

## *Tarefa Orientada 4*

### *Aplic. Manutenção de Produtos - ComboBox*

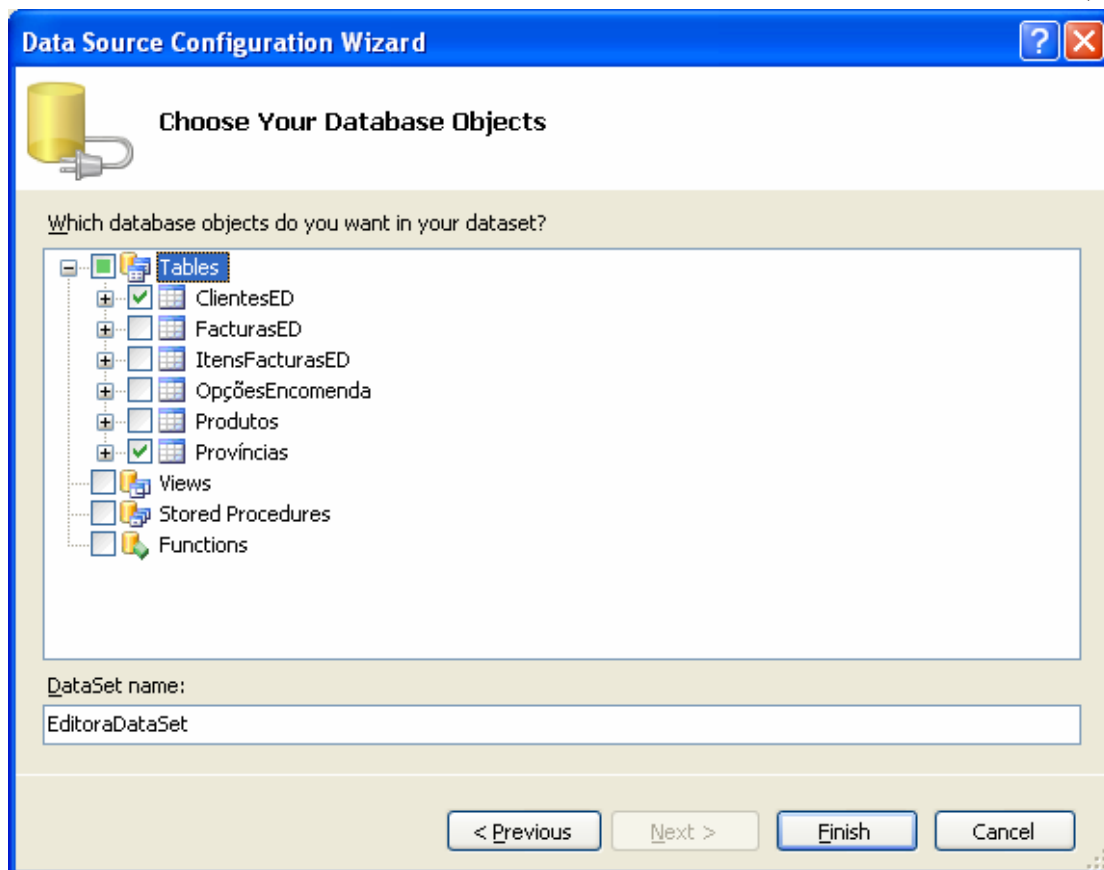
#### Objectivos:

- Utilização de uma fonte de dados.
- Criação de controlos Combobox ligados a uma fonte de dados.
- Utilização de parâmetros em consultas SQL.

#### 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 2.12 da tarefa orientada 2 para criar uma fonte de dados a partir das tabelas *ClientesED* e *Províncias* da base de dados *Editora*.
  - 2.2 Agora escolha os objectos da base de dados para a fonte de dados que vão ser incluídos no *DataSet*. Expanda o nó associado à caixa de verificação *Tables* e seleccione as tabelas *ClientesED* e *Províncias*.

