHERE ITN.GERENTE = EXT.NUMERO);
```

### 6.1.48 União

Retorna todas as linhas selecionadas por uma consulta. Use esse operador para retornar todas as linhas de várias tabelas e eliminar as linhas duplicadas.

#### 6.1.48.1 Dicas

Necessariamente os campos das tabelas unidas devem possuir os mesmos tipos de dados;  
Os campos devem estar posicionados na mesma ordem;  
A quantidade de campos deve ser a mesma;  
A cláusula ORDER BY deve ser utilizada no final da união.

#### 6.1.48.2 Práticas

Exibir a união das tabelas Cargo e Departamento para os campos Número e Nome.

```
SELECT NUMERO, NOME  
FROM CARGO  
UNION  
SELECT NUMERO, NOME  
FROM DEPARTAMENTO;
```

#### **6.1.49 Intersecção**

Serve para retornar todas as linhas comuns a várias consultas.

##### **6.1.49.1 Práticas**

Exibir a intersecção das tabelas Cargo e Departamento para os campos Número e Nome.

```
SELECT NUMERO, NOME FROM CARGO INTERSECT SELECT NUMERO, NOME FROM  
DEPARTAMENTO;
```

#### **6.1.50 Subtração**

Serve para obter linhas retornadas pela primeira consulta e ausentes na segunda consulta.

##### **6.1.50.1 Práticas**

Exibir a subtração das tabelas Cargo e Departamento para os campos Número e Nome.

```
SELECT NUMERO, NOME FROM CARGO MINUS SELECT NUMERO, NOME FROM  
DEPARTAMENTO;
```

#### **6.1.51 Macrossubstituição**

Alguns bancos permitem passagem de parâmetros com o uso de variáveis de substituição.

##### **6.1.51.1 Dicas**

No Oracle a macrossubstituição é com (&);

No SQLServer a macrossubstituição é com (:);

##### **6.1.51.2 Práticas**

Exibir o Sobrenome do empregado que possuir o Número informado pelo usuário.

```
SELECT SOBRENOME  
FROM EMPREGADO  
WHERE NUMERO = :Informe_Numero;
```

#### **6.1.52 Seleção de Seleção**

Alguns BD permitem a seleção de dados nas cláusulas SELECT e/ou FROM.

```
SELECT E.NUMERO, E.SOBRENOME, E.SALARIO, E.DEPARTAMENTO AS DEPCOD, (SELECT  
D.NOME FROM DEPARTAMENTO D WHERE D.NUMERO = E.DEPARTAMENTO) AS DEPNAME  
FROM EMPREGADO E;
```

#### **6.1.53 Análise Top-N**

Alguns BD permitem a seleção das N primeiras linhas que satisfazem a uma determinada seleção.

##### **6.1.53.1 Práticas**

Exibir os 2 primeiros registros de empregados cujos os salários sejam menores que 2000.

```
SELECT TOP 2 NUMERO, APELIDO, SALARIO FROM EMPREGADO  
WHERE SALARIO < 2000;
```

#### 6.1.54 Excluir Registros

Excluir registros.

##### 6.1.54.1 Sintaxe

```
DELETE FROM nome  
[WHERE definição]  
[ORDER BY ...];
```

#### 6.1.55 Excluir e recriar tabela:

Apaga e recria uma tabela.

##### 6.1.55.1 Sintaxe

```
TRUNCATE TABLE nome_tabela;
```

#### 6.1.56 Excluir Tabela

Apaga uma tabela.

##### 6.1.56.1 Sintaxe

```
DROP TABLE nome_tabela;
```

#### 6.1.57 Transações

Tratamento de dados de forma atômica.

##### 6.1.57.1 Dicas

Em alguns BD as instruções DDL e DCL são automáticas.

##### 6.1.57.2 Práticas

```
Exibir os dados da tabela Empregado;  
Salvar um ponto A;  
Apagar todos os empregados do Departamento 50;  
Exibir os dados da tabela Empregado;  
Salvar um ponto B;  
Apagar todos os empregados com Salário = 1000;  
Exibir os dados da tabela Empregado;  
Restaurar o ponto A;  
Exibir os dados da tabela Empregado;  
Restaurar o ponto B;  
Exibir os dados da tabela Empregado.
```

```
SAVE TRANSACTION A;  
SELECT * FROM DEPARTAMENTO;  
DELETE FROM DEPARTAMENTO WHERE NUMERO = 110;  
SELECT * FROM DEPARTAMENTO;  
UPDATE DEPARTAMENTO SET NOME = 'TESTE' WHERE NUMERO = 90;  
SELECT * FROM DEPARTAMENTO;  
ROLLBACK TRANSACTION A;  
SAVE TRANSACTION B;  
INSERT INTO DEPARTAMENTO (NUMERO, NOME) VALUES (200, 'TESTE2');  
SELECT * FROM DEPARTAMENTO;  
COMMIT TRANSACTION B;  
SELECT * FROM DEPARTAMENTO;
```

#### 6.1.58 Visão

Serve para apresentar combinações ou subconjuntos lógicos de dados através de views de tabelas. A view é uma tabela lógica baseada em uma tabela ou em outra view. Ela não contém dados próprios, mas é uma janela por meio da qual os dados das tabelas podem ser vistos e algumas

vezes alterados. As tabelas nas quais uma view se baseia são denominadas de tabelas-base. A view é armazenada como uma instrução SELECT no dicionário de dados.

### 6.1.58.1 Sintaxe

```
CREATE VIEW [ schema_name . ] view_name  
[ (column [ ,...n ] ) ]  
AS select_statement;
```

### 6.1.58.2 Dicas

- Serve para restringir o acesso a dados;
- Serve para facilitar as consultas complexas;
- Serve para permitir a independência dos dados;
- Serve para apresentar diferentes visões dos mesmos dados.

### 6.1.58.3 View Simples

- Cria dados a partir de uma tabela apenas;
- Não contém funções e nem grupo de dados;
- Permite operações DML.

### 6.1.58.4 Práticas

Criar uma visão denominada VISA0\_1, que contenha os campos Número, Primeironome, Sobrenome dos empregados do departamento 80. Depois de criada insira um novo registro nessa visão. Finalizando exiba todos os dados da tabela Empregado.

```
CREATE VIEW VISA0_1 AS  
SELECT NUMERO, PRIMEIRONOME, SOBRENOME FROM EMPREGADO  
WHERE DEPARTAMENTO=80;
```

```
INSERT INTO VISA0_1 VALUES (200, 'ANTONIETA', 'LARANJEIRA');
```

```
SELECT * FROM EMPREGADO;
```

### 6.1.58.5 View Complexas

- Cria dados a partir de várias tabelas;
- Contém funções ou grupo de dados;
- Nem sempre permite operações DML.

### 6.1.58.6 Práticas

Criar uma visão denominada VISA0\_2, que contenha os campos Nome (Primeironome + ' ' + Sobrenome) e o campo Salariobase (Salário – 500) dos empregados do departamento 30. Depois de criada tente inserir um novo registro nessa visão. Finalizando exiba todos os dados da tabela Empregado.

```
CREATE VIEW VISA0_2 AS  
SELECT (PRIMEIRONOME + ' ' + SOBRENOME) AS NOME, (SALARIO - 500) AS SALARIOBASE  
FROM EMPREGADO WHERE DEPARTAMENTO=10;
```

```
INSERT INTO VISA0_2 VALUES ('JULIETA PEREIRA', 400);
```

```
SELECT * FROM EMPREGADO;
```

## 6.1.59 Controle de Usuários

Criar novos usuários para o BD.

### 6.1.59.1 Sintaxe

```
CREATE LOGIN login  
WITH PASSWORD = 'senha';  
CREATE USER usuario
```

<http://www.alvarofpinheiro.eti.br>

```
FOR LOGIN [operador]  
WITH DEFAULT_SCHEMA = [dbo];  
Oracle  
CREATE USER usuario  
IDENTIFIED BY 'senha';
```

### 6.1.59.2 Práticas

Para adicionar um usuário ao banco de dados do SQL Server você tem que seguir três passos: Primeiro: você deve criar um login, que é um "cara" que tem permissão de se logar no SQL Server

```
CREATE LOGIN USUARIO WITH PASSWORD = 'senha';
```

Segundo: você deve criar um usuário para o banco de dados que deseja mapeando esse usuário para o login criado, assim seu usuário conseguirá se logar no SQL Server e entrar no banco de dados desejado.

```
CREATE USER USUARIO FROM LOGIN USUARIO;
```

Terceiro: você deve dar ou remover permissões ao usuário porque até o segundo passo o usuário criado só tem direito a entrar no banco de dados, dando as permissões o usuário já pode operar no banco de dados.

Se o usuário for comum você pode adicioná-lo apenas as roles de db\_reader e db\_writer, que permitirá que o usuário faça select, insert, delete e update em todas as tabelas do referido banco de dados.

```
EXEC SP_ADDROLEMEMBER 'DB_DATAREADER', 'USUARIO'
```

### 6.1.60 Privilégios

Fornecer acesso aos objetos dos BD.

#### 6.1.60.1 Sintaxe

```
GRANT privilégio ON objeto TO usuário;  
REVOKE privilégio ON objeto FROM usuário;
```

#### 6.1.60.2 Práticas

Fornecer acesso aos objetos dos BD.

```
GRANT CREATE TABLE, CREATE VIEW TO operador1;  
GRANT SELECT ON EMPREGADO TO operador1;
```

Retirar acesso aos objetos dos BD.

```
REVOKE CREATE TABLE, CREATE VIEW FROM operador1;  
REVOKE SELECT ON EMPREGADO FROM operador1;
```

### 6.1.61 Sequências

Alguns bancos como o Postgre e o Oracle possuem um objeto chamado de sequência que equivale no SQLServer a regra de IDENTITY( 1, 1).

#### 6.1.61.1 Sintaxe

```
CREATE [TEMPORARY | TEMP] SEQUENCE name  
[INCREMENT [BY] increment ]  
[MINVALUE minvalue | NO MINVALUE]  
[MAXVALUE maxvalue | NO MAXVALUE]  
[START [ WITH ] start]  
[CACHE cache]
```

[[ NO ] CYCLE]

### 6.1.62 Vínculos

A grande vantagem dos vínculos de banco de dados é que eles permitem aos usuários acessarem dados de um banco de dados remoto de modo que eles fiquem conectados por meio do conjunto de privilégios do proprietário do objeto. Em outras palavras, um usuário local pode acessar um banco de dados remoto sem necessariamente ser um usuário do banco de dados remoto.

#### 6.1.62.1 Práticas

Vincular dados.

