

## *Tarefa Orientada 3*

### *Aplic. Manutenção de Produtos – Text Box*

#### Objectivos:

- Utilização de uma fonte de dados.
- Criação de controlos *Text Box* ligados a uma fonte de dados.
- Criação de classes.

Se uma tabela contém um grande número de atributos, pode ser conveniente, para aceder aos seus dados, utilizar controlos *Text Box*, em vez de um controlo *DataGridView*.

Nesta tarefa orientada vamos implementar uma aplicação de manutenção de produtos semelhante à aplicação desenvolvida na tarefa orientada anterior. Contudo, para aceder aos dados armazenados na base de dados remota, vamos utilizar controlos *Text Box*, em vez de um controlo *DataGridView*.

## Utilização de uma fonte de dados

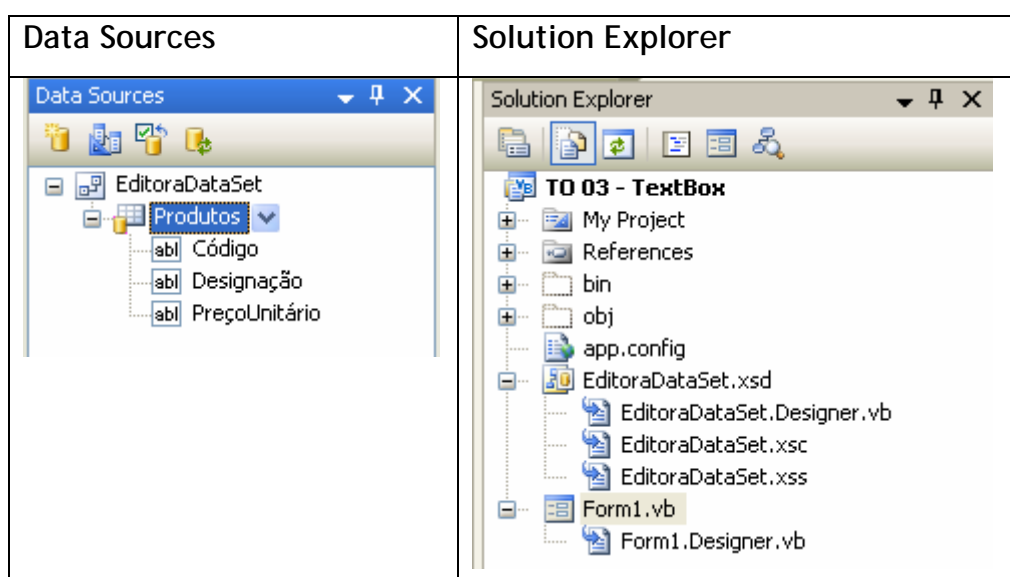
Neste projecto, vamos continuar a utilizar o *DataSource Configuration Wizard* para criar uma fonte de dados.

- 1 Execute os passos 1 a 5 da tarefa orientada 1, sobre Visual Basic 2005, para criar um novo projecto para uma aplicação Windows.
- 2 Criação de uma fonte de dados, a partir de uma base de dados no Sql Server, recorrendo ao assistente de configuração.
  - 2.1 Siga os passos 2.1 a 2.15 da tarefa orientada anterior para criar uma fonte de dados a partir da tabela *Produtos* da base de dados *Editora*.

Deste modo, irá ser criada uma *data table* com o nome *Produtos* e que vai conter os campos *código*, *Designação* e *PreçoUnitário*. Note que, tal como na tarefa orientada anterior, não foi seleccionado o campo *Quantidade*, pois trata-se de um atributo que tem associado um valor predefinido (*Default*). Este valor não é incluído na definição do *DataSet*. Assim, ao omitir este campo na definição do *DataSet*, quando é inserido um registo, é tomado o valor predefinido na base de dados para este atributo.

Caso tivesse incluído o atributo *Quantidade* neste passo, deveria depois especificar o valor predefinido na propriedade *DefaultValue* desse campo. Para tal, depois de terminar este passo, teria que aceder ao *DataSet Designer*, seleccionar o campo *Quantidade*, aceder à janela de propriedades e especificar um valor para a propriedade *DefaultValue*.

Terminados os passos do assistente, a nova fonte de dados é mostrada na janela *DataSources*. Na janela *Solution Explorer*, é também incluído um ficheiro com o nome que foi atribuído ao *DataSet* e com a extensão *xsd*. Este ficheiro contém o esquema para a classe *DataSet* gerada pelo *wizard*, isto é, define a estrutura do *DataSet*.



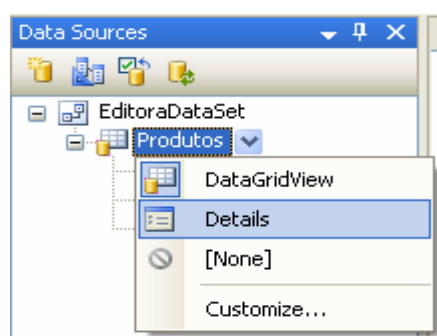
## Criação de controlos *Text Box* ligados a uma fonte de dados

Depois de ter criado um uma fonte de dados, pode utilizar controlos *Text Box* para visualizar, inserir, actualizar e eliminar dados.

