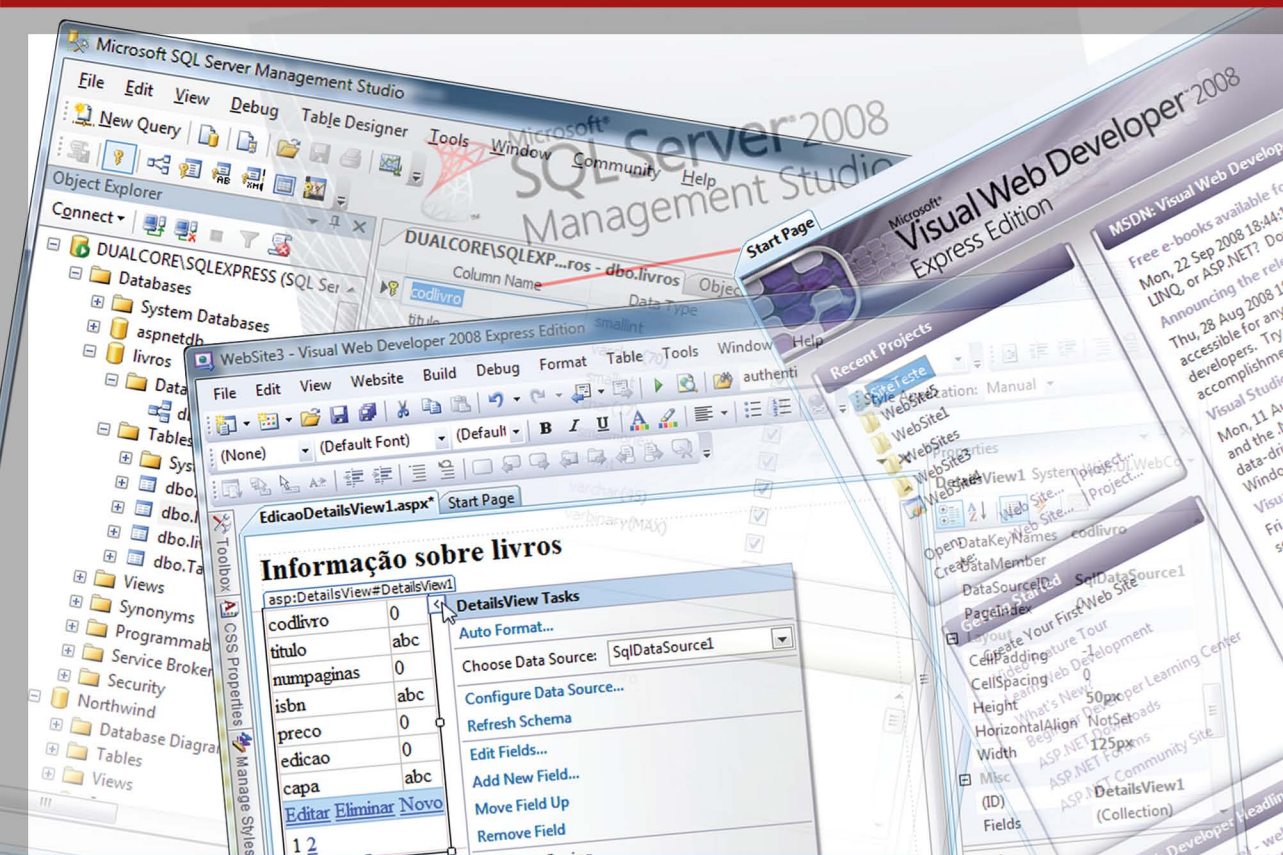


O GUIA PRÁTICO DO

ASP.NET

com

SQL Server 2008



PEDRO REMOALDO

O GUIA PRÁTICO DO

ASP.NET

com

SQL Server 2008



CENTRO ATLÂNTICO .PT

Portugal/2008

'Microsoft' is a registered trademark of Microsoft Corporation in the United States and/or other countries and is used by Centro Atlântico under license from owner. 'O Guia Prático do ASP.NET com SQL Server 2008' is an independent publication not affiliated with Microsoft Corporation.

Reservados todos os direitos por Centro Atlântico, Lda.

Qualquer reprodução, incluindo fotocópia, só pode ser feita com autorização expressa dos editores da obra.

O GUIA PRÁTICO DO ASP.NET COM SQL SERVER 2008

Colecção: Tecnologias

Autor: Pedro Remoaldo

Direcção gráfica: Centro Atlântico

Revisão técnica: Centro Atlântico

Capa: Paulo Buchinho

© Centro Atlântico, Lda., 2008

Ap. 413 – 4764-901 V. N. Famalicão

Rua da Misericórdia, 76 – 1200-273 Lisboa

Portugal

Tel. 808 20 22 21

geral@centroatlantico.pt

www.centroatlantico.pt

Impressão e acabamento: Inova

1ª edição: Outubro de 2008

ISBN: 978-989-615-070-9

Depósito legal: /08

Marcas registadas: Todos os termos mencionados neste livro conhecidos como sendo marcas registadas de produtos e serviços foram apropriadamente capitalizados. A utilização de um termo neste livro não deve ser encarada como afectando a validade de alguma marca registada de produto ou serviço.

O Editor e os Autores não se responsabilizam por possíveis danos morais ou físicos causados pelas instruções contidas no livro nem por endereços Internet que não correspondam às *Home-Pages* pretendidas.

Índice

PREFÁCIO	13
Introdução	13
Público-alvo e pré-requisitos	14
Características principais do livro.....	14
Convenções usadas.....	14
Visão geral e Organização.....	15
 SOBRE O AUTOR	 17
 1 DESENVOLVIMENTO DE SITES WEB	 19
1.1 Páginas estáticas	19
1.2 Páginas interactivas	20
1.3 Tecnologias <i>client-side</i>	21
1.4 Páginas dinâmicas	22
1.4.1 Metodologia das três camadas	24
1.4.2 Ambientes de desenvolvimento e produção	25
1.5 A .NET Framework.....	26
1.5.1 Arquitectura	27
1.6 ASP.NET	29
 2 VISUAL STUDIO E VISUAL WEB DEVELOPER EXPRESS Ed.	 31
2.1 Visual Studio	31
2.2 Visual Web Developer Express Edition	32
2.3 Instalar o Visual Web Developer	33
2.4 Primeiro arranque	37
 3 CRIAR UM SITE WEB	 41
3.1 Criar um <i>site</i> Web	42
3.2 Localização de <i>sites</i>	44
3.2.1 Criar um <i>site</i> Web no File System	44
3.2.2 Criar um <i>site</i> Web IIS local	46
3.2.3 Criar um <i>site</i> Web IIS remoto.....	53
3.2.4 Criar um <i>site</i> Web acessível por FTP	54
3.2.5 Configuração de opções	56
3.3 Abrir um <i>site</i> existente.....	57
3.3.1 Abrir um <i>site</i> recentemente utilizado.....	57
3.3.2 Abrir um <i>site</i>	59

3.4 Fechar projectos e alternar entre projectos	62
3.5 Visual Web Developer e aplicações IIS	63
3.6 Solution Explorer	63
3.6.1 Pastas especiais	66
3.7 Páginas Web ASP.NET	66
3.7.1 Abrir páginas Web ASP.NET	69