```
CREATE PUBLIC DATABASE LINK banco  
USING 'usuário';  
SELECT * FROM objeto@banco;
```

## 6.2 Objetos de Bancos de Dados

Postgre:

- Tables
- Views
- Functions
- Domains
- Rules
- Triggers
- Indices
- Sequences
- Composite Types
- Enum Types
- Base Types
- Aggregates
- Operators

## 6.3 Principais Funções

ASCII(*String*)  
pega o valor em ASCII da *String*

CHAR(integer)  
troca inteiro do ASCII em um caractere

LEN(*String*)  
Identifica o comprimento de uma expressão em caracteres

LOWER(*String*)  
converte uma *String* uppercase para lowercase.

LTRIM(*String*)  
remove os espaços em branco

PATINDEX(posicao, expressao)  
devolve a posição de uma *String* dentro de um texto. Se não encontrar, retorna zero.

REPLICATE(*String*, integer)

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

repete N vezes um caractere especificado

REVERSE(*String*)

retorna o inverso de uma expressão

RTRIM(*String*)

remove os espaços em branco à direita de uma *String*

SPACE(integer)

que retorna o número de espaços em branco informados no parâmetro

STUFF(*String* texto, X, Y, *String* texto\_a\_inserir)

apaga da *String* "texto" os y caracteres a partir da posição x e os substitui por "texto\_a\_inserir"

SUBSTRING(*String* texto, posicao\_inicial, tamanho)

retorna uma *String* com o comprimento definido em "tamanho" extraída da *String* "texto", a partir da "posicao\_inicial"

UPPER(*String*)

retorna *String* em maiúsculas

DATEADD (parte, numero, data)

adiciona um valor a parte de uma data

DATEDIFF (parte, data inicial, data final)

subtrai a data inicial da data final, indicando o resultado na unidade definida em "parte"

GETDATE()

retorna a data atual do sistema

DATENAME (parte, data)

retorna o nome da parte de uma data

DATEPART(parte, data)

retorna a parte de uma data

CAST(expressao as datatype)

converte uma expressão no datatype informado

COL\_LENGTH(nome\_da\_tabela, nome\_da\_coluna)

retorna o tamanho da coluna

COL\_NAME(id\_da\_tabela, id\_da\_coluna)

retorna o nome da coluna

DATALENGTH(expressao)

retorna o número de bytes usados para armazenar a expressão

DB\_ID(nome\_do\_banco)

retorna o ID do banco informado

DB\_NAME(id\_do\_banco)

retorna o nome do banco

HOST\_ID()

retorna a ID da estação que está acessando o SQL Server

HOST\_NAME()

<http://www.alvarofpinheiro.eti.br>

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

retorna o nome da estação que está acessando o SQL Server

IDENT\_INCR(nome\_da\_tabela\_ou\_view)

retorna o valor incrementado

IDENT\_SEED(tabela\_ou\_view)

retorna o valor inicial da coluna

INDEX\_COL(nome\_da\_tabela, indice\_id, chave\_id)

retorna o nome da coluna que participa do índice

ISNULL(expressao, valor)

se a expressao for null, troca pelo valor especificado

ISNUMERIC(expressao)

retorna 1 se a expressao for numerica e 0 se não for

NEWID()

retorna um novo valor do tipo uniqueidentifier

NULLIF(expressao\_1, expressao\_2)

retorna nulo se as duas expressoes forem equivalentes. Se não forem, retorna a primeira expressao.

OBJECT\_ID(nome\_do\_objeto)

retorna o ID de um objeto, a partir do nome fornecido

OBJECT\_NAME(ID\_do\_objeto)

retorna o nome do objeto, a partir do ID fornecido

PARSENAME(objeto, parte)

retorna a parte do nome de um objeto, desde que tenha sido qualificado

STATS\_DATE(tabela\_id, indice\_id)

retorna a data em que as estatísticas do índice foram atualizadas

SUSER\_SID(nome\_do\_usuario)

retorna o ID do usuario informado

SUSER\_NAME(usuario\_id)

retorna o id do usuário no servidor. O argumento é opcional.

SUSER\_SNAME(id\_do\_usuario)

retorna o nome do usuario informado. Se nenhum ID de usuario for passado para a função, retorna o nome do usuario logado

USER\_ID(nome\_do\_usuario)

retorna o ID do usuario informado para o BD em uso

USER\_NAME(id\_do\_usuario)

retorna o usuario conectado ao BD

ABS(numero)

retorna o valor absoluto do numero

ACOS(float)

retorna o arco-coseno do numero informado

ASIN(float)

<http://www.alvarofpinheiro.eti.br>

retorna o arco-seno do numero informado

ATAN(float)

retorna o arco-tangente do numero informado

ATN2(Float expressao\_1, float expressao\_2)

arco-tangente do valor definido pela divisão da primeira expressão pela segunda

CEILING(numero)

retorna o menor inteiro que seja maior ou igual ao numero informado

COS(float)

retorna o cosseno do numero informado

COT(float)

retorna o cotangente do numero informado

DEGREES(numero)

converte radianos para graus

EXP(float)

retorna o exponencial de um numero especificado

FLOOR(numero)

retorna o maior inteiro que seja menor ou igual ao numero informado

LOG(float)

retorna o logaritmo natural do numero informado

LOG10(float)

retorna o logaritmo base 10 do numero informado

PI()

retorna o valor de PI 3.1415926535897931.

POWER(numero, potencia)

retorna o valor elevado à potencia informada

RADIANS(numero)

converte graus para radianos

RAND(expressao)

um número aleatório entre 0 e 1. Expressão é opcional e será usada como semente da cadeia pseudo-aleatória

ROUND(numero, precisao, arredonda\_ou\_trancar)

arredonda ou trunca o numero fornecido de acordo com a precisao informada. Se o terceiro parametro não for passado para a função, o numero é arredondado. Se quiser que o numero seja truncado, deve-se fornecer o valor 1

SIGN(numero)

retorna sinal positivo, negativo ou zero do numero

SIN(float)

retorna o seno do angulo especificado

SQRT(float)

retorna a raiz quadrada de um numero

**TAN(float)**

retorna a tangente de um numero informado

**SQUARE(float)**

retorna o quadrado de um numero

**Principais Funções do Oracle**

**ABS(n)**

Devolve o valor absoluto de (n).

**CEIL(n)**

Obtém o valor inteiro imediatamente superior ou igual a "n".

**FLOOR(n)**

Devolve o valor inteiro imediatamente inferior ou igual a "n".

**MOD (m, n)**

Devolve o resto resultante de dividir "m" entre "n".

**NVL (valor, expressão)**

Substitui um valor nulo por outro valor.

**POWER (m, expoente)**

Calcula a potência de um número.

**ROUND (numero [, m])**

Arredonda números com o número de dígitos de precisão indicados.

**SIGN (valor)**

Indica o signo do "valor".

**SQRT(n)**

Devolve a raiz quadrada de "n".

**TRUNC (numero, [m])**

Trunca números para que tenham uma certa quantidade de dígitos de precisão.

**AVG (valor)**

Devolve a média de um conjunto de valores.

**CHR(n)**

Devolve o caractere cujo valor em binário é equivalente a "n".

**CONCAT (cad1, cad2)**

Devolve "cad1" concatenada com "cad2".

**LOWER (cad)**

Devolve a cadeia "cad" em minúsculas.

**UPPER (cad)**

Devolve a cadeia "cad" em maiúsculas.

**INITCAP (cad)**

Converte a cadeia "cad" a tipo título.

**LPAD (cad1, n[,cad2])**

Adiciona caracteres à esquerda da cadeia até que tenha uma certa longitude.

**RPAD** (cad1, n[,cad2])

Adiciona caracteres à direita até que tenha uma certa longitude.

**LTRIM** (cad [,set])

Suprime um conjunto de caracteres à esquerda da cadeia.

**RTRIM** (cad [,set])

Suprime um conjunto de caracteres à direita da cadeia.

**REPLACE** (cad, cadeia\_busca [, cadeia\_substitucao])

Substitui um caractere ou caracteres de uma cadeia com 0 ou mais caracteres.

**SUBSTR** (cad, m [,n])

Obtém parte de uma cadeia.

**TRANSLATE** (cad1, cad2, cad3)

Converte caracteres de uma cadeia em caracteres diferentes, segundo um plano de substituição marcado pelo usuário.

**ASCII**(cad)

Devolve o valor ASCII da primeira letra da cadeia "cad".

**INSTR** (cad1, cad2 [, comeco [,m]])

Permite uma busca de um conjunto de caracteres em uma cadeia, mas não suprime nenhum caractere depois.

**LENGTH** (cad)

Devolve o número de caracteres de cad.

**SYSDATE**

Devolve a data do sistema.

**ADD\_MONTHS** (data, n)

Devolve a data "data" incrementada em "n" meses.

**LASTDAY** (data)

Devolve a data do último dia do mês que contém "data".

**MONTHS\_BETWEEN** (data1, data2)

Devolve a diferença em meses entre as datas "data1" e "data2".

**NEXT\_DAY** (data, cad)

Devolve a data do primeiro dia da semana indicado por "cad" depois da data indicada por "data".

**TO\_CHAR**

Transforma um tipo DATE ou NUMBER em uma cadeia de caracteres.

**TO\_DATE**

Transforma um tipo NUMBER ou CHAR em DATE.

**TO\_NUMBER**

Transforma uma cadeia de caracteres em NUMBER.

### 6.4 Tipos de Dados

- **BIGINT**: valores numéricos inteiros.

- BIT: bits ou seja somente poderá conter os valores lógicos 0 ou 1.
- CHAR(N): N caracteres fixos no formato não Unicode, com espaços em branco.
- DATE: somente data.
- DATETIME: data e hora, com precisão de centésimos de segundos.
- DECIMAL(18,0): mesmas funcionalidades do tipo NUMERIC.
- FLOAT: valores numéricos aproximados com precisão de ponto flutuante.
- IMAGE: dados no formato binário.
- INT: valores numéricos inteiros.
- NCHAR(N): N caracteres fixos no formato Unicode, com espaços em branco.
- NTEXT: caracteres no formato Unicode.
- NUMERIC(18,0): valores numéricos com casas decimais, utilizando precisão.
- NVARCHAR(N): Armazena N caracteres no formato Unicode.
- REAL: valores numéricos aproximados com precisão de ponto flutuante.
- SMALLDATETIME: data e hora, com precisão de minutos.
- SMALLINT: valores numéricos inteiros.
- TEXT: caracteres no formato não Unicode.
- TIME: somente hora. Pode armazenar segundos até a fração de 9999999
- TINYINT: valores numéricos inteiros.
- VARCHAR(N): N caracteres (até 8.000) no formato não Unicode.

### 6.5 Usuários e Senhas Padrões

SQLServer: Usuário sa; Senha definida.

### 6.6 Exemplos