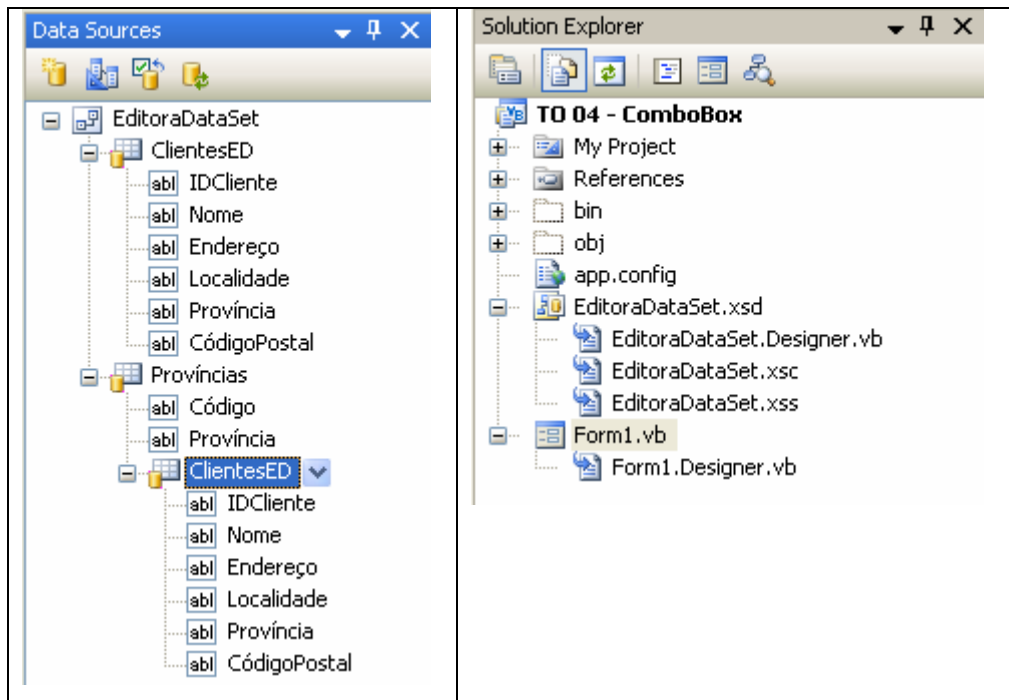


2.3 Neste passo, pode ainda especificar o nome para o *DataSet* que vai ser criado. De modo predefinido, este nome é constituído pelo nome da base de dados seguido da palavra *DataSet*

2.4 Pressione o botão *Finish*.

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*.

Data Sources	Solution Explorer
--------------	-------------------



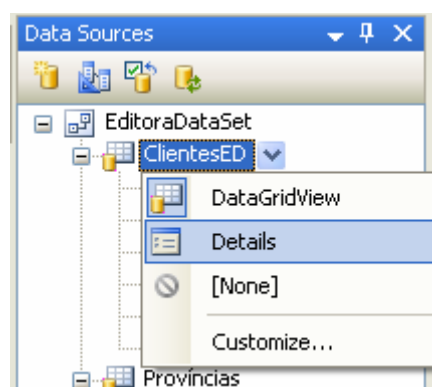
## Criação de controlos *Text Box* e *ComboBox* ligados a uma fonte de dados

Depois de ter criado um uma fonte de dados, pode utilizar controlos *TextBox* e *ComboBox* 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 á possível criar outros controlos ligados, tais como caixas de texto (*TextBox*), caixas de combinação (*ComboBox*) e caixas de listagem (*ListBox*).

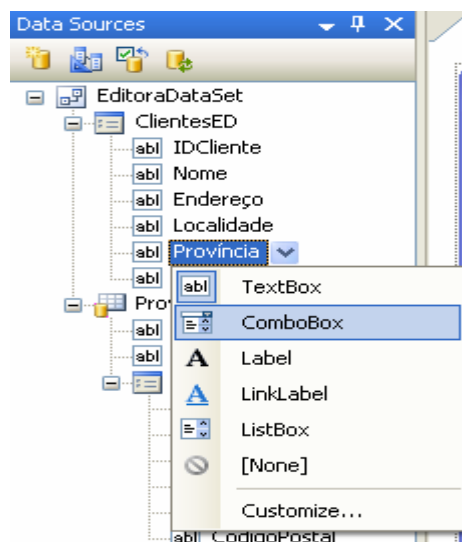
3 Criação de controlos *TextBox* e *ComboBox*, a partir de uma fonte de dados, recorrendo à janela *Data Sources*.