4 AMBIENTE DE TRABALHO 71

4.1 Painéis e barras de ferramentas	75
4.1.1 Painéis	75
4.1.2 Barras de ferramentas	79
4.1.3 Outros painéis	79
4.2 Alterar a disposição dos painéis	80
4.2.1 Esconder os painéis	82
4.2.2 Personalizar as barras de ferramentas	83
4.2.3 Janela Documento	85
4.3 Ajuda	86

5 CONTROLOS WEB E LAYOUT DE PÁGINAS 89

5.1 TOOLBOX	90
5.1.1 Personalizar a TOOLBOX	91
5.2 Layout de páginas	93
5.2.1 Flow layout	94
5.2.2 Two-Dimensional Positioning	94
5.2.3 Utilizar os tipos de layout disponíveis	95
5.2.4 Régua e grelha	97
5.3 Estrutura de uma página ASP.NET	98
5.3.1 Colocar controlos em páginas Web	99
5.4 Controlos HTML	99
5.4.1 Tabelas	100
5.4.2 Divs	101
5.4.3 Imagens	102
5.4.4 Links	102
5.4.5 Transformar controlos HTML em controlos de servidor	104
5.5 Web Server Controls	104
5.6 Adicionar controlos a partir do Solution Explorer	106
5.7 Seleccionar controlos	107
5.8 Definir as propriedades de um controlo	109
5.8.1 Utilizar o painel PROPERTIES	109
5.8.2 Utilizar a vista SOURCE	111
5.8.3 Smart Tags	112
5.8.4 Menu FORMAT	112
5.8.5 Barras de ferramentas	112

6 CSS	113
6.1 Localização dos estilos CSS	113
6.1.1 Estilos <i>inline</i>	113
6.1.2 Estilos internos utilizando o elemento <code><style></code>	117
6.1.3 Estilos externos	118
6.2 Trabalhar com estilos CSS	119
6.2.1 Criar um ficheiro externo de estilos CSS	119
6.2.2 Definir estilos internos	125
6.2.3 Aplicar estilos CSS	130
6.2.4 Mover estilos	135
6.2.5 Gestão de estilos	137
6.3 Opções de personalização	137
 7 MASTER PAGES, THEMES E SKINS	 139
7.1 Centralizar o <i>layout</i> das páginas com <i>master pages</i>	139
7.1.1 Criar uma <i>master page</i>	140
7.1.2 <i>Master page</i> exemplo	142
7.1.3 Aplicar <i>master pages</i> a <i>content pages</i>	155
7.2 Controlar a aparência com os <i>themes</i>	162
7.2.1 Criar <i>themes</i>	162
7.2.2 Adicionar <i>skins</i> a um <i>theme</i> ASP.NET	163
7.2.3 Adicionar folhas de estilo CSS a um <i>theme</i>	166
7.2.4 Aplicar <i>themes</i> a páginas, <i>sites</i> e controlos	167
7.2.5 Gráficos e outros recursos	169
7.2.6 <i>Themes</i> e <i>master pages</i>	170
 8 ELEMENTOS DE NAVEGAÇÃO	 171
8.1 Criar um mapa de <i>site</i>	172
8.2 <i>Site</i> exemplo	175
8.3 Identificar o mapa de <i>site</i>	177
8.4 Criar menus de <i>drop-down</i> e de <i>fly-out</i> – controlo MENU	181
8.4.1 Configurar o menu	182
8.5 Criar estruturas em árvore – controlo TREEVIEW	187
8.5.1 Configurar a TREEVIEW	187
8.6 Situar o utilizador no <i>site</i> – controlo SITEMAPPATH	192
 9 FORMULÁRIOS E VALIDAÇÕES	 195
9.1 Controlos HTML	196
9.2 Controlos servidor ASP.NET	197
9.2.1 Controlo TEXTBOX	199
9.2.2 <i>Radio buttons</i>	201
9.2.3 <i>Checkboxes</i>	205
9.2.4 <i>Drop-down lists</i>	207

9.2.5 Campos escondidos	211
9.2.6 <i>File uploads</i>	212
9.2.7 CALENDAR.....	212
9.2.8 Controlos de submissão e operação POSTBACK.....	215
9.3 Validação do preenchimento de formulários	217
9.3.1 Controlos de validação	218
9.3.2 Validação <i>server-side</i> e <i>client-side</i>	219
9.3.3 Propriedades dos controlos de validação	220
9.3.4 Validar o preenchimento de um campo	221
9.3.5 Validar intervalos de valores	222
9.3.6 Comparar valores	223
9.3.7 Validação avançada.....	225
9.3.8 Validation Summary	230
9.3.9 Grupos de validação	232
9.3.10 Funções de validação personalizadas	233
9.3.11 Criar código personalizado de validação	236
9.3.12 Propriedade ISPOSTBACK.....	236
9.3.13 Propriedade ISVALID.....	237
9.3.14 Confirmação do preenchimento do formulário	237
10 TRABALHAR COM CÓDIGO	241
10.1 Directivas ASP.NET	241
10.1.1 Código ASP.NET <i>design-time</i>	242
10.2 Modelos de código <i>server-side</i>	243
10.2.1 Código <i>server-side inline</i>	243
10.2.2 Código <i>server-side code-behind</i>	244
10.2.3 Mudar entre modelos de código <i>server-side</i>	244
10.2.4 Escolher um modelo de programação	245
10.3 Eventos	246
10.3.1 <i>Client-scripting</i>	248
10.3.2 O ciclo de vida de uma página.....	249
10.4 Trabalhar com código fonte.....	252
10.4.1 Controlar a janela de edição	253
10.4.2 Utilizar <i>outlining</i>	255
10.4.3 Navegar no código	258
10.4.4 Word Wrap	269
10.4.5 IntelliSense	270
10.4.6 Detectar erros de sintaxe	274
10.4.7 Localizar e substituir texto.....	276
10.4.8 Formatar o código fonte	287
10.4.9 Imprimir código	292
10.4.10 Outros comandos de ajuda à edição de código	294
10.4.11 <i>Object browser</i> e <i>Class view</i>	296
10.4.12 Code Snippets	301

11 A LINGUAGEM VISUAL BASIC	305
11.1 Variáveis e tipos de dados	305
11.1.1 Declaração de variáveis	306
11.1.2 Nomes de variáveis	306
11.1.3 Tipos de dados	307
11.1.4 Atribuição de valores a variáveis	309
11.1.5 Arrays	310
11.1.6 Enumerações.....	312
11.1.7 Conversões entre tipos de dados	313
11.2 Comentários.....	314
11.3 Operações sobre variáveis	314
11.3.1 Operações matemáticas.....	314
11.3.2 Operações sobre <i>strings</i>	316
11.4 Terminação de linhas.....	316
11.5 Estruturas de controlo	316
11.5.1 Lógica condicional	317
11.5.2 Bloco <i>If . . . Then . . . End If</i>	318
11.5.3 Bloco <i>Select...Case</i>	319
11.5.4 Ciclos.....	320
11.5.5 Esfera de acção.....	323
11.6 Funções, subrotinas e métodos	324
11.6.1 Parâmetros	325
11.7 Programação orientada para objectos	326
11.7.1 Uma classe simples.....	327
11.7.2 Criar um objecto	329
11.7.3 Membros <i>Shared</i> e <i>Instance</i>	330
11.7.4 Adicionar propriedades	330
11.7.5 Adicionar um método	333
11.7.6 Adicionar um <i>constructor</i>	333
11.7.7 Adicionar um evento	335
11.8 .NET <i>types</i>	338
11.8.1 <i>Namespaces</i> e <i>assemblies</i>	338
11.8.2 Utilizar <i>namespaces</i>	339
11.8.3 Importar <i>namespaces</i>	340
11.8.4 <i>Assemblies</i>	341
11.9 Manipulação de objectos	341
11.9.1 O tipo <i>STRING</i>	342
11.9.2 Os tipos <i>DATEIME</i> e <i>TIMESPAN</i>	344
11.9.3 O tipo <i>ARRAY</i>	347
12 EXECUTAR PÁGINAS E LIDAR COM ERROS	351
12.1 Testar páginas	351
12.1.1 Ver páginas individuais no <i>browser</i>	351
12.1.2 Executar o <i>site</i> sem depurar erros.....	353