```
CREATE TABLE DISCIPLINA (COD INTEGER, DSC VARCHAR(50), PRIMARY KEY(COD));
CREATE TABLE PROFESSOR (MAT INTEGER, NOM VARCHAR(50), PRIMARY KEY(MAT));
CREATE TABLE ALUNO (MAT INTEGER, NOM VARCHAR(50), PRIMARY KEY(MAT));
CREATE TABLE NOTA (MAT INTEGER, DIS INTEGER, NT1 FLOAT, NT2 FLOAT, NT3 FLOAT,
PRIMARY KEY (MAT,DIS), FOREIGN KEY (MAT) REFERENCES ALUNO (MAT), FOREIGN KEY
(DIS) REFERENCES DISCIPLINA (COD) ON DELETE CASCADE ON UPDATE CASCADE);
DROP TABLE PROFESSOR;
CREATE TABLE TEMPOR (MED FLOAT);
```

```
CREATE USER 'U1'@'localhost' IDENTIFIED BY 'u1';
CREATE USER 'U2'@'localhost' IDENTIFIED BY 'u2';
CREATE USER 'U3'@'localhost' IDENTIFIED BY 'u3';
GRANT ALL PRIVILEGES ON *.* TO 'U1'@'localhost';
GRANT INSERT ON exemplo.disciplina TO 'U2'@'localhost';
GRANT UPDATE ON exemplo.disciplina TO 'U2'@'localhost';
GRANT SELECT ON exemplo.disciplina TO 'U2'@'localhost';
GRANT DELETE ON exemplo.disciplina TO 'U3'@'localhost';
REVOKE DELETE ON exemplo.disciplina TO 'U3'@'localhost'
FLUSH PRIVILEGES;
DROP USER 'U3'@'localhost';
```

```
START TRANSACTION;
SELECT @A:=AVG(NT1) FROM NOTA WHERE MAT=1;
UPDATE TEMPOR SET MED=@A;
COMMIT;
```

```
CREATE TRIGGER GATILHO BEFORE INSERT ON NOTA  
FOR EACH ROW UPDATE NT3 SET NT3 = (NT1 + NT2) / 2;
```

## **6.7 Exercícios**

*Obs: a solução foi realizada no SQLServer.*

```
--Tabela Cargos  
CREATE TABLE CARGO (  
  NUMERO INTEGER IDENTITY( 1, 1),  
  NOME VARCHAR(30) NOT NULL,  
  PRIMARY KEY (NUMERO)  
);  
INSERT INTO CARGO (NOME) VALUES ('ANALISTA TRAINEE');  
INSERT INTO CARGO (NOME) VALUES ('ANALISTA JUNIOR');  
INSERT INTO CARGO (NOME) VALUES ('ANALISTA PLENO');  
INSERT INTO CARGO (NOME) VALUES ('ANALISTA SÊNIOR');  
INSERT INTO CARGO (NOME) VALUES ('ANALISTA MASTER');  
INSERT INTO CARGO (NOME) VALUES ('CONSULTOR');  
SELECT * FROM CARGO;  
  
--Tabela Departamentos  
CREATE TABLE DEPARTAMENTO (  
  NUMERO INTEGER,  
  NOME VARCHAR(30) NOT NULL,  
  FATOR DECIMAL(3,2),  
  PRIMARY KEY (NUMERO)  
);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (10, 'ADMINISTRAÇÃO  
(ADMIN)', 1.75);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (20, 'MARKETING (MKT)',  
1.85);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (50, 'COMPRAS (COM)',  
2.73);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (60, 'TECNOLOGIA (TEC)',  
1.55);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (80, 'VENDAS (VEN)', 1.87);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (90, 'EXECUTIVO (EXEC)',  
1.92);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (110, 'CONTABILIDADE  
(CONTAB)', 2.35);  
INSERT INTO DEPARTAMENTO (NUMERO, NOME, FATOR) VALUES (190, 'CONTRATOS (CTT)',  
2.23);  
  
SELECT * FROM DEPARTAMENTO;  
  
--Tabela Empregados  
CREATE TABLE EMPREGADO (  
  NUMERO INTEGER,  
  PRIMEIRONOME VARCHAR(25),  
  SOBRENOME VARCHAR(25),  
  CARGO INTEGER,  
  DEPARTAMENTO INTEGER,  
  SALARIO REAL,  
  COMISSAO INTEGER,  
  NASCIMENTO DATE,  
  APELIDO VARCHAR(10),  
http://www.alvarofpinheiro.eti.br
```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
GERENTE INTEGER,  
PRIMARY KEY (NUMERO),  
FOREIGN KEY (CARGO) REFERENCES CARGO (NUMERO) ON UPDATE CASCADE,  
FOREIGN KEY (DEPARTAMENTO) REFERENCES DEPARTAMENTO (NUMERO) ON UPDATE  
CASCADE  
);
```

```
INSERT INTO EMPREGADO (NUMERO, PRIMEIRONOME, SOBRENOME) VALUES (100,  
'ANTÔNIO','PALM');  
INSERT INTO EMPREGADO (NUMERO, PRIMEIRONOME, SOBRENOME) VALUES (101, 'JOSÉ',  
'FIGUEIRA');  
INSERT INTO EMPREGADO (NUMERO, PRIMEIRONOME, SOBRENOME) VALUES (102, 'MARIA',  
'MANGUEIRA');  
INSERT INTO EMPREGADO (NUMERO, PRIMEIRONOME, SOBRENOME) VALUES (103, 'JOÃO',  
'ABACATEIRO');  
INSERT INTO EMPREGADO (NUMERO, PRIMEIRONOME, SOBRENOME) VALUES (104,  
'CARLOS', 'CAST');  
INSERT INTO EMPREGADO (NUMERO, PRIMEIRONOME, SOBRENOME) VALUES (107,  
'MÔNICA', 'VIDEIRA');  
INSERT INTO EMPREGADO (NUMERO, PRIMEIRONOME, SOBRENOME) VALUES (124,  
'PEDRO', 'LARANJA');  
INSERT INTO EMPREGADO (NUMERO, PRIMEIRONOME, SOBRENOME) VALUES (141, 'TIAGO',  
'PESSEGUE');  
INSERT INTO EMPREGADO (NUMERO, PRIMEIRONOME, SOBRENOME) VALUES (142,  
'RENATA', 'VIDEIRA');  
INSERT INTO EMPREGADO (NUMERO, PRIMEIRONOME, SOBRENOME) VALUES (143,  
'GABRIEL', 'MACIEIRA');
```

```
UPDATE EMPREGADO SET CARGO=7, DEPARTAMENTO=10, GERENTE=NULL,  
SALARIO=1000, COMISSAO=NULL, NASCIMENTO='01/07/80', APELIDO='TONHO' WHERE  
NUMERO=100;  
UPDATE EMPREGADO SET CARGO=7, DEPARTAMENTO=20, GERENTE=100, SALARIO=1500,  
COMISSAO=NULL, NASCIMENTO='18/05/70', APELIDO='ZÉ' WHERE NUMERO=101;  
UPDATE EMPREGADO SET CARGO=8, DEPARTAMENTO=10, GERENTE=100, SALARIO=2000,  
COMISSAO=3, NASCIMENTO='22/07/80', APELIDO='MA' WHERE NUMERO=102;  
UPDATE EMPREGADO SET CARGO=9, DEPARTAMENTO=50, GERENTE=100, SALARIO=1500,  
COMISSAO=2, NASCIMENTO='03/03/71', APELIDO='JO' WHERE NUMERO=103;  
UPDATE EMPREGADO SET CARGO=NULL, DEPARTAMENTO=60, GERENTE=100,  
SALARIO=2500, COMISSAO=3, NASCIMENTO='01/01/65', APELIDO='CA' WHERE NUMERO=104;  
UPDATE EMPREGADO SET CARGO=7, DEPARTAMENTO=90, GERENTE=NULL,  
SALARIO=1000, COMISSAO=2, NASCIMENTO='15/02/60', APELIDO='MO' WHERE  
NUMERO=107;  
UPDATE EMPREGADO SET CARGO=8, DEPARTAMENTO=80, GERENTE=107, SALARIO=1100,  
COMISSAO=NULL, NASCIMENTO='30/07/83', APELIDO='PEDROCA' WHERE NUMERO=124;  
UPDATE EMPREGADO SET CARGO=9, DEPARTAMENTO=110, GERENTE=107, SALARIO=2000,  
COMISSAO=2, NASCIMENTO='21/06/78', APELIDO='TI' WHERE NUMERO=141;  
UPDATE EMPREGADO SET CARGO=10, DEPARTAMENTO=80, GERENTE=107, SALARIO=1500,  
COMISSAO=NULL, NASCIMENTO='02/12/72', APELIDO='RE' WHERE NUMERO=142;  
UPDATE EMPREGADO SET CARGO=NULL, GERENTE=110, SALARIO=2000, COMISSAO=2,  
NASCIMENTO='11/11/81', APELIDO='BIEL' WHERE NUMERO=143;  
SELECT * FROM EMPREGADO;
```

--Questões:

--1.Mostre todos os dados dos empregados que nasceram após o ano de 1980.

```
SELECT * FROM EMPREGADO WHERE YEAR(NASCIMENTO) > 1980;
```

--2.Mostre o sobrenome, o cargo, o salário e a comissão dos empregados que recebem comissão.

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
SELECT SOBRENOME, CARGO, SALARIO, COMISSAO FROM EMPREGADO WHERE  
COMISSAO IS NOT NULL;
```

--3. Mostre os empregados que não tem comissão com um aumento de 10% em seus salários, arredondando-os.