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



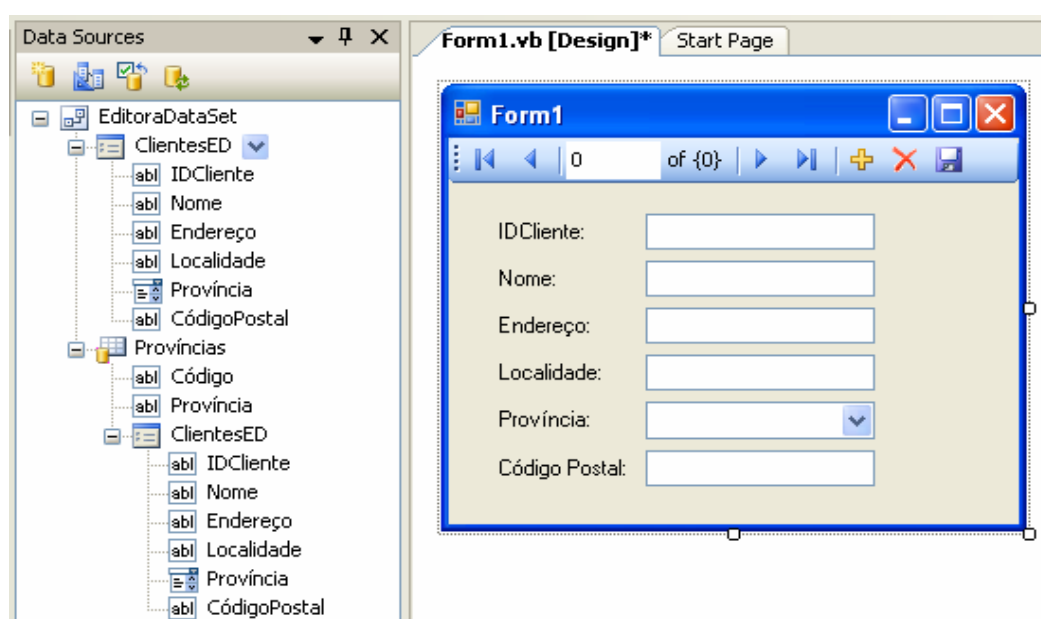
Deste modo, ao arrastar a tabela *ClientesED* para o formulário iriam ser criados objectos *Label* e *TextBox* ligados a cada uma das colunas da tabela.

3.2 Selecciona, na janela *Data Sources*, a opção *ComboBox* da caixa de listagem associada à coluna *Província* da tabela *ClientesED*.



Deste modo, ao arrastar a tabela *ClientesED* para o formulário irá ser criado um controlo *ComboBox* ligado à coluna *Província* da tabela *ClientesED* e irão ser criados objectos *Label* e *TextBox* ligados a cada uma das restantes colunas da tabela.

3.3 Arraste a tabela *ClientesED* 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*, *clientesEDTableAdapter*, *clientesEDsBindingSource* e *clientesEDBindingNavigator*. Adicionalmente, o Visual Studio liga as caixas de texto e a caixa de combinação às respectivas colunas da tabela.

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 ClientesEDBindingNavigatorSaveItem_Click(ByVal sender _
        As System.Object, ByVal e As System.EventArgs)
        Me.Validate()
        Me.ClientesEDBindingSource.EndEdit()
        Me.ClientesEDTableAdapter.Update(Me.EditoraDataSet.ClientesED)
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As _
        System.EventArgs) Handles MyBase.Load
        Me.ClientesEDTableAdapter.Fill(Me.EditoraDataSet.ClientesED)
    End Sub