12.1.3 Executar o <i>site</i> com depuração de erros	354
12.1.4 Definir uma página de início.....	358
12.2 Erros	360
12.2.1 Tipos de erros	360
12.2.2 Apresentação de erros no Visual Web Developer	361
12.2.3 Erros de <i>runtime</i>	365
12.3 Localizar e corrigir erros.....	367
12.3.1 Activar o <i>debugging</i>	367
12.3.2 <i>Breakpoints</i>	367
12.3.3 Painéis de <i>debugging</i>	372
12.3.4 Rastreamento (<i>tracing</i>)	376
12.4 Tratamento de erros.....	383
12.4.1 Excepções	384
12.4.2 Tratamento de erros em Visual Basic	385
12.4.3 Procedimentos <i>PAGE_ERROR</i> ou <i>ONERROR</i>	392
12.4.4 Página de erro específica	395
12.4.5 Tratar erros ao nível da aplicação.....	396
12.4.6 Página de erro <i>standard</i> ao nível da aplicação.....	398
12.4.7 Erros ao nível do servidor Web	403
12.4.8 Informação importante	405

13 CONFIGURAÇÃO DE APLICAÇÕES 407

13.1 Ficheiros de configuração	407
13.1.1 Segurança de acesso aos ficheiros	409
13.2 Web.config	409
13.2.1 Estrutura	409
13.2.2 Acesso programático	411
13.3 ASP.NET Web Site Administration Tool	412
13.4 Ficheiro <i>Global.asax</i>	414

14 BASES DE DADOS E SQL SERVER 417

14.1 SQL Server.....	417
14.2 SQL Server Express.....	418
14.2.1 Instalar o SQL Server Express.....	418
14.2.2 Ferramentas de gestão do SQL Express	424
14.2.3 SQL Server Management Studio Basic	425
14.3 Bases de dados SQL Server	428
14.3.1 Objectos da base de dados	429
14.3.2 Criar bases de dados e tabelas	430
14.3.3 Bases de dados exemplo.....	437
14.4 SQL	445
14.4.1 SQL no Management Studio Basic	445
14.4.2 Linguagem SQL	449
14.5 SQLCMD – Trabalhar com a linha de comandos	452

14.6 Visual Web Developer.....	458
14.6.1 Adicionar uma ligação a uma base de dados	458
14.6.2 Criar bases de dados.....	461
14.6.3 Alterar a estrutura da base de dados.....	462
14.6.4 Ver dados e modificar bases de dados.....	464

15 ADO.NET 469

15.1 Arquitectura.....	470
15.1.1 DATA PROVIDERS	473
15.2 Ligações a fontes de dados	474
15.2.1 Ficheiro <code>web.config</code>	476
15.2.2 Bases de dados <i>attached</i>	478
15.3 Execução de comandos SQL.....	479
15.3.1 Extrair um único valor	480
15.3.2 Extrair várias linhas	481
15.3.3 Preencher uma DROPDOWNLIST	484
15.4 Inserir, alterar dados e alterar a estrutura da base de dados.....	485
15.4.1 Adicionar novas linhas a uma tabela	486
15.4.2 Actualizar valores e eliminar linhas.....	488
15.5 DATAADAPTERS e DATASETS	494
15.5.1 Criar, preencher e percorrer objectos DATASET.....	495
15.5.2 Atribuir valores DATATABLE a controlos DROPDOWNLIST	497
15.5.3 Inserir, actualizar e eliminar linhas.....	500

16 ACESSO A DADOS COM CONTROLOS DATABOUND 505

16.1 Utilizar fontes de dados (<i>data sources</i>).....	506
16.1.1 Configurar fontes de dados (<i>data sources</i>).....	511
16.2 Apresentar informação da base de dados	517
16.2.1 Controlos DROPDOWNLIST e LISTBOX	519
16.2.2 Controlo GRIDVIEW	526
16.2.3 Controlo DETAILSVIEW.....	544
16.2.4 Controlo DATALIST	549
16.2.5 Controlo REPEATER.....	557
16.2.6 Controlo LISTVIEW.....	561

17 MANIPULAÇÃO DE DADOS 577

17.1 Fontes de dados	577
17.1.1 Métodos e eventos	580
17.2 Inserir dados	581
17.2.1 Controlo FORMVIEW	581
17.2.2 Outros controlos	586
17.2.3 Inserir dados utilizando formulários	587
17.3 Alteração de dados	589
17.3.1 Utilização de um controlo GRIDVIEW.....	589

17.3.2 Combinar GRIDVIEWS com FORMVIEWS	594
17.3.3 Confirmação da eliminação de registos de dados	598
17.3.4 Actualizar dados utilizando formulários.....	600
17.3.5 Controlo de acesso concorrential.....	600
17.4 Controlo LISTVIEW.....	603
17.5 Obter um valor IDENTITY	605
17.6 Upload de imagens	610
17.6.1 Armazenar ficheiros na base de dados.....	616
17.6.2 Downloads de ficheiros	618

18 GESTÃO DE ESTADOS 625

18.1 Control State	626
18.2 View state.....	627
18.2.1 Aceder ao view state.....	628
18.2.2 Desactivar o view state	629
18.2.3 Tornar o view state seguro.....	630
18.2.4 Exemplo	631
18.3 Transferir informação entre páginas.....	633
18.3.1 Cross-Page Posting.....	633
18.3.2 Querystring	639
18.4 Cookies	646
18.4.1 Esfera de acção dos cookies	650
18.4.2 Cookies e segurança	651
18.4.3 Determinar se um browser aceita cookies	652
18.5 Session State	652
18.5.1 Controlo de sessões	653
18.5.2 Utilizar o session state	653
18.5.3 Configuração do session state.....	654
18.5.4 Ficheiro Global.asax	658
18.6 Application state	659
18.6.1 Ficheiro Global.asax	661

19 CONTROLO DE ACESSOS 663

19.1 Configurar os parâmetros de segurança	663
19.1.1 Configurações adicionais.....	669
19.2 Adicionar controlos de login ao seu site	670
19.2.1 Controlo LOGIN	672
19.2.2 Controlo CREATEUSERWIZARD	673
19.2.3 Controlos LOGINSTATUS, LOGINNAME e LOGINVIEW	675

Prefácio

Introdução

O desenvolvimento de sítios Web, partilhado, entre outros, por informáticos, *designers*, arquitectos de informação e profissionais de marketing, tem sido uma das actividades que mais alterações tem sofrido ao longo dos seus poucos anos de existência. As páginas Web foram sempre evoluindo, passando de estáticas para interactivas e depois para dinâmicas, podendo ser personalizadas para cada visitante do *site* e suportadas por sofisticados acessos a bases de dados.