```
SELECT NUMERO, APELIDO, COMISSAO, SALARIO, SALARIO + (SALARIO *.10) AS BRUTO  
FROM EMPREGADO WHERE COMISSAO IS NULL;
```

--4. Mostre os sobrenomes de todos os empregados juntamente com o número de anos e de meses completados de nascimento.

```
SELECT SOBRENOME, NASCIMENTO, GETDATE() AS DATACORRENTE,  
DATEDIFF(YEAR, NASCIMENTO, GETDATE()) AS ANOS,  
DATEDIFF(MONTH, NASCIMENTO, GETDATE()) AS MESES FROM EMPREGADO
```

--5. Mostre os empregados cujos os sobrenomes iniciem com J, K, L ou M.

```
SELECT SOBRENOME FROM EMPREGADO WHERE SOBRENOME LIKE 'J%' OR SOBRENOME  
LIKE 'K%' OR SOBRENOME LIKE 'L%' OR SOBRENOME LIKE 'M%';
```

--6. Mostre todos os funcionários e indique, através de "Sim" ou "Não", se eles recebem ou não uma comissão.

```
SELECT APELIDO, CASE WHEN COMISSAO IS NULL THEN 'NÃO' ELSE 'SIM' END AS  
COMISSAO FROM EMPREGADO;
```

--7. Mostre os nomes de departamentos, os nomes dos empregados, os nomes dos cargos e os salários dos empregados que trabalham nos departamentos 50, 60 ou 90.

```
SELECT E.PRIMEIRONOME AS EMPREGADO, D.NOME AS DEPARTAMENTO, C.NOME AS  
CARGO, E.SALARIO  
FROM EMPREGADO E  
INNER JOIN DEPARTAMENTO D ON E.DEPARTAMENTO = D.NUMERO  
INNER JOIN CARGO C ON E.CARGO = C.NUMERO  
WHERE E.DEPARTAMENTO IN (50, 60, 90);
```

--8. Quantos empregados têm um primeironome terminado em "o" e em "a"?

```
SELECT 'O' AS TERMINA, COUNT(NUMERO) AS QTDE FROM EMPREGADO WHERE  
PRIMEIRONOME LIKE '%O'  
UNION  
SELECT 'A' AS TERMINA, COUNT(NUMERO) AS QTDE FROM EMPREGADO WHERE  
PRIMEIRONOME LIKE '%A';
```

--9. Mostre o nome de todos os departamentos, além dos números dos empregados que trabalham em cada departamento. Certifique-se de que os departamentos sem empregados também sejam incluídos.

```
SELECT D.NUMERO,  
D.NOME,  
(SELECT COUNT(E.NUMERO) FROM EMPREGADO E WHERE E.DEPARTAMENTO =  
D.NUMERO) AS QTDE_EMP  
FROM DEPARTAMENTO D;
```

--10. Quais cargos são encontrados nos departamentos 10 e 50?

```
SELECT DISTINCT C.NOME FROM EMPREGADO E INNER JOIN CARGO C ON E.CARGO =  
C.NUMERO WHERE E.DEPARTAMENTO IN (10, 50);
```

--11. Quais cargos são encontrados nos departamentos administrativo e executivo, e quantos empregados desempenham esses cargos? Mostre primeiro o cargo com o maior número de empregados.

```
SELECT E.DEPARTAMENTO, C.NOME,
```

```
(SELECT COUNT(E.NUMERO) FROM EMPREGADO E WHERE E.CARGO = C.NUMERO) AS
QTDE_EMP
FROM EMPREGADO E
INNER JOIN CARGO C ON E.CARGO = C.NUMERO
WHERE E.DEPARTAMENTO IN (SELECT X.NUMERO FROM DEPARTAMENTO X WHERE
X.NOME LIKE 'ADM%' OR X.NOME LIKE 'EXE%');
```

--12. Mostre todos os empregados que nasceram na primeira metade do mês (antes do dia 16).  
SELECT E.APELIDO, E.NASCIMENTO FROM EMPREGADO E WHERE DAY(E.NASCIMENTO) < 16;

--13. Mostre o primeironome, o salário e o número de todos os empregados.  
SELECT E.PRIMEIRONOME, E.SALARIO, E.NUMERO FROM EMPREGADO E;

--14. Mostre todos os empregados que têm gerentes com um salário acima ou igual a 1000. Mostre os seguintes dados: primeironome do empregado e do gerente e salário do gerente.  
SELECT E.APELIDO, E.SALARIO, E.GERENTE FROM EMPREGADO E WHERE E.GERENTE IS NOT NULL  
AND EXISTS (SELECT 'X' FROM EMPREGADO E WHERE E.GERENTE IS NULL  
AND E.SALARIO >= 1000);

--15. Mostre o número do departamento, o nome, o número do empregado e o salário médio de todos os departamentos juntamente com os nomes, os salários e os cargos dos empregados que trabalham em cada departamento.

```
SELECT E.NUMERO,
       E.APELIDO,
       E.SALARIO,
       C.NOME AS CARGO,
       E.DEPARTAMENTO,
       D.NOME,
       (SELECT AVG(X.SALARIO) FROM EMPREGADO X WHERE
X.DEPARTAMENTO = E.DEPARTAMENTO) AS SAL_MED
FROM EMPREGADO E
INNER JOIN DEPARTAMENTO D ON E.DEPARTAMENTO = D.NUMERO
INNER JOIN CARGO C ON E.CARGO = C.NUMERO;
```

--16. Mostre o número do departamento e o menor salário e o maior salário.  
SELECT D.NUMERO,
 (SELECT MIN(E.SALARIO) FROM EMPREGADO E WHERE
E.DEPARTAMENTO = D.NUMERO) AS MENOR,
 (SELECT MAX(E.SALARIO) FROM EMPREGADO E WHERE
E.DEPARTAMENTO = D.NUMERO) AS MAIOR
FROM DEPARTAMENTO D;

--17. Mostre os números, os nomes dos departamentos em que não há empregados trabalhando.  
SELECT D.NUMERO, D.NOME FROM DEPARTAMENTO D WHERE D.NUMERO NOT IN (SELECT E.DEPARTAMENTO FROM EMPREGADO E);

--18. Mostre os números, os nomes dos departamentos e os números dos empregados trabalhando em cada departamento que inclua mais de 2 empregados, inclusive.  
SELECT D.NUMERO,
 D.NOME,
 (SELECT COUNT(E.NUMERO) FROM EMPREGADO E WHERE
E.DEPARTAMENTO = D.NUMERO) AS QTDE\_EMP
FROM DEPARTAMENTO D
WHERE (SELECT COUNT(E.NUMERO) FROM EMPREGADO E WHERE E.DEPARTAMENTO =
D.NUMERO) >= 2;

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

--19. Mostre os números, os nomes dos departamentos e os números dos empregados trabalhando em cada departamento que tenha o maior número de empregados.

```
SELECT TOP 1
    D.NUMERO,
    D.NOME,
    (SELECT COUNT(E.NUMERO) FROM EMPREGADO E WHERE E.DEPARTAMENTO =
D.NUMERO) AS QTDE_EMP
FROM DEPARTAMENTO D
ORDER BY 3 DESC;
```

--20. Mostre os números, os nomes dos departamentos e os números dos empregados trabalhando em cada departamento que tenha o menor número de empregados.

```
SELECT TOP 1
    D.NUMERO,
    D.NOME,
    (SELECT COUNT(E.NUMERO) FROM EMPREGADO E WHERE E.DEPARTAMENTO =
D.NUMERO) AS QTDE_EMP
FROM DEPARTAMENTO D
ORDER BY 3;
```

--21. Mostre o número do funcionário, o sobrenome, o salário, o número do departamento e o salário médio de seu departamento para todos os funcionários.

```
SELECT E.NUMERO,
    E.SOBRENOME,
    E.SALARIO,
    D.NOME AS DEPARTAMENTO,
    (SELECT AVG(X.SALARIO) FROM EMPREGADO X WHERE
X.DEPARTAMENTO = E.DEPARTAMENTO) AS SAL_MED
FROM EMPREGADO E
INNER JOIN DEPARTAMENTO D ON E.DEPARTAMENTO = D.NUMERO;
```

--22. Mostre todos os funcionários nascidos no dia da semana em que foi correspondente ao número de empregados.

```
SELECT E.APELIDO, DAY(E.NASCIMENTO) AS DIA FROM EMPREGADO E WHERE
DAY(E.NASCIMENTO) = (SELECT COUNT(X.NUMERO) FROM EMPREGADO X);
```

--23. Crie uma visão geral de aniversários com base na data de nascimento dos empregados. Classifique os aniversários em ordem crescente.

```
SELECT E.NUMERO, E.APELIDO, E.NASCIMENTO FROM EMPREGADO E ORDER BY
E.NASCIMENTO ASC;
```

--24. Crie uma consulta composta para produzir uma lista de empregados que mostre percentuais de aumento, número dos empregados.

--Considerando que os empregados dos departamentos 10, 50 e 110 receberão aumento de 5%,

--os do departamento 60 receberão um aumento de 10%,

--os dos departamentos 20 e 80 receberão um aumento de 15%

--e os do departamento 90 não receberão aumento.

```
SELECT E.NUMERO,
    E.APELIDO,
    E.SALARIO,
    E.DEPARTAMENTO,
    CASE
        WHEN E.DEPARTAMENTO IN (10,50,110) THEN E.SALARIO + (E.SALARIO * .5)
        WHEN E.DEPARTAMENTO IN (60) THEN E.SALARIO + (E.SALARIO * .10)
        WHEN E.DEPARTAMENTO IN (20,80) THEN E.SALARIO + (E.SALARIO * .15)
        WHEN E.DEPARTAMENTO IN (90) THEN E.SALARIO
    END AS AUMENTO
FROM EMPREGADO E;
```

--25.Exiba a data corrente.

```
SELECT TOP 1 GETDATE() FROM CARGO;
```

--26.Exiba o tempo em anos, mês e dias decorridos do nascimento até a data corrente.

```
SELECT  
    E.APELIDO,  
    E.NASCIMENTO,  
    DATEDIFF(YEAR, E.NASCIMENTO, GETDATE()) AS ANOS,  
    DATEDIFF(MONTH, E.NASCIMENTO, GETDATE()) AS MESES,  
    DATEDIFF(DAY, E.NASCIMENTO, GETDATE()) AS DIAS  
FROM EMPREGADO E
```

--27.Crie uma consulta para exibir os seguintes itens referentes aos departamentos:

--o número seja maior que 80 com o salário total de cada cargo em um departamento.