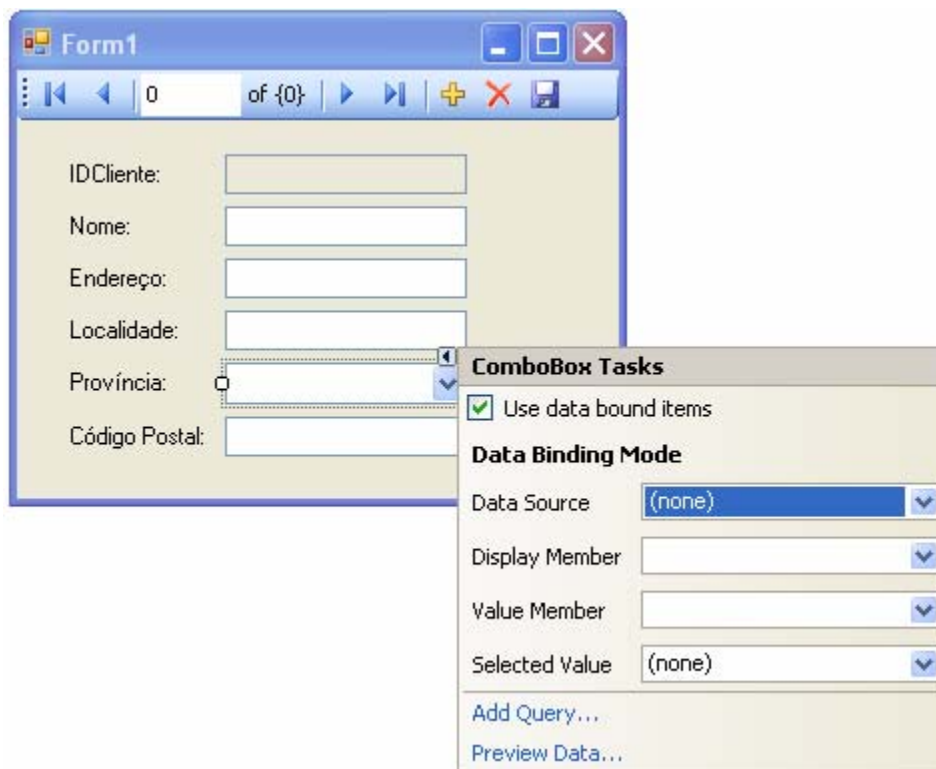
End Class
```

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

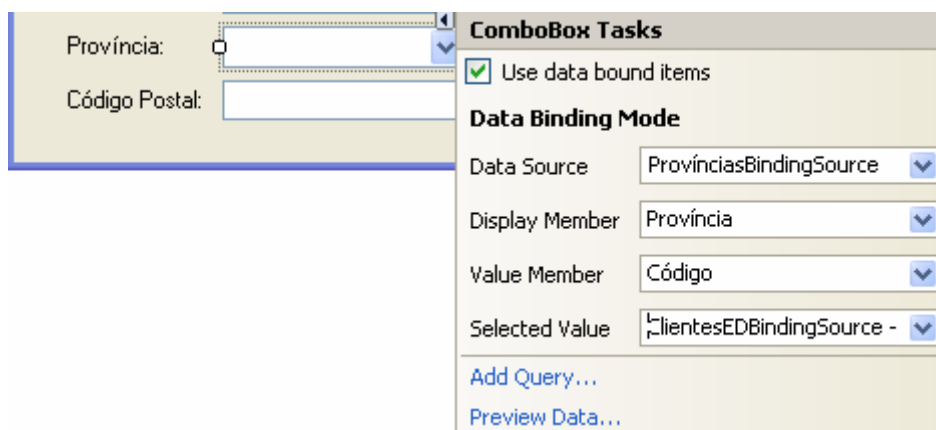
3.5 Atribua à propriedade *ReadOnly*, do objecto *IDClienteTextBox*, o valor *True*, de modo a que o utilizador não possa introduzir nem alterar valores nesta caixa de texto. Note que o atributo *IDCliente* é uma coluna *IDENTITY*.

Agora vamos ligar o controlo *ComboBox* à tabela *Províncias*, de modo a que seja possível, ao utilizador, atribuir a província para um cliente, a partir das províncias existentes na tabela *Províncias*.

3.6 Aceda ao menu *smart tag* do controlo *ProvínciaComboBox*. Active a caixa de verificação *Use data bound items*. Como resultado, vão ser mostradas as propriedades de ligação.

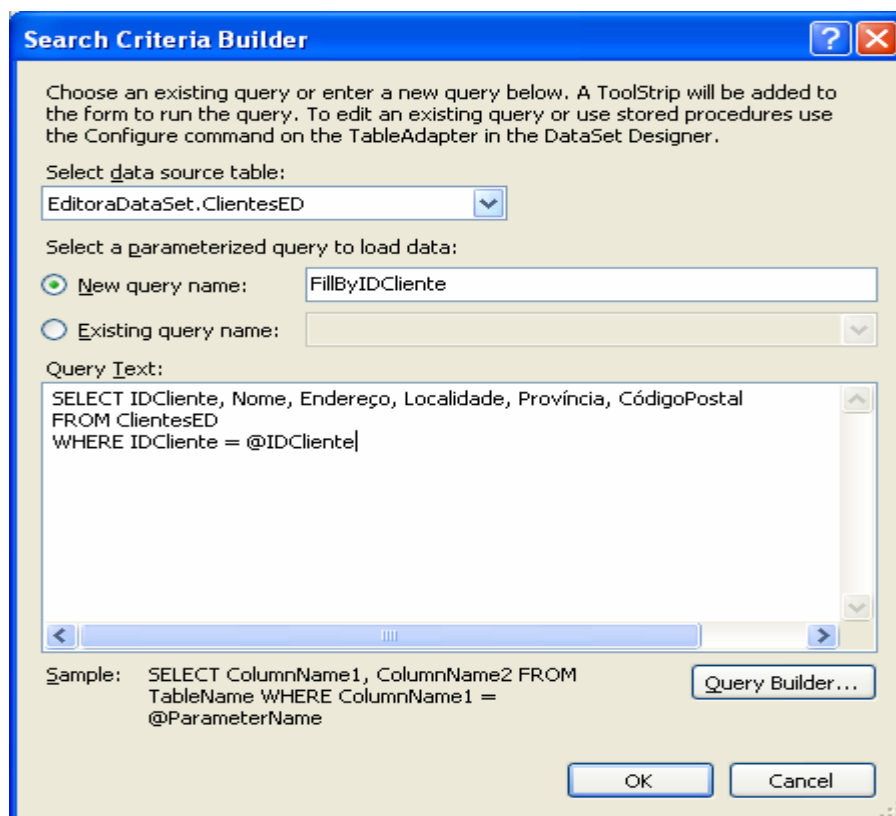


3.7 Atribua os valores para as propriedades *Data Source*, *Display Member*, *Value Member* e *Selected Value*, de acordo com a imagem seguinte.