A Microsoft, embora tendo entrado tardiamente no negócio da Internet, cedo tentou recuperar o tempo perdido. Com o aparecimento das Active Server Pages (ASP), os programadores Windows passaram a ter uma tecnologia ideal para a criação de *sites* Web dinâmicos. Não contente com este sucesso, a Microsoft resolveu reinventar os ambientes de desenvolvimento de aplicações, lançando no início da década a .NET Framework.

O ASP.NET representa a aplicação desta *framework* ao desenvolvimento de páginas Web e de *web services*, sendo uma verdadeira revolução relativamente às Active Server Pages. E com cada nova versão do ASP.NET e da .NET Framework a Microsoft tem apresentado melhorias significativas.

Mas sem um ambiente integrado de desenvolvimento, seria muito complicada a criação de *sites* Web utilizando o ASP.NET. Para resolver esse problema a Microsoft disponibiliza o Visual Studio e o Visual Web Developer, verdadeiros IDE (Integrated Development Environment) que possuem todas as ferramentas que um *web developer* pode necessitar.

Finalmente, a utilização de bases de dados para a produção de *sites* Web dinâmicos ficou mais facilitada com a disponibilização do SQL Server Express, uma versão gratuita do SQL Server. Considerado um dos melhores sistemas de gestão de bases de dados existentes no mercado, o SQL Server, na sua versão Express, é uma solução poderosa para o desenvolvimento Web.

Público-alvo e pré-requisitos

Este livro é destinado a todos os *web developers*, existentes ou potenciais, que pretendam utilizar tecnologias recentes na criação de sítios Web interactivos e dinâmicos. É especialmente orientado para os *web developers* que utilizem tecnologias Microsoft.

Embora seja recomendada alguma experiência ao nível do desenho de páginas Web, da utilização de linguagens de programação e da manipulação de bases de dados, o livro pretende ser acessível a qualquer pessoa que tenha interesse no desenvolvimento de sítios Web.

Características principais do livro

Neste livro todos os capítulos tentam ter uma componente prática, mesmo aqueles em que são apresentadas, de forma mais ‘teórica’, as tecnologias que vão ser utilizadas.

Existem diversos projectos práticos, do tipo ‘exercício guiado’, em que o leitor pode reproduzir, passo-a-passo, a demonstração ou projecto a desenvolver.

Todos os capítulos encontram-se profusamente ilustrados. Este aspecto é considerado fundamental de forma a melhor guiar o leitor na reprodução dos passos a efectuar em cada projecto.

Os ficheiros dos exemplos práticos encontram-se disponíveis para os leitores. Veja na página 18 como pode ter acesso a esses ficheiros.

Convenções usadas

Este livro utiliza diversas convenções com vista a facilitar a assimilação da informação:

- Termos em inglês são apresentados, de uma forma geral, em *itálico*:
 - " ... que definem propriedades para estilos *inline* e para o atributo ..."
 - "... redimensionar o controlo recorrendo às respectivas *handles*."
- O código encontra-se formatado em Courier New:

```
Try
    paises.Items.Clear()
    While dr1.Read()
        paises.Items.Add(dr1.GetSQLString(0))
    End While
```

- As alterações a serem efectuadas no código são assinaladas a negrito:

```
países.Items.Clear()
países.Items.Add("-- Seleccione um país --")
While dr1.Read()
    países.Items.Add(rdrl.GetSQLString(0))
End While
```

- O acesso a opções de menus ou de caixas de diálogo é formatado em letras maiúsculas pequenas. O carácter ">" é utilizado para separar o menu da opção:
 "... o ideal é definir uma classe recorrendo à opção **FORMAT > NEW STYLE**, atribuindo ..."
- Combinações de teclas são identificadas através do carácter "+". Por exemplo, CTRL+TAB significa pressionar a tecla "Control" e, mantendo esta tecla premida, de seguida premir a tecla "Tab", soltando depois as duas teclas.

Visão geral e Organização

Este livro está organizado em dezanove capítulos:

- **Capítulo 1 – Desenvolvimento de *sítes* Web.** Conceitos sobre páginas estáticas, interactivas e dinâmicas, e sobre as tecnologias e metodologias utilizadas no desenvolvimento de páginas e *sítes* Web.
- **Capítulo 2 – Visual Studio e Visual Web Developer Express Edition.** Apresentação dos ambientes integrados de desenvolvimento que são disponibilizados pela Microsoft. Processo de instalação do Visual Web Developer Express Edition.
- **Capítulo 3 – Criar um *site* Web.** Descreve a definição e gestão de *sítes* no Visual Web Developer, a gestão de ficheiros e as pastas especiais.
- **Capítulo 4 – Ambiente de trabalho.** Apresenta o ambiente de trabalho do Visual Web Developer.
- **Capítulo 5 – Controlos Web e *layout* de páginas.** Aborda a utilização de controlos em páginas Web e o *layout* desses controlos nas páginas.
- **Capítulo 6 – CSS.** Demonstra como é que o Visual Web Developer pode ser utilizado para aplicar estilos CSS a páginas Web.
- **Capítulo 7 – Master Pages, Themes e Skins.** Explica a utilização de modelos/*templates* no ASP.NET, bem como outros processos para consolidar e centralizar a aparência das páginas de um *site*.

- **Capítulo 8 – Elementos de navegação.** Enumera os controlos que podem ser utilizados para a criação de elementos de navegação dos *sites*.
- **Capítulo 9 – Formulários e validações.** Apresenta os controlos que são utilizados na criação de formulários de introdução de dados, bem como os mecanismos de validação do preenchimento desses formulários.
- **Capítulo 10 – Trabalhar com código.** Explica as funcionalidades existentes no Visual Web Developer que facilitam a vida ao *web developer* na produção de código para páginas ASP.NET.
- **Capítulo 11 – A linguagem Visual Basic.** Introduz a linguagem Visual Basic, apresentando a respectiva sintaxe, a utilização de variáveis, as estruturas de controlo do fluxo de programa, os operadores e as funções, entre outros assuntos. Aborda-se também a programação orientada para objectos e os diversos objectos internos da .NET Framework.
- **Capítulo 12 – Executar páginas e lidar com erros.** Explica o processamento de páginas ASP.NET e o tratamento e depuração de erros.
- **Capítulo 13 – Configuração de aplicações.** Aborda os ficheiros de configuração existentes nos *sites* Web ASP.NET que permitem personalizar o respectivo comportamento e as funcionalidades disponibilizadas.
- **Capítulo 14 – Bases de dados e SQL Server.** Introduz o SQL Server Express, a sua instalação e administração, bem como a linguagem SQL que é utilizada para interagir com as bases de dados.
- **Capítulo 15 – ADO.NET.** Apresenta a tecnologia de acesso a dados da .NET Framework, que permite a extracção e manipulação de dados provenientes de bases de dados e de outras fontes de dados.
- **Capítulo 16 – Acesso a dados com controlos *databound*.** Descreve e exemplifica a utilização dos controlos especiais do ASP.NET para o acesso a fontes de dados.
- **Capítulo 17 – Manipulação de dados.** Descreve e exemplifica a utilização dos controlos especiais do ASP.NET para a inserção, alteração e eliminação de dados.
- **Capítulo 18 – Gestão de estados.** Aborda os mecanismos de gestão de estados que permitem controlar a interacção dos visitantes com um *site*.
- **Capítulo 19 – Controlo de acessos.** Faz uma abordagem introdutória à problemática de controlo de acessos a *sites* através de mecanismos de autenticação, recorrendo a controlos servidor ASP.NET.