```
SELECT D.NUMERO, D.NOME AS DEPARTAMENTO, C.NOME AS CARGO, SUM(E.SALARIO) AS  
SAL_TOT  
FROM EMPREGADO E  
INNER JOIN DEPARTAMENTO D  
ON E.DEPARTAMENTO = D.NUMERO  
INNER JOIN CARGO C  
ON E.CARGO = C.NUMERO  
WHERE D.NUMERO > 80  
GROUP BY D.NUMERO, D.NOME, C.NOME;
```

--28.Crie uma consulta para exibir os seguintes itens referentes aos departamentos:

--o número seja maior que 80 com o salário total.

```
SELECT D.NUMERO, D.NOME AS DEPARTAMENTO, SUM(E.SALARIO) AS SAL_TOT  
FROM EMPREGADO E  
INNER JOIN DEPARTAMENTO D  
ON E.DEPARTAMENTO = D.NUMERO  
INNER JOIN CARGO C  
ON E.CARGO = C.NUMERO  
WHERE D.NUMERO > 80  
GROUP BY D.NUMERO, D.NOME;
```

--29.Crie uma consulta para exibir os seguintes itens referentes aos departamentos:

--o número seja maior que 80 com o salário total de cada cargo, independente do departamento.

```
SELECT SUM(E.SALARIO) AS SAL_TOT  
FROM EMPREGADO E  
INNER JOIN DEPARTAMENTO D  
ON E.DEPARTAMENTO = D.NUMERO  
INNER JOIN CARGO C  
ON E.CARGO = C.NUMERO  
WHERE D.NUMERO > 80;
```

--30.Crie uma consulta que exiba o agrupamento por departamento e por cargo.

```
SELECT E.DEPARTAMENTO, E.CARGO  
FROM EMPREGADO E  
GROUP BY E.DEPARTAMENTO, E.CARGO;
```

--31.Crie uma consulta para exibir os 2 empregados que ganham mais.

```
SELECT TOP 2 E.APELIDO, E.SALARIO  
FROM EMPREGADO E  
ORDER BY E.SALARIO DESC;
```

--32.Crie uma consulta para exibir o número, o sobrenome, o nascimento e o salário dos empregados com os nomes de seus respectivos gerentes.

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
SELECT E.NUMERO, E.SOBRENOME AS EMPREGADO, E.NASCIMENTO, E.SALARIO,  
G.SOBRENOME AS GERENTE  
FROM EMPREGADO E  
LEFT JOIN EMPREGADO G ON E.GERENTE = G.NUMERO;
```

--33.Crie uma consulta hierárquica para exibir o número do empregado, o número do gerente e o sobrenome de todos os empregados que estejam um nível abaixo do gerente.

```
SELECT E.NUMERO, 'GERENTE: ' + E.SOBRENOME AS RECURSO, E.NASCIMENTO,  
E.SALARIO  
FROM EMPREGADO E  
WHERE E.GERENTE IS NULL  
UNION  
SELECT E.NUMERO, 'EMPREGADO: ' + E.SOBRENOME AS RECURSO, E.NASCIMENTO,  
E.SALARIO  
FROM EMPREGADO E  
WHERE E.GERENTE IS NOT NULL;
```

--34.Exibir os 2 primeiros registros de empregados cujos os salários sejam maiores que 1500.  
SELECT TOP 2 E.APELIDO, E.SALARIO FROM EMPREGADO E WHERE E.SALARIO > 1500;

--35.Exibir a subtração das tabelas Cargo e Departamento para os campos número e nome.

```
SELECT C.NUMERO, C.NOME FROM CARGO C  
EXCEPT  
SELECT D.NUMERO, D.NOME FROM DEPARTAMENTO D;
```

--36.Exibir a intersecção das tabelas Cargo e Departamento para os campos número e nome.

```
SELECT C.NUMERO, C.NOME FROM CARGO C  
INTERSECT  
SELECT D.NUMERO, D.NOME FROM DEPARTAMENTO D;
```

--37.Exibir a união das tabelas Cargo e Departamento para os campos número e nome.

```
SELECT C.NUMERO, C.NOME FROM CARGO C  
UNION  
SELECT D.NUMERO, D.NOME FROM DEPARTAMENTO D;
```

--38.Exibir os empregados com os dados sobrenome, salário e número do departamento, com no mínimo uma pessoa subordinada a eles.

```
SELECT E.NUMERO, E.SOBRENOME, E.DEPARTAMENTO FROM EMPREGADO E WHERE  
E.GERENTE IS NULL AND EXISTS (SELECT 'X' FROM EMPREGADO X WHERE X.GERENTE =  
E.NUMERO);
```

--39.Exibir o somatório de Salário agrupado por Departamento e Cargo dos empregados com Número de Departamento menor que 60, exibindo os subtotais de Departamento e Cargo.

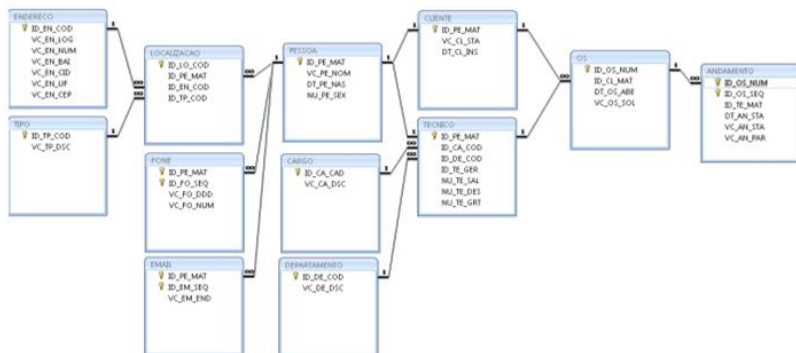
```
SELECT E.DEPARTAMENTO, E.CARGO, SUM(SALARIO) AS SOM_SAL  
FROM EMPREGADO E  
GROUP BY E.DEPARTAMENTO, E.CARGO WITH ROLLUP
```

--40.Quais os empregados possuem salário maior que o de João Abacateiro?

```
SELECT E.PRIMEIRONOME + ' ' + E.SOBRENOME AS NOME, E.SALARIO FROM EMPREGADO  
E WHERE E.SALARIO > (SELECT X.SALARIO FROM EMPREGADO X WHERE  
X.PRIMEIRONOME = 'JOÃO' AND X.SOBRENOME = 'ABACATEIRO')
```

--Baseado no modelo abaixo responda as questões abaixo:

Figura 6.1 Modelo Entidade Relacionamento



Fonte: Próprio Autor

--41) A matrícula e o nome dos clientes com a idade superior a 45 anos.

```

SELECT CLIENTE.ID_PE_MAT, PESSOA.VC_PE_NOM
FROM PESSOA INNER JOIN CLIENTE ON PESSOA.ID_PE_MAT = CLIENTE.ID_PE_MAT
WHERE (((Now()-[PESSOA].[DT_PE_NAS])>45));
    
```

--42) A matrícula dos técnicos cujos os nomes iniciem com A,F,G ou M.

```

SELECT TECNICO.ID_PE_MAT
FROM PESSOA INNER JOIN TECNICO ON PESSOA.ID_PE_MAT=TECNICO.ID_PE_MAT
WHERE PESSOA.VC_PE_NOM Like 'A%' OR PESSOA.VC_PE_NOM Like 'F%' OR
PESSOA.VC_PE_NOM Like 'G%' OR PESSOA.VC_PE_NOM Like 'M%';
    
```

--43) A quantidade de pessoas que possuem apenas uma localização.

```

SELECT LOCALIZACAO.ID_PE_MAT, Count(LOCALIZACAO.ID_LO_COD) AS QTDE
FROM LOCALIZACAO
GROUP BY LOCALIZACAO.ID_PE_MAT
HAVING Count(LOCALIZACAO.ID_LO_COD)=1;
    
```

--44) A quantidade de gestores.

```

SELECT Count(*) AS QTDE_GESTORES
FROM TECNICO
WHERE (((TECNICO.ID_TE_GER) Is Null));
    
```

--45) O nome e o sexo M se 0 e F se 1 para os departamentos técnico e comercial.

```

SELECT PESSOA.VC_PE_NOM, NU_PE_SEX
FROM DEPARTAMENTO INNER JOIN (PESSOA INNER JOIN TECNICO ON PESSOA.ID_PE_MAT
= TECNICO.ID_PE_MAT) ON DEPARTAMENTO.ID_DE_COD = TECNICO.ID_DE_COD
WHERE (((DEPARTAMENTO.VC_DE_DSC)="TÉCNICO")) OR
(((DEPARTAMENTO.VC_DE_DSC)="COMERCIAL"));
    
```

--46) Os 2 maiores salários líquidos dos técnicos.

```

SELECT TOP 2 TECNICO.NU_TE_SAL + TECNICO.NU_TE_GRT - TECNICO.NU_TE_DES AS LIQ
FROM TECNICO
ORDER BY 1 DESC;
    
```

--47) A média de salários base por departamento.

```

SELECT DEPARTAMENTO.VC_DE_DSC, Avg(TECNICO.NU_TE_SAL) AS MédiaDeNU_TE_SAL
FROM DEPARTAMENTO INNER JOIN TECNICO ON DEPARTAMENTO.ID_DE_COD =
TECNICO.ID_DE_COD
GROUP BY DEPARTAMENTO.VC_DE_DSC;
    
```

--48) A quantidade de ordens que foram canceladas e encerradas do ano corrente.

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
SELECT COUNT(ANDAMENTO.ID_OS_NUM) AS QTDE  
FROM ANDAMENTO  
WHERE (YEAR(ANDAMENTO.DT_AN_STA)=2013) AND  
(ANDAMENTO.VC_AN_STA="ENCERRADAS" OR ANDAMENTO.VC_AN_STA="CANCELADAS");
```

--49) A matrícula e o nome dos técnicos sem cargo informado.

```
SELECT TECNICO.ID_PE_MAT, PESSOA.VC_PE_NOM  
FROM PESSOA INNER JOIN TECNICO ON PESSOA.ID_PE_MAT = TECNICO.ID_PE_MAT  
WHERE (((TECNICO.ID_CA_COD) Is Null));
```

--50) A matrícula e o nome dos todos os técnicos e a descrição de seus departamentos caso exista a informação.