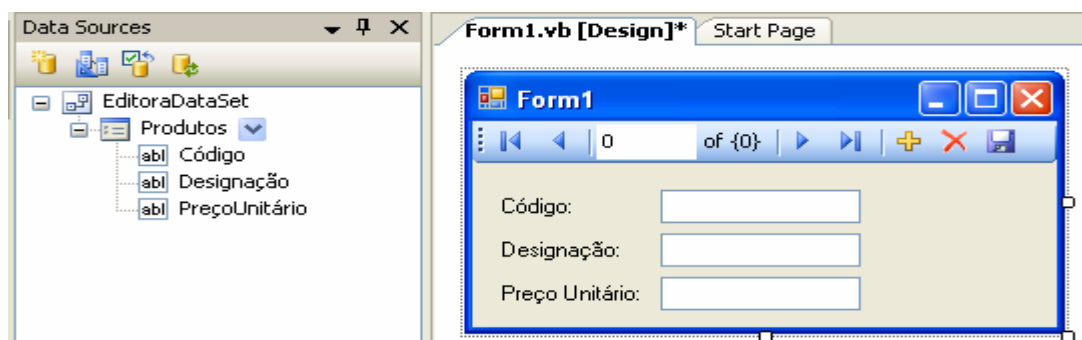
De modo predefinido, se arrastar uma tabela da janela *Data Source* para um formulário, o Visual Studio adiciona um controlo *DataGridView* ao formulário e liga-o à tabela. Todavia, se pretender, também é possível criar outros controlos ligados, tais como caixas de texto (*TextBox*), caixas de combinação (*ComboBox*) e caixas de listagem (*ListBox*).

3 Criação de controlos *TextBox*, a partir de uma fonte de dados, recorrendo à janela *Data Sources*.

3.1 Selecione, na janela *Data Sources*, a opção *Details* da caixa de listagem associada à tabela *Produtos*.

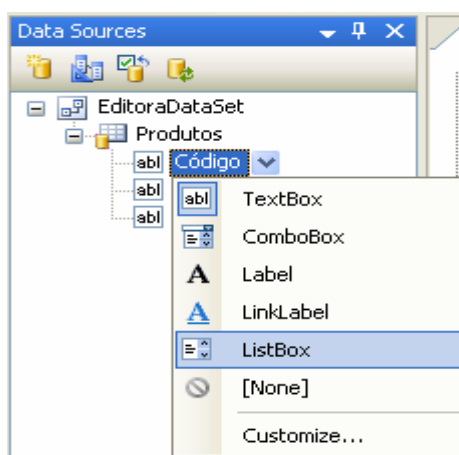


3.2 Arraste a tabela *Produtos* da janela *Data Sources* para o formulário do projecto.



Como resultado do passo anterior, além dos objectos *TextBox* e *Label*, criados para cada coluna da tabela, foram também criados os objectos *EditoraDataSet*, *ProdutosTableAdapter*, *ProdutosBindingSource* e *ProdutosBindingNavigator*. Adicionalmente, o Visual Studio liga as caixas de texto às respectivas colunas da tabela.

Como pode observar, por omissão, os campos com atributos alfanuméricos e numéricos são associados a controlos *TextBox*. Contudo, pode alterar este modo predefinido se seleccionar, na janela *Data Sources*, os atributos e, depois seleccionar a opção pretendida na caixa de listagem associada. Por exemplo, poderia criar um controlo *ListBox* para o campo *Código* da tabela *Produtos*, do seguinte modo.



Adicionalmente, o Visual Studio gera também o seguinte código. Se necessário, **como iremos ver mais à frente**, este código pode ser alterado.

```
Public Class Form1
    Private Sub ProdutosBindingNavigatorSaveItem_Click(ByVal sender As _
        System.Object, ByVal e As System.EventArgs) Handles _
        ProdutosBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.ProdutosBindingSource.EndEdit()
        Me.ProdutosTableAdapter.Update(Me.EditoraDataSet.Produtos)
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
        System.EventArgs) Handles MyBase.Load
        Me.ProdutosTableAdapter.Fill(Me.EditoraDataSet.Produtos)
    End Sub
End Class
```

3.3 Analise o código gerado e as propriedades dos objectos criados.

Se tiver alguma dificuldade a perceber este código, consulte a tarefa orientada anterior.

Note, por exemplo, que é atribuído à propriedade *Text*, da colecção *DataBindings*, de cada objecto *TextBox* o nome da coluna do objecto ao qual o controlo *TextBox* está ligado.

3.4 Verifique que a caixa de texto *CódigoTextBox* está ligada à coluna *Código* do objecto *ProdutosBindingSource*.

Se pretender criar um formulário só de leitura, pode editar a barra de ferramentas definida pelo objecto *BindingNavigator*, de modo a remover os seus botões *Add*, *Delete* e *Save*. Mais tarde, voltaremos a este assunto.