Sobre o Autor

Pedro Remoaldo

Pedro Remoaldo é licenciado em Informática, formador e consultor em sistemas e tecnologias de informação com perto de uma década de experiência como docente no ensino superior.

Autor de 16 livros sobre sistemas operativos, software aplicacional, Internet, desenvolvimento Web e segurança, dos quais os 9 títulos mais recentes foram publicados pelo Centro Atlântico.

Os seus interesses são bastante abrangentes, mas atribui particular atenção às tecnologias Internet, aos sistemas de gestão de bases de dados e ao Business Intelligence.

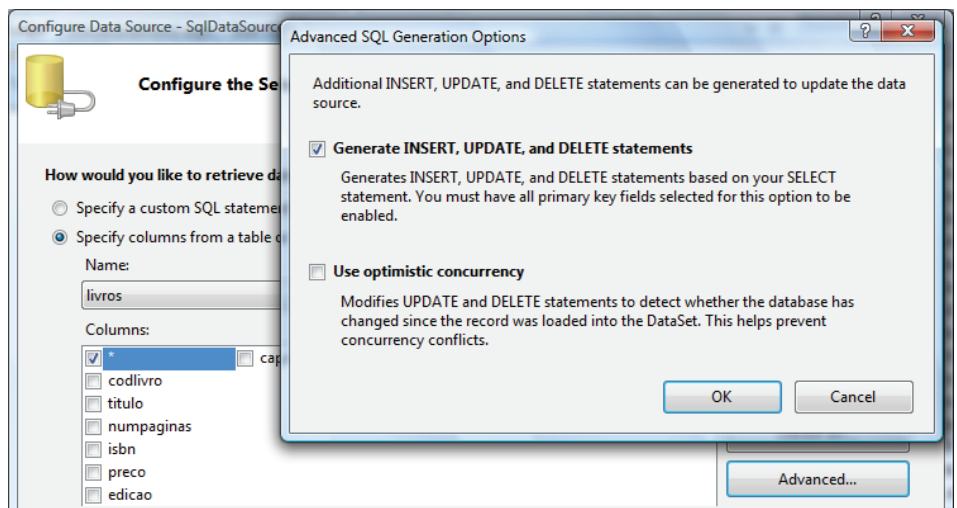
17 Manipulação de dados

A maior parte dos *rich data controls* do ASP.NET, como a GRIDVIEW ou a LISTVIEW, permite a manipulação de dados, isto é, a inserção, alterações e eliminação de registos de dados. Porém, nem todos os controlos permitem efectuar essas operações de uma forma fácil.

Estes controlos necessitam de um controlo SQLDATASOURCE associado para poderem manipular dados numa fonte de dados. São estes controlos que efectuem a interacção com a fonte de dados (SQL Server) quando solicitados pelos *rich data controls*.

17.1 Fontes de dados

Os controlos SQLDATASOURCE são fundamentais na inserção, alteração e eliminação de registos de dados. Mas têm de ser configurados nesse sentido, o que é efectuado clicando no botão ADVANCED, do ecrã CONFIGURE THE SELECT STATEMENT, e activando a opção GENERATE INSERT, UPDATE, AND DELETE STATEMENTS:



Neste método apenas apresentamos uma mensagem ao visitante da página a informá-lo sobre o código do livro produzido. Mas podemos utilizar agora este valor para efectuar outro tipo de operações.

17.6 Upload de imagens

A possibilidade de transferir ficheiros para um *site* (em inglês, *file upload*) é muito apreciada nomeadamente nos sistemas de gestão de conteúdos. É também utilizada em *sites* de *webmail*, como o Google Mail ou o HotMail, para enviar ficheiros anexados a mensagens.

Existem dois controlos no painel TOOLBOX que permitem este *upload* de ficheiros:

- Controlo INPUT (FILE) disponível na categoria HTML;
- Controlo FILEUPLOAD da categoria STANDARD;

Visualmente os dois controlos possuem a mesma aparência:



Um controlo de *upload* é assim constituído por uma caixa de texto e um botão. Estes elementos não possuem grande formatação e mesmo o texto que aparece no botão depende do *browser*, do sistema operativo e da língua utilizada pelo visitante.

Quando o visitante da página clica no botão BROWSE/PROCURAR aparece uma caixa de diálogo que lhe permite seleccionar o ficheiro que pretende transferir. O visitante também pode preencher manualmente a caixa de texto com a localização do ficheiro. Porém, o conteúdo da caixa de texto não pode ser preenchida programaticamente, nem por ASP.NET nem por JavaScript.

É possível utilizar o controlo HTML (INPUT (FILE)) para fazer o *upload*, mas para poder manipular o controlo em código teremos de o converter para um controlo de servidor, acrescentando o seguinte atributo:

```
<input id="File1" type="file" runat="server" />
```

Mas o ideal é utilizar o controlo servidor FILEUPLOAD. De qualquer forma este controlo é convertido no controlo INPUT (FILE) de HTML quando a página é enviada para o visitante.

A presença deste controlo numa página Web faz com que o ASP.NET acrescente o atributo `enctype` ao elemento `form`:

```
<form name="form1" method="post" action="Upload.aspx"
id="form1" enctype="multipart/form-data">
```

Este atributo `enctype` possui, por omissão, o valor `application/x-www-form-urlencoded`, que é um tipo de codificação ineficiente para enviar grandes quantidades de dados binários, ou texto que contenha caracteres não-ASCII. Por isso é que se utiliza o tipo de codificação `multipart/form-data` que contém uma série de partes (daí o nome), cada uma das quais se refere a um controlo do formulário.

Quando o formulário é submetido, o servidor recebe o(s) ficheiro(s) e cabe ao código ASP.NET examinar, ignorar ou guardar esse(s) ficheiro(s). Pode-se guardar os ficheiros em disco ou numa base de dados.

No servidor, para obtermos informação sobre o ficheiro submetido temos de utilizar a classe `FILEUPLOAD` e as seguintes propriedades:

- `FILEBYTES` – conteúdo do ficheiro como um *array* de *bytes*.
- `FILECONTENT` – conteúdo do ficheiro como uma *stream*, permitindo assim a sua leitura.
- `FILENAME` – nome original do ficheiro.
- `HASFILE` – devolve `TRUE` se um ficheiro tiver sido transferido, ou `FALSE` caso contrário.
- `POSTEDFILE` – permite aceder a um objecto `HTTPPOSTEDFILE` que possui informação sobre um ficheiro transferido.

Utilizando a propriedade `POSTEDFILE` temos acesso a propriedades adicionais sobre o ficheiro transferido:

- `CONTENTLENGTH` – tamanho do ficheiro em *bytes*.
- `CONTENTTYPE` – *MIME type* do ficheiro transferido, isto é, o tipo de ficheiro (imagem, documento, etc.)
- `FILENAME` – *fully qualified name* do ficheiro, isto é, o nome do ficheiro bem como a sua localização completa no computador do visitante, do tipo `c:\ficheiros\imagens\ca.gif`
- `INPUTSTREAM` – Permite obter o conteúdo do ficheiro como uma *stream*, permitindo assim a sua leitura.