```
SELECT TECNICO.ID_PE_MAT, PESSOA.VC_PE_NOM, DEPARTAMENTO.VC_DE_DSC  
FROM DEPARTAMENTO RIGHT JOIN (PESSOA INNER JOIN TECNICO ON PESSOA.ID_PE_MAT  
= TECNICO.ID_PE_MAT) ON DEPARTAMENTO.ID_DE_COD = TECNICO.ID_DE_COD;
```

--51) A matrícula e o nome dos técnicos e a descrição de todos os departamentos caso exista a informação.

```
SELECT TECNICO.ID_PE_MAT, PESSOA.VC_PE_NOM, DEPARTAMENTO.VC_DE_DSC  
FROM DEPARTAMENTO LEFT JOIN (PESSOA RIGHT JOIN TECNICO ON PESSOA.ID_PE_MAT  
= TECNICO.ID_PE_MAT) ON DEPARTAMENTO.ID_DE_COD = TECNICO.ID_DE_COD;
```

--52) A soma dos salários base dos técnicos por cargo cujo o salário base seja maior que 2000.

```
SELECT CARGO.VC_CA_DSC, Sum(TECNICO.NU_TE_SAL) AS SomaDeNU_TE_SAL  
FROM CARGO INNER JOIN TECNICO ON CARGO.ID_CA_CAD=TECNICO.ID_CA_COD  
WHERE (((TECNICO.NU_TE_SAL)>2000))  
GROUP BY CARGO.VC_CA_DSC;
```

--53) A soma dos salários base dos técnicos por cargo cujo o valor da soma seja maior que 5000.

```
SELECT CARGO.VC_CA_DSC, Sum(TECNICO.NU_TE_SAL) AS SomaDeNU_TE_SAL  
FROM CARGO INNER JOIN TECNICO ON CARGO.ID_CA_CAD = TECNICO.ID_CA_COD  
GROUP BY CARGO.VC_CA_DSC  
HAVING Sum(TECNICO.NU_TE_SAL) > 5000;
```

--54) A soma dos salários líquidos dos técnicos por cargo cujo a gratificação seja maior que 1000 e a soma dos salários líquido seja maior que 3000.

```
SELECT CARGO.VC_CA_DSC, Sum(TECNICO.NU_TE_SAL+TECNICO.NU_TE_GRT-  
TECNICO.NU_TE_DES) AS LIQ  
FROM CARGO INNER JOIN TECNICO ON CARGO.ID_CA_CAD = TECNICO.ID_CA_COD  
WHERE TECNICO.NU_TE_GRT > 1000  
GROUP BY CARGO.VC_CA_DSC  
HAVING Sum(TECNICO.NU_TE_SAL+TECNICO.NU_TE_GRT-TECNICO.NU_TE_DES) > 3000;
```

--55) Criar índice que não permita duplicidade para nomes dos cargos.

```
CREATE UNIQUE INDEX IX01 ON CARGOS (VC_CA_NOM);
```

--56) Nomes das pessoas sem duplicidade.

```
SELECT DISTINCT PESSOA.VC_PE_NOM  
FROM PESSOA;
```

--57) Quantidade de OS abertas mas não atendidas no mês corrente.

```
SELECT OS.DT_OS_ABE  
FROM OS INNER JOIN ANDAMENTO ON OS.ID_OS_NUM <> ANDAMENTO.ID_OS_NUM  
WHERE MONTH(OS.DT_OS_ABE) = MONTH(Now());
```

--58) A matrícula, o nome e as descrições dos departamentos e cargos dos técnicos.

```
SELECT TECNICO.ID_PE_MAT, PESSOA.VC_PE_NOM, DEPARTAMENTO.VC_DE_DSC,  
CARGO.VC_CA_DSC
```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
FROM CARGO INNER JOIN (DEPARTAMENTO INNER JOIN (PESSOA INNER JOIN TECNICO ON  
PESSOA.ID_PE_MAT = TECNICO.ID_PE_MAT) ON DEPARTAMENTO.ID_DE_COD =  
TECNICO.ID_DE_COD) ON CARGO.ID_CA_CAD = TECNICO.ID_CA_COD;
```

--59) A matrícula, o nome, os e-mails e fones dos clientes.

```
SELECT CLIENTE.ID_PE_MAT, PESSOA.VC_PE_NOM, FONE.VC_FO_NUM,  
EMAIL.VC_EM_END  
FROM ((PESSOA INNER JOIN CLIENTE ON PESSOA.ID_PE_MAT = CLIENTE.ID_PE_MAT)  
INNER JOIN FONE ON PESSOA.ID_PE_MAT = FONE.ID_PE_MAT) INNER JOIN EMAIL ON  
PESSOA.ID_PE_MAT = EMAIL.ID_PE_MAT;
```

--60) A matrícula, o nome, o fone, o e-mail do cliente da solicitação de OS=15.

```
SELECT CLIENTE.ID_PE_MAT, PESSOA.VC_PE_NOM, FONE.VC_FO_NUM,  
EMAIL.VC_EM_END  
FROM ((PESSOA INNER JOIN (CLIENTE INNER JOIN OS ON CLIENTE.ID_PE_MAT =  
OS.ID_CL_MAT) ON PESSOA.ID_PE_MAT = CLIENTE.ID_PE_MAT) INNER JOIN EMAIL ON  
PESSOA.ID_PE_MAT = EMAIL.ID_PE_MAT) INNER JOIN FONE ON PESSOA.ID_PE_MAT =  
FONE.ID_PE_MAT  
WHERE (((OS.ID_OS_NUM)=15));
```

--61) O parecer da solicitação de número 25.

```
SELECT ANDAMENTO.ID_OS_NUM, ANDAMENTO.VC_AN_PAR  
FROM ANDAMENTO  
WHERE (((ANDAMENTO.ID_OS_NUM)=25));
```

--62) O andamento da solicitação de número 30.

```
SELECT ANDAMENTO.*  
FROM ANDAMENTO  
WHERE (((ANDAMENTO.ID_OS_NUM)=30));
```

--63) A(s) matrícula(s) e nome(s) do(s) técnico(s) que possuem o maior salário.

```
SELECT TECNICO.ID_PE_MAT, PESSOA.VC_PE_NOM  
FROM PESSOA INNER JOIN TECNICO ON PESSOA.ID_PE_MAT = TECNICO.ID_PE_MAT  
WHERE NU_TE_SAL = (SELECT MAX(NU_TE_SAL) FROM TECNICO);
```

--64) O(s) nome(s) do(s) técnico(s) que mais realizaram atendimentos.

```
SELECT TECNICO.ID_PE_MAT, PESSOA.VC_PE_NOM  
FROM PESSOA INNER JOIN TECNICO ON PESSOA.ID_PE_MAT = TECNICO.ID_PE_MAT  
WHERE TECNICO.ID_PE_MAT IN (SELECT TOP 2 TECNICO.ID_PE_MAT  
FROM TECNICO INNER JOIN ANDAMENTO ON TECNICO.ID_PE_MAT =  
ANDAMENTO.ID_TE_MAT  
GROUP BY TECNICO.ID_PE_MAT  
ORDER BY Count(ANDAMENTO.ID_OS_NUM) DESC);
```

--65) A quantidade de caracteres existentes no nome das pessoas.

```
SELECT LEN(PESSOA.VC_PE_NOM) AS QTDE  
FROM PESSOA;
```

--66) A união dos cargos e departamentos ordenado por descrição decrescente.

```
SELECT ID_CA_CAD, VC_CA_DSC  
FROM CARGO  
UNION SELECT ID_DE_COD, VC_DE_DSC  
FROM DEPARTAMENTO;
```

--67) O menor salário líquido pago por departamento.

```
SELECT DEPARTAMENTO.VC_DE_DSC, Min(TECNICO.NU_TE_SAL+TECNICO.NU_TE_GRT-  
TECNICO.NU_TE_DES) AS LIQ
```

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

```
FROM DEPARTAMENTO INNER JOIN TECNICO ON DEPARTAMENTO.ID_DE_COD =  
TECNICO.ID_DE_COD  
GROUP BY DEPARTAMENTO.VC_DE_DSC;
```

--68) O maior salário líquido pago por departamento e por cargo.

```
SELECT DEPARTAMENTO.VC_DE_DSC, CARGO.VC_CA_DSC,  
Max([TECNICO].[NU_TE_SAL]+[TECNICO].[NU_TE_GRT]-[TECNICO].[NU_TE_DES]) AS LIQ  
FROM CARGO INNER JOIN (DEPARTAMENTO INNER JOIN TECNICO ON  
DEPARTAMENTO.ID_DE_COD = TECNICO.ID_DE_COD) ON CARGO.ID_CA_CAD =  
TECNICO.ID_CA_COD  
GROUP BY DEPARTAMENTO.VC_DE_DSC, CARGO.VC_CA_DSC;
```

## 7 Referências Bibliográficas

- BEIGHLEY, Lynn. Use a Cabeça SQL. 1ªed. Rio de Janeiro: Alta Books, 2008.
- CARVALHO, Rogério. SQL Guia Prático. 2ªed. Rio de Janeiro: Brasport, 2007.
- COMER, Douglas. Redes de Computadores e Internet. 4ªed. US: Bookman, 2007.
- DATE, Christopher. Introdução a Sistemas de Banco de Dados. 8ªed. Rio de Janeiro: Campus, 2004.
- DEITEL, Paul; DEITEL, Harvey. Ajax, Rich Internet Applications. 1ªed. US: Prentice Hall, 2008.
- MANZANO, Alves. Guia de Orientação e Desenvolvimento de Sites. 1ªed. São Paulo: Érica, 2008.
- MISTRY, Ross. Introducing Microsoft SQL Server 2008 R2. Washington: Microsoft Press, 2014.
- O'BRIEN, James. Sistemas de Informação e as Decisões Gerenciais na Era da Internet. 2ªed. Saraiva, 2004.
- VALADARES, Fabrício. SQL dos Conceitos às Consultas Complexas. Ciência Moderna, 2009.
- VIREIRA, Eduardo. Os bastidores da Internet no Brasil. 1ªed. São Paulo: Manole, 2003.

## 8 Referências Web

AJAX (wiki). Disponível em < <http://goo.gl/PFOd> >  
ASPX (wiki). Disponível em < <http://goo.gl/1cyFrR> >  
Aulas de SQL. Disponível em < <http://goo.gl/BbpyDJ> >  
Exemplos de Utilização dos Comandos SQL. Disponível em < <http://goo.gl/QSRtlv> >  
Guia de Referência do SQLServer. Disponível em < <http://goo.gl/01O84d> >  
HTML (wiki). Disponível em < <http://goo.gl/clQQz> >  
JavaScript (wiki). Disponível em < <http://goo.gl/Gsw2g> >  
Padrão da Linguagem SQL. Disponível em < <http://goo.gl/xwKvG1> >  
Tecnologias da Internet. Disponível em < <http://goo.gl/16DUpW> >  
Tim O'Reilly, O que é Web 2.0?. Disponível em < <http://goo.gl/9b4znT> >  
XHTML. Disponível em < <http://goo.gl/UQZVQ> >  
XML. Disponível em < <http://goo.gl/39NB> >

### 9 Biografias

**Alan Turing** nasceu na Inglaterra em 1912, foi matemático, lógico, criptoanalista e cientista da computação. É um dos nomes importantes na Ciência da Computação. Formulou conceitos de algoritmo e computação com a máquina de Turing. Trabalhou no *National Physical Laboratory*. Saber mais.

Disponível em < [http://en.wikipedia.org/wiki/Alan\\_Turing](http://en.wikipedia.org/wiki/Alan_Turing) >

**Alfred Vaino Aho** é cientista da computação canadense. Conhecido internacionalmente como co-autor da linguagem de programação AWK, junto com Peter Weinberger e Brian Kernighan, e também pela sua co-autoria no Livro do Dragão como é mais conhecido o livro, *Compilers: Principles, Techniques, and Tools*, escrito em parceria com Ravi Sethi e Jeffrey Ullman. Suas contribuições para área foram às versões iniciais das ferramentas Unix egrep e fgrep. Também possui livros nas áreas de algoritmos e estruturas de dados. Saber mais.

Disponível em < [http://pt.wikipedia.org/wiki/Alfred\\_Aho](http://pt.wikipedia.org/wiki/Alfred_Aho) >

**Avram Noam Chomsky** nasceu nos Estados Unidos na Filadélfia em 1928, é um linguista e filósofo. É professor de Linguística no *Massachusetts Institute of Technology*. É autor de trabalhos fundamentais sobre as propriedades matemáticas das linguagens formais, sendo o seu nome associado à chamada Hierarquia de Chomsky. Seus trabalhos estão relacionados a uma abordagem matemática da linguagem. Saber mais.

Disponível em < [http://pt.wikipedia.org/wiki/Noam\\_Chomsky](http://pt.wikipedia.org/wiki/Noam_Chomsky) >

**Christopher Date** formou-se no Reino Unido pela High Wycombe Royal Grammar School em 1958 e recebeu o bacharelado em Matemática pela Universidade de Cambridge em 1962. Iniciou as atividades na área da computação como programador na *Leo Computers Ltd* em Londres onde também começou a lecionar. Em 1967 começou a trabalhar na IBM na Inglaterra também como programador e instrutor de computação, ficando entre os anos de 1969 e 1974 como o principal instrutor da área educacional da IBM. Saber mais.

Disponível em < [http://en.wikipedia.org/wiki/Christopher\\_J.\\_Date](http://en.wikipedia.org/wiki/Christopher_J._Date) >

**Craig Larman** é um canadense nascido em 1958 que trabalha como cientista da computação especializado em desenvolvimento iterativo e incremental, desenvolvimento de *software* ágil, análise orientada a objetos e modelagem ágil. Ele é o autor de vários textos sobre Scrum. Saber mais.

Disponível em < [http://en.wikipedia.org/wiki/Craig\\_Larman](http://en.wikipedia.org/wiki/Craig_Larman) >

**Erich Gamma** é um dos quatro autores do livro *Design Patterns: Elements of Reusable Object-Oriented Software*. Ele é também o principal projetista do JUnit e do Eclipse. Nascido em 1961, em Zurique é cientista da computação e co-autor influente de livros didáticos sobre engenharia de *software*. Co-escreveu o JUnit framework de teste de *software* com Kent Beck e liderou o projeto do Eclipse da plataforma *Java Development Tools* (JDT). Também trabalhou na IBM Rational projeto Jazz. Saber mais.

Disponível em < [http://en.wikipedia.org/wiki/Erich\\_Gamma](http://en.wikipedia.org/wiki/Erich_Gamma) >

**Grady Booch** nasceu em 1955 nos Estados Unidos, é engenheiro de software e cientista-chefe da IBM Research. Obteve seu grau de bacharel pela *United States Air Force Academy* e título de mestre em engenharia elétrica pela Universidade da Califórnia em Santa Barbara. Saber mais.

Disponível em < [http://en.wikipedia.org/wiki/Grady\\_Booch](http://en.wikipedia.org/wiki/Grady_Booch) >

**Hal Varian**, nascido em 1947, em Ohio, é economista especializado em microeconomia e economia da informação. É professor emérito nas universidades de Berkeley e Haas *School of Business*. Formou-se como bacharel em economia pelo e licenciado em matemática pelo MIT. Possui o doutorado em economia pela *Berkeley University*, e é doutor honorário pelo Instituto de Tecnologia de Karlsruhe, Alemanha. Já lecionou em *Stanford University*, *Oxford University*, *Michigan University*. Nos dias atuais é o economista-chefe da Google e publica regularmente artigos no *New York Times* e *Wall Street Journal*. Saber mais.

Disponível em < [http://en.wikipedia.org/wiki/Hal\\_Varian](http://en.wikipedia.org/wiki/Hal_Varian) >

<http://www.alvarofpinheiro.eti.br>

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

**Harvey Deitel** tem mais de 40 anos de experiência na área de computação. Graduou-se e fez o mestrado no *Massachusetts Institute of Technology* e o doutorado na *Boston University*. Possuindo mais de 20 anos de experiência no ensino universitário. Possuindo diversas publicações traduzidas em diversos idiomas. Saber mais.

Disponível em < [http://www.informit.com/authors/author\\_bio.aspx?ISBN=9780137058426](http://www.informit.com/authors/author_bio.aspx?ISBN=9780137058426) >

**Ian Sommerville**, inglês nascido em 1951 é professor titular de engenharia de software na Universidade de St. Andrews na Escócia. Destacado pesquisador no campo da engenharia de sistemas, segurança e informática social. Saber mais.

Disponível em < [http://en.wikipedia.org/wiki/Ian\\_Sommerville\\_%28academic%29](http://en.wikipedia.org/wiki/Ian_Sommerville_%28academic%29) >

**Idalberto Chiavenato**, nascido em 1936 em São Paulo, é autor na área de administração de empresas e de recursos humanos. Atualmente, atua como conselheiro no Conselho Regional de Administração do Estado de São Paulo (CRA-SP) e é Presidente do Instituto Chiavenato de Educação. Saber mais.

Disponível em < [http://pt.wikipedia.org/wiki/Idalberto\\_Chiavenato](http://pt.wikipedia.org/wiki/Idalberto_Chiavenato) >

**Ivar Jacobson** é um sueco nascido em 1939, se graduou no Instituto de Tecnologia de Chalmers, onde fez seu mestrado em Engenharia Elétrica e possuindo doutorado pela *Royal Institute of Technology de Estocolmo*. Saber mais.

Disponível em < [http://en.wikipedia.org/wiki/Ivar\\_Jacobson](http://en.wikipedia.org/wiki/Ivar_Jacobson) >

**James Gosling** nasceu em Calgary no Canadá em 1955 é conhecido como o pai da linguagem de programação Java. Em 1977, formou Bacharel em Ciência da Computação pela Universidade de Calgary, e em 1983 tornou-se PhD em Ciência da Computação pela Universidade Carnegie Mellon. Enquanto estudava no doutorado, Gosling programou a versão original do editor de texto Emacs para UNIX, e antes de se juntar a Sun Microsystems ele construiu uma versão de UNIX para máquinas multiprocessadas, assim como diversos compiladores e sistemas de mensagens eletrônicas. Trabalhou de 1984 até 2010 na Sun Microsystems, atualmente está na Google. Saber mais.

Disponível em < [http://pt.wikipedia.org/wiki/James\\_Gosling](http://pt.wikipedia.org/wiki/James_Gosling) >

**James Rumbaugh** é Bacharel em Física pelo Massachusetts Institute of Technology (MIT), possui mestrado em Astronomia pela Caltech, e é Ph. D. em Ciência da Computação também pelo MIT. Trabalhou na *Digital Equipment Corporation* (DEC), como cientista. Foi um dos líderes na criação da OMT e em 1994 se juntou a *Rational Software*, e junto com Ivar Jacobson e Booch de Grady desenvolveram a *Unified Modeling Language* (UML). Mais tarde, eles fundiram as suas metodologias de desenvolvimento de *software*, OMT, OOSE e Booch em um único processo batizado de *Rational Unified Process* (RUP). Saber mais.

Disponível em < [http://en.wikipedia.org/wiki/James\\_Rumbaugh](http://en.wikipedia.org/wiki/James_Rumbaugh) >

**John von Neumann** nasceu na Hungria em 1903 naturalizado como estadunidense, foi um matemático. Membro do Instituto de Estudos Avançados em Princeton, *New Jersey*, do qual também faziam parte Albert Einstein. Professor na Universidade de Princeton e um dos construtores do ENIAC. Trabalhou no desenvolvimento da Física Nuclear e contribuiu com diversas teorias, a exemplo as dos conjuntos, análise funcional, ergódica, mecânica quântica, ciência da computação, economia, teoria dos jogos, análise numérica, estatística, entre outras. Saber mais.

Disponível em < [http://pt.wikipedia.org/wiki/John\\_von\\_Neumann](http://pt.wikipedia.org/wiki/John_von_Neumann) >

**Kenneth Laudon** é professor de Sistemas de Informação na New York University's Stern School of Business. Bacharel em Economia pela *Stanford University* e Ph. D pela *Columbia University*. Autor de vários livros que tratam do tema Sistemas de Informação, Organizações e Sociedade. Saber mais.