A classe `HTTPFILECOLLECTION` disponibiliza acesso, como uma colecção de ficheiros, a todos os ficheiros que são transferidos. Caso a página permita a transferência de mais de um ficheiro (recorrendo a vários controlos `FILEUPLOAD`), então esta colecção pode ser percorrida para obter informação sobre cada um dos ficheiros. Por exemplo:

```
Dim listaFicheiros As HttpFileCollection = Request.Files
For Each ficheiro As String In listaFicheiros.AllKeys
    ' Se o ficheiro não tiver tamanho 0
    If (listaFicheiros(ficheiro).ContentLength > 0) Then
        ' Fazer algo com o ficheiro
```

```

        ' Obtém-se o nome do ficheiro (incluindo pasta)
        ' utilizando listaFicheiros(ficheiro).FileName
        ...
    End If
Next

```

Normalmente a operação mais efectuada no servidor é armazenar o ficheiro numa pasta específica. Para isso utiliza-se o método `SAVEAS`:

```
ElementoFileUpload.SaveAs("pasta/nomeficheiro")
```

Mas antes de invocar este método é conveniente utilizar a propriedade `HASFILE` para verificar se o controlo `FILEUPLOAD` possui um ficheiro para transferir com um tamanho superior a 0KB. Caso isso não aconteça, podemos ignorar a transferência ou apresentar ao visitante uma mensagem informando-o do ocorrido.

Este código é normalmente colocado no *event handler* associado ao evento `CLICK` do botão que submete o formulário:

```

Protected Sub btInserir_Click(ByVal sender As Object, ByVal e
    As System.EventArgs) Handles btInserir.Click

    If (fuFicheiro.HasFile) Then
        Dim nome_ficheiro As String = Server.MapPath("fich") &
        "/" & fuFicheiro.FileName
        fuFicheiro.SaveAs(nome_ficheiro)
    Else
        lblMensagem.Text = "Não foi especificado nenhum
        ficheiro para ser transferido."
    End If

End Sub

```

Um dos problemas que podemos aqui encontrar é o facto de na pasta `fich` poder já existir um ficheiro com o mesmo nome que pretendemos atribuir. Uma forma de resolver este problema é gerar um nome baseado na identificação do utilizador, na hora e data corrente ou em valores aleatórios (por exemplo, utilizando a estrutura `GUID`).

Por exemplo:

```

If (fuFicheiro.HasFile) Then
    Dim nomeTemporario As String
    Dim pasta As String = Server.MapPath("fich") & "/"
    Dim nome_ficheiro As String = fuFicheiro.FileName
    Dim pastaFicheiro As String = pasta & nome_ficheiro
    If (System.IO.File.Exists(pastaFicheiro)) Then
        Dim contador As Integer = 1
        While (System.IO.File.Exists(pastaFicheiro))
            nomeTemporario = contador.ToString() &
            nome_ficheiro
            pastaFicheiro = pasta + nomeTemporario
            contador = contador + 1
        End While
    End If
End Sub

```

```

        pastaFicheiro = pasta & nomeTemporario
    End If
    fuFicheiro.SaveAs(pastaFicheiro)
Else
    ...

```

Neste caso utilizamos um valor inteiro (1, 2, 3, ...) que colocamos no início do nome original. Teremos assim nomes do tipo 1computador.gif, 2computador.gif, etc. O método `SYSTEM.IO.FILE.EXISTS` permite determinar se um dado ficheiro existe numa pasta do disco.

Outra possibilidade a testar é apenas aceitar determinados tipos de ficheiros. No exemplo seguinte criamos uma função que apenas permite a transferência de imagens:

```

Private Function TipoFicheiro(ByVal nomeficheiro As String) As Boolean
    Dim extensao As String =
        System.IO.Path.GetExtension(nomeficheiro)
    Select Case extensao.ToLower()
        Case ".gif", ".png", ".jpg", ".jpeg"
            Return True
        Case Else
            lblMensagem.Text = "O ficheiro transferido não é uma imagem!"
            Return False
    End Select
End Function

```

Para invocar a função:

```

Dim pastaFicheiro as String = pasta & nome_ficheiro
Dim nome_ficheiro as String = fuFicheiro.FileName
If (TipoFicheiro(nome_ficheiro)) Then
    ...
    ficheiro.SaveAs(pastaFicheiro)

```

Em vez de extrairmos a extensão do ficheiro o mais adequado é obter o respectivo MIME *type* recorrendo à propriedade `CONTENTTYPE`:

```

Private Function TipoFicheiro(ByVal tipo As String) As Boolean
    Select Case tipo
        Case "image/gif", "image/png", "image/x-png",
            "image/jpeg", "image/pjpeg"
            Return True
        Case Else
            lblMensagem.Text = "O ficheiro transferido não é uma imagem!"
            Return False
    End Select
End Function

...
Dim pastaFicheiro as String = pasta & nome_ficheiro
Dim nome_ficheiro as String = ficheiro.FileName

```

```
Dim MIMEType as String = ficheiro.PostedFile.ContentType;  
If (TipoFicheiro(MIMEType.ToString())) Then  
    ...  
    ficheiro.SaveAs(pastaFicheiro)  
Else  
    lblMensagem="O ficheiro transferido não é do tipo  
    autorizado!"  
End If
```

Na função `TipoFicheiro` utilizam-se os MIME *type* `image/pjpeg` e `image/x-png`, além dos tradicionais MIME *types* associados às imagens permitidas em *sites*, porque é assim que o Internet Explorer identifica as imagens do tipo JPEG e PNG, respectivamente

Como é óbvio, convém também efectuar este teste utilizando JavaScript, evitando assim uma ‘viagem’ ao servidor para processar o código ASP.NET.

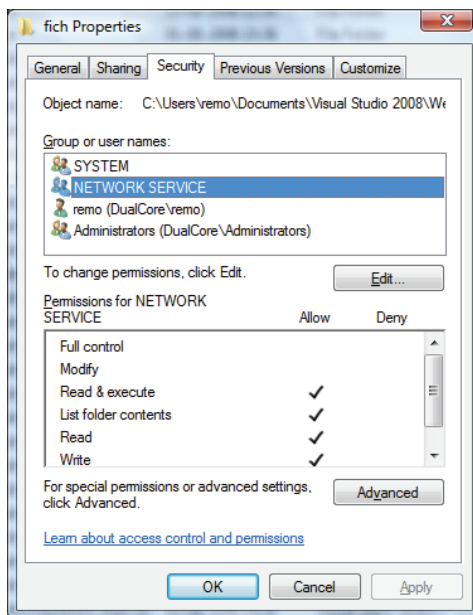
Dado que se tentarmos guardar um ficheiro não existente será desencadeada uma excepção `HTTPException`, o ideal é colocar a invocação do método `SAVEAS` dentro de uma estrutura `TRY..CATCH..FINALLY`, mesmo que utilizemos todas as validações apresentadas anteriormente:

```
Try  
    fuFicheiro.SaveAs(pastaFicheiro)  
Catch Err As Exception  
    lblMensagem.Text = "Ocorreu um erro na transferência do  
    ficheiro!"  
End Try
```

Podemos guardar o ficheiro onde quisermos. Porém, para evitar obter erros `access denied` quando se tenta guardar ficheiros directamente numa pasta da sua aplicação Web, temos de verificar as permissões.

Assegure-se que o directório virtual no IIS possui permissões de leitura e escrita. Para guardar um ficheiro no *file system* do servidor, a conta Windows associada com a página ASP.NET tem de possuir permissões suficientes. No caso do Windows XP (versão 5 do IIS) é a conta `ASPNET`. Nas versões posteriores do IIS (Windows Vista e Windows Server 2003/2008) é a conta `NETWORK SERVICE`.