Disponível em < <http://pages.stern.nyu.edu/~klaudon/> >

**Patrick Naughton** americano nascido em 1965 é um desenvolvedor de software conhecido como sendo um dos criadores originais da linguagem de programação Java, pois trabalho na Sun

<http://www.alvarofpinheiro.eti.br>

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

Microsystems no Projeto Verde a origem do Java. Em 1994, saiu da Sun e foi para Starwave de propriedade de Paul Allen para desenvolver um servidor Java. É autor do livro *O Manual do Java*, Osborne, 1995 e co-autor do *Java: The Complete Reference*, com Herbert Schildt, Osborne, 1996. Em 1998, torna-se vice-presidente executivo da Infoseek. Saber mais.

Paul Deitel é graduado em Tecnologia da Informação e pós-graduação em Administração pela Massachusetts *Institute of Technology*. Trabalha como consultor em diversas organizações internacionais. Saber mais.

Disponível em < [http://www.informit.com/authors/author\\_bio.aspx?ISBN=9780137058426](http://www.informit.com/authors/author_bio.aspx?ISBN=9780137058426) >

Peter Ferdinand Drucker, austríaco nascido 1909 e que faleceu recentemente em 2005, foi um filósofo e economista, que por suas contribuições a comunidade, passou a ser considerado o pai da administração moderna, se tornando um dos pensadores mais citados pela teoria da globalização em economia em geral. Saber mais.

Disponível em < [http://pt.wikipedia.org/wiki/Peter\\_Drucker](http://pt.wikipedia.org/wiki/Peter_Drucker) >

Roger Pressman é Bacharel em Engenharia de Software pela University of Connecticut, com Mestrado pela *University of Bridgeport* e PhD pela *University of Connecticut*. Pertence ao conselho editorial da *IEEE Software* e do *The Cutter Tl Journal*. Autor e consultor na área de Engenharia de Software. Trabalha na área de engenharia de software a mais de 30 anos, atuando como administrador, professor e consultor. É presidente da *R.S. Pressman & Associates, Inc.*, empresa de consultoria especializada em práticas de Engenharia de Software. Saber mais. Disponível em < <http://www.cutter.com/meet-our-experts/rpbio.html> >

Stephen Schach é professor emérito do curso de Ciência da Computação e Engenharia de Computação do Departamento de Engenharia Elétrica e Ciência da Computação da Universidade de Vanderbilt. Possui mestrado em Física pelo *Weizmann Institute of Science* e doutorado em Matemática Aplicada pela *University of Cape Town*. Saber mais.

Disponível em < <http://www.vuse.vanderbilt.edu/~srs/> >

Tim O'Reilly, nasceu em 1954 na Irlanda, é o fundador da O'Reilly Media e estudioso e fomentador de software livre, para muitos é o criador da expressão Web 2.0. É Bacharel em filologia clássica pela Harvard University em 1975. Saber mais.

Disponível em < [http://pt.wikipedia.org/wiki/Tim\\_O'Reilly](http://pt.wikipedia.org/wiki/Tim_O'Reilly) >

Tom DeMarco, nascido em 1940 na Pensilvânia, é autor, professor e palestrante em engenharia de software e é reconhecido como um dos criadores da Análise Estruturada nos anos 1980. É Bacharel em Engenharia Elétrica pela Universidade de *Cornell University*, Mestre pela *Columbia University* e Doutor pela University of Paris at the Sorbonne. Saber mais.

Disponível em < [http://en.wikipedia.org/wiki/Tom\\_DeMarco](http://en.wikipedia.org/wiki/Tom_DeMarco) >

## 10 Artefatos (Templates) para Projetos de Software

Especificação de Caso de Uso de Negócios Nome do Caso de Uso de Negócios. Disp.:<  
<http://goo.gl/XkILXt> >

Especificação de Caso de Uso Nome do Caso de Uso. Disponível em < <http://goo.gl/8OsCF8> >

Especificação dos Requisitos de Software. Disponível em < <http://goo.gl/dfnJMy> >

Plano de Desenvolvimento de Software (Projeto Pequeno). Disponível em < <http://goo.gl/J9gsvB> >

Plano de Desenvolvimento de Software. Disponível em < <http://goo.gl/5KRFCJ> >

Plano de Requisitos. Disponível em < <http://goo.gl/sLM5Lu> >

Regras de Negócios. Disponível em < <http://goo.gl/Zx5CWY> >

## 11 Ferramentas

Administração SQLServer Studio. Disponível em < <http://goo.gl/4sn8XZ> >

Administração SQLServer. Disponível em < <http://goo.gl/P01RTZ> >

Easyecclipse. Disponível em < <http://sourceforge.net/projects/easyecclipse/files/> >

Instalação SQLServer. Disponível em < <http://goo.gl/K7IPPI> >

Modelagem DBDesigner. Disponível em < <http://goo.gl/mG2VLh> >

## 12 Apresentações e Vídeos

BD. Disponível em < <http://www.slideshare.net/alvarofpinheiro/fundamentos-de-banco-de-dados-relacionais> >

Script da Aula de 29 Abril 2014. Disponível em < <http://goo.gl/dsYCwi> >

### 13 Sugestões de Leitura

Evolução da Programação Web. Disponível em < <http://goo.gl/8Rqnlq> >  
Fundamentos de Programação. Disponível em < <http://goo.gl/xwYOfC> >  
Introdução a Sistemas de Banco de Dados. Disponível em < <http://goo.gl/7w6bjia> >  
Lógica de Programação. Disponível em < <http://goo.gl/TuRR9d> >  
SQL Guia Prático. Disponível em < <http://goo.gl/qcYxz8> >  
Tecnologias da Internet. Disponível em < <http://goo.gl/16DUp> >  
Usabilidade de Interfaces na Web. Disponível em < <http://goo.gl/SwbleX> >  
Web 2.0 e Mashups. Disponível em < <http://goo.gl/Bd5FFN> >

## Fundamentos da Engenharia de Software: Linguagem de Programação C#

### Livros da série Fundamentos da Engenharia de Software



Fundament  
os da  
Engenharia  
de  
Software:  
Conceitos  
Básicos é  
uma  
coletânea  
de

disciplinas que integradas servem para fundamentar o entendimento da construção de projetos de software com qualidade, isto é, baseado em processos maduros e reconhecidos pela comunidade tecnológica. O objetivo deste livro é fornecer ao leitor as bases necessárias para o desenvolvimento de aplicações sejam Desktop, Web ou Mobile. Iniciando a leitura na Teoria da Computação, passando por Processos, Linguagens, Bancos de Dados e finalizando com Sistemas de Informação e Colaboração. Este livro pode ser lido capítulo a capítulo ou somente a disciplina desejada, pois sua elaboração consiste na compilação das disciplinas fundamentais da Engenharia de Software que são independentes, mas ao mesmo tempo se integram objetivando o desenvolvimento de aplicações.



Introdução à  
Banco de  
Dados.  
Neste são  
abordados  
os conceitos  
básicos de  
bancos de  
dados e  
seus

sistemas gerenciadores, mas com o foco na arquitetura relacional, porque ainda hoje o mercado faz uso em larga escala desses bancos de dados, mesmo que o paradigma predominante seja o orientado a objetos e que, já existam a um bom tempo bancos orientados a objeto, até mesmo os bancos objetos-relacionais que são um híbrido entre essas duas arquiteturas, o que predomina ainda é o relacional, assim, este material é focado na linguagem de consulta estruturada para os SGBD-Rs do mercado, com foco na comparação de cinco dos mais utilizados bancos relacionais, os quais são: Oracle, SQLServer, MySQL, SQLBase e Interbase.



Este livro é  
sobre  
processos  
de  
desenvolvim  
ento de  
software,  
evidenciand  
o a  
necessidade

de qualidade na construção de sistemas, conceituando a diferença entre desenvolvimento Adhoc e com processo. Para isso é realizado a introdução à engenharia de requisitos abordando as técnicas para a elicitação de requisitos que forneçam subsídios necessários para uma construção de software com maior qualidade, enfatizando a necessidade de se aplicar na construção de qualquer sistema as técnicas de análise e modelagem, evidenciando o uso da linguagem da Linguagem de Modelagem Unificada (UML) para diagramar um projeto de software, explicando a necessidade do uso de modelos na construção, entrando com detalhes na análise orientada a objetos, com o objetivo de explorar os seus conceitos de requisitos e modelagem integrados. Este material é finalizado com a introdução à medidas de esforço de desenvolvimento, técnica necessária para responder as perguntas básicas de qualquer desenvolvimento: Qual o prazo e custo? E para responder a essas questões é abordado o uso da métrica análise de ponto de função.

## Fundamentos da Engenharia de Software: Linguagem de Programação C#



Este livro aborda os sistemas que são classificados como informação, a exemplo, sistemas de apoio a

decisão, sistemas estratégicos, sistemas gerenciais e sistemas transacionais. A produção deste material que compõe o volume 4 da coleção Fundamentos da Engenharia de Software é resultado da compilação das aulas produzidas nas disciplinas que compõem os capítulos deste livro.



A motivação deste livro é exemplificar os conceitos de Padrões de Projetos utilizando a linguagem de programação

o Java, sendo a construção uma compilação das aulas produzidas com o intuito de facilitar o entendimento do assunto abordando os seguintes temas: Paradigma Orientado a Objetos que introduz o leitor nos conceitos do POO; Linguagem de Modelagem Unificada para apresentar a simbologia UML dos conceitos de POO; Linguagem de Programação Java apresentando essa poderosa linguagem de programação orientada a objetos para exemplificar os padrões de projeto; e Padrões de Projetos que neste livro aborda os mais referenciados nas academias, sendo eles o GRASP e GoF.



Este livro é o resultado do uso da ferramenta MS Project da Microsoft utilizada na aplicação dos conceitos de

gestão de projetos do PMBOK com as premissas da engenharia de testes para aquisição de qualidade nos produtos de software.



Este livro aborda basicamente e os conceitos básicos de programação o como autômatos, tipos de

linguagens, princípios dos compiladores, paradigmas de desenvolvimento e lógica de programação.



Este livro introduz nas tecnologias Web abordando os conceitos básicos para desenvolvimento para

Internet com a apresentação da plataforma Dot Net e exibindo dicas de codificação para a linguagem de marcação ASPX, para a linguagem de script mais utilizada pelos navegadores o JavaScript com exemplos de CSS e principalmente dicas de código para a linguagem de programação CSharp e de banco de dados SQL com foco no SQLServer.