No Windows Explorer, clique com o botão direito do rato na pasta onde pretende guardar os ficheiros, seleccione a opção `PROPERTIES` e depois o separador `SECURITY`. Clique no botão `EDIT`, depois no botão `ADD`, escreva `NETWORK SERVICE` (ou `ASPNET`, conforme o sistema operativo) na caixa `ENTER THE OBJECT NAMES TO SELECT`, clique em `OK`, e depois atribua a permissão `WRITE` na coluna `ALLOW` da lista `PERMISSIONS FOR NETWORK SERVICE`.



Por omissão, o ASP.NET rejeitará um pedido de *upload* de um ficheiro que seja maior que cerca de 4MB (4096KB), desencadeando uma *exception*. Pode-se alterar este valor máximo através da modificação do atributo `maxRequestLength` do elemento `httpRuntime` no ficheiro `web.config`. No exemplo seguinte aumentámos o tamanho máximo para 8MB. Utiliza-se 8192 porque o valor é especificado em kilobytes (KB):

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    ...
    <httpRuntime maxRequestLength="8192"/>
    ...
  </system.web>
</configuration>
```

Na realidade, o atributo `maxRequestLength` define o tamanho máximo da informação que pode ser enviada por um formulário para o servidor, incluindo a informação de todos os campos do formulário. Por isso, o tamanho máximo de um ficheiro transferido será ligeiramente inferior ao especificado pelo atributo `maxRequestLength`.

Pode-se também utilizar outro atributo, `requestLengthDiskThreshold`, para definir como é que o envio do conteúdo do formulário (e o ficheiro transferido) é *buffered* para o sistema de ficheiros, para evitar a utilização da memória para o armazenamento temporário do ficheiro e da informação transferida.

Quando o tamanho de um ficheiro/informação transferido ultrapassa o valor do atributo `requestLengthDiskThreshold`, o resto do ficheiro é *buffered* para

o *file system* para a pasta Temporary ASP.NET Files (que se encontra na pasta C:\Windows\Microsoft.NET\Framework\v2.0.50727).

Por omissão, a .NET Framework está configurada para fazer o *buffering* de ficheiros (mais concretamente de *posts* de formulários) maiores que 80KB. Se pretender alterar este valor terá de ter em atenção que deverá ser inferior ao valor do atributo `maxRequestLength`:

```
<httpRuntime maxRequestLength="8192"  
    requestLengthDiskThreshold="200" />
```

Se possuir um valor do atributo `maxRequestLength` demasiado alto pode sofrer um ataque *denial-of-service* através do envio de mensagens com grandes ficheiros, tornando o seu servidor muito lento ou mesmo ‘parando-o’.

17.6.1 Armazenar ficheiros na base de dados

Embora o armazenamento de ficheiros no disco do servidor seja muito comum, o *web developer* pode desejar guardar os ficheiros na base de dados.

Uma das vantagens de armazenar ficheiros em disco é que o processo é fácil, bastando invocar o método `SAVEAS` como vimos anteriormente. Outra vantagem é a facilidade de efectuar cópias de segurança de uma pasta. Além disso, podem-se efectuar cópias de segurança incrementais.

Como desvantagens, a inexistência de uma relação directa entre os ficheiros e os registos na tabela da base de dados. Isto pode trazer problemas quando se eliminam registos e nos esquecemos de eliminar o ficheiro respectivo. A principal desvantagem são as permissões de acesso e escrita em disco, que podem ser problemáticas quando se aloja o *site* num ISP.

Relativamente ao armazenamento de ficheiros em bases de dados, evita problemas com permissões de pastas ao nível do *file system* além de que possibilita uma cópia de segurança mais fácil dos dados do *site*, dado que toda a informação se encontra num único sítio, isto é, na base de dados.

Como desvantagens, a performance é mais lenta, mas depende do tipo de ficheiros armazenados, do servidor e de outros factores. O acesso aos ficheiros também é uma desvantagem dado que não podemos aplicar um processo (como escalar ou rodar imagens) a todos os ficheiros, sem primeiro extraí-los da base de dados.

Quando armazena ficheiros na base de dados é vulgar armazenar a seguinte informação sobre o ficheiro:

- conteúdo do ficheiro (tipo de dados `image` ou `varbinary(max)`);
- tipo de ficheiro (tipo de dados `varchar`);
- comprimento em *bytes* (tipo de dados `int`);

- nome que o ficheiro possuía quando foi *uploaded* (tipo de dados `varchar`). Pode ser utilizado quando um visitante solicitar posteriormente o *download* do ficheiro.

De forma a exemplificarmos a inserção de ficheiros em bases de dados SQL, vamos alterar a estrutura da nossa tabela `livros`, acrescentando um novo campo, `excerto`, do tipo `varbinary(MAX)`, que vai conter um ficheiro PDF com um excerto de cada livro:

	Column Name	Data Type	Allow Nulls
?	codlivro	smallint	<input type="checkbox"/>
	titulo	varchar(70)	<input checked="" type="checkbox"/>
	numpaginas	smallint	<input checked="" type="checkbox"/>
	isbn	char(17)	<input checked="" type="checkbox"/>
	preco	smallmoney	<input checked="" type="checkbox"/>
	edicao	tinyint	<input checked="" type="checkbox"/>
	capa	varchar(45)	<input checked="" type="checkbox"/>
▶	excerto	varbinary(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

O nosso formulário de inserção de livros tem agora o seguinte aspecto:

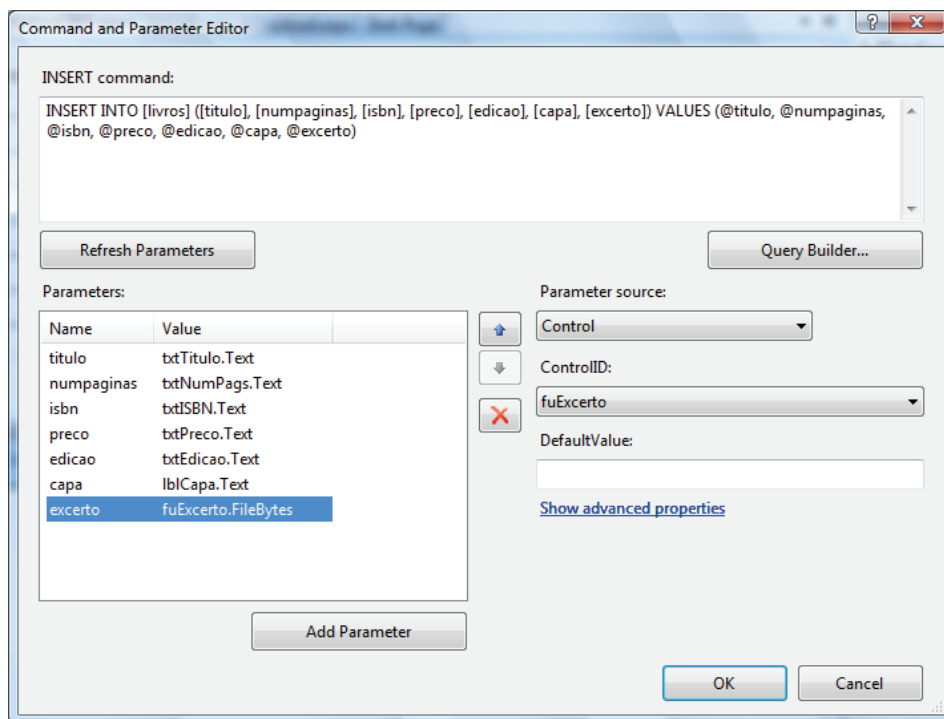
div

Inserção de livros

Título:
 N° de pág.:
 ISBN:
 Preço:
 Edição:
 Capa:
 Excerto:

Repare que vamos utilizar dois controlos do tipo `FILEUPLOAD`. Não existe limite relativamente ao número de controlos deste tipo que se podem utilizar em páginas Web, se bem que quantos mais controlos utilizar mais lento será o processamento da página.

No controlo `SQLDATASOURCE` associado a este formulário vamos efectuar a seguinte alteração ao comando SQL:



O valor a inserir no campo `excerto` da tabela `livros` provém da propriedade `FILEBYTES` do controlo `FILEUPLOAD`. Esta propriedade permite obter o conteúdo do ficheiro como um *array* de *bytes*. No código aparece:

```
<asp:ControlParameter Name="excerto" ControlID="fuExcerto"
  PropertyName="FileBytes" />
```

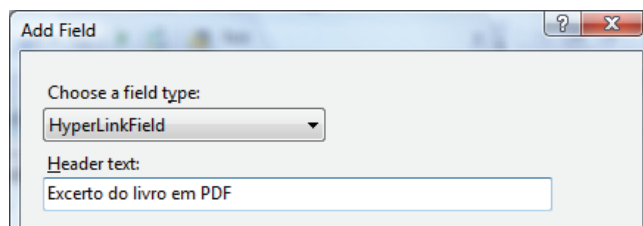
Basta invocar o método `INSERT` do controlo `SQLDATASOURCE` para inserir os dados na base de dados incluindo o ficheiro.

17.6.2 Downloads de ficheiros

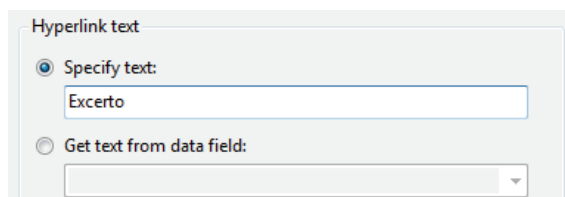
Agora que já temos os ficheiros armazenados na base de dados é necessário um procedimento para permitir ao visitante aceder a esses ficheiros.

Vamos arrastar a tabela `livros` do painel `DATABASE EXPLORER` para uma nova página. Como já foi anteriormente explicado, esta operação criará um controlo `GRIDVIEW` e um controlo `SQLDATASOURCE`. Este controlo `SQLDATASOURCE` extrairá todas as linhas e todas as colunas da tabela `livros`, excepto a coluna `excerto` dado que é do tipo `varbinary`.

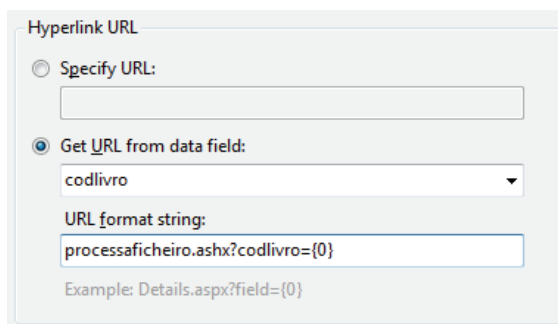
Vamos utilizar um *link* para possibilitar a transferência do excerto do livro em formato PDF. No controlo `GRIDVIEW` seleccione a opção `ADD NEW COLUMN` do respectivo menu de tarefas. Na caixa de diálogo `ADD FIELD` seleccione `HYPERLINKFIELD` da lista `CHOOSE A FIELD TYPE`:



Se quiser pode introduzir, na caixa HEADER TEXT, o texto que deve aparecer no cabeçalho da coluna na GRIDVIEW. A seguir, na área HYPERLINK TEXT introduza o texto que pretende que apareça no *link*:



Finalmente, na área HYPERLINK URL, seleccione a opção GET URL FROM DATA FIELD e depois seleccione a coluna/campo `codlivro` na lista:

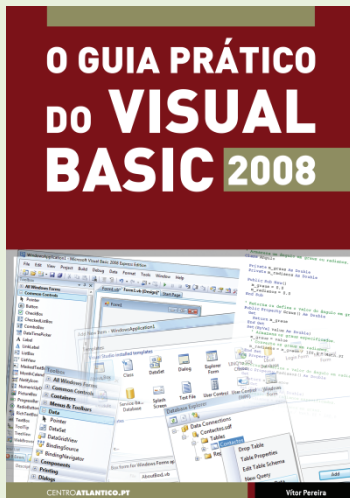


Na caixa URL FORMAT STRING introduzimos:

```
processaficheiro.ashx?codlivro={0}
```

Isto significa que sempre que se clicar no *link* `Excerto` será invocada a página `processaficheiro.ashx`, e é passada a essa página, como parâmetro, o valor da coluna/campo `codlivro` do livro seleccionado. Neste caso, o nome do parâmetro (`codlivro`) é idêntico ao nome da coluna/campo da tabela `livros`, mas isso não é obrigatório.

Para enviar para o visitante de um *site* um ficheiro que está armazenado na base de dados, temos de recorrer a um ficheiro HTTP Handler (que possui a extensão `.ashx`). Para criar um ficheiro deste tipo seleccione a opção FILE > NEW FILE (ou ADD NEW ITEM no menu de contexto do *site* no painel SOLUTION EXPLORER) e, na caixa de diálogo ADD NEW ITEM, seleccione o *template*



Este livro aborda os vários aspectos necessários à criação de sítios Web dinâmicos, recorrendo à tecnologia ASP.NET e ao sistema de gestão de bases de dados SQL Server, ambos produtos da Microsoft.

Para tal recorre-se ao Visual Web Developer, um ambiente integrado de desenvolvimento disponibilizado gratuitamente pela Microsoft – as diversas funcionalidades existentes neste programa permitem a criação de projectos sofisticados, incluindo *websites* de comércio electrónico.

O livro começa com uma apresentação das tecnologias e metodologias utilizadas no desenvolvimento de páginas e *sites* Web. Em seguida apresenta-se o Visual Web Developer e a sua utilização para o desenho de páginas utilizando controlos servidor ASP.NET, CSS e Master Pages.

A linguagem Visual Basic e a sua utilização em ASP.NET são outros dos assuntos abordados, bem como o trabalho com código e o tratamento de erros por se tratarem de aspectos fundamentais no desenvolvimento de páginas Web.

O livro atribui um especial destaque à utilização de bases de dados na criação de páginas Web dinâmicas – o ASP.NET utiliza preferencialmente o SQL Server como fonte de dados, recorrendo ao ADO.NET para interagir com este sistema de gestão de bases de dados.

Também são descritos os mecanismos de gestão de estados que permitem controlar a interacção dos visitantes com um sítio Web, bem como a problemática do controlo de acessos através de mecanismos de autenticação.

O livro apresenta diversos exercícios ‘guiados’ que permitem, mesmo ao utilizador mais desconhecedor destas tecnologias, criar páginas assaz avançadas e utilizar os conhecimentos adquiridos para resolver problemas nos seus projectos de desenvolvimento Web.

Pedro Remoaldo é licenciado em Informática, formador e consultor em sistemas e tecnologias de informação com cerca de uma década de experiência como docente no ensino superior.

Autor de 16 livros sobre sistemas operativos, software aplicacional, Internet, desenvolvimento Web e segurança, dos quais os 9 títulos mais recentes foram publicados pelo Centro Atlântico.

Os seus interesses são bastante abrangentes, mas atribui particular atenção às tecnologias Internet, aos sistemas de gestão de bases de dados e ao Business Intelligence.