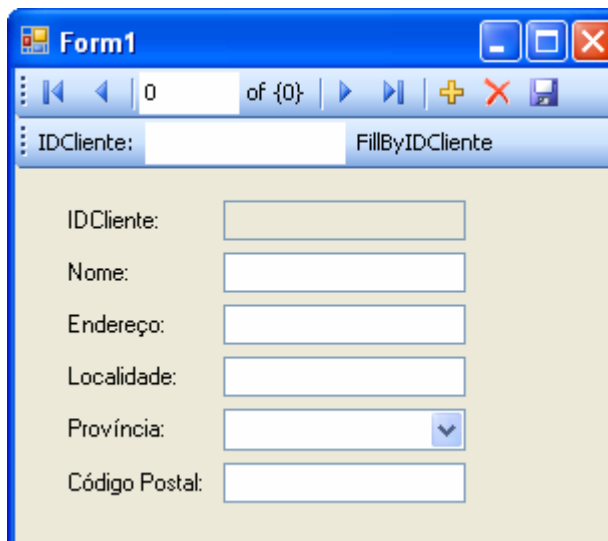
- 3.8 Seleccione a opção *DropDownList* para a propriedade *DropDown* do controlo *ProvínciaComboBox*, de modo a impedir que o utilizador introduza dados nesta caixa de combinação.
- 3.9 Altere, na secção *DataBindings* do controlo *ProvínciaComboBox*, o valor da propriedade *Text* para *none*, de modo a que o valor armazenado nesta propriedade seja ligado ao *data source*.
- 3.10 Seleccione um qualquer controlo do formulário que esteja ligado ao *data source*. Por exemplo, a caixa de combinação *ProvínciaComboBox*. Aceda ao seu menu *smart tag* e escolha a opção *Add query*.

Vai surgir a caixa de diálogo *Search Criteria Builder*.



- 3.11 Atribua o nome *FillByIDCliente* à nova consulta que está a criar.
- 3.12 Na secção *Query Text*, introduza o código SQL necessário de modo a que fique de acordo com a figura acima ilustrada.
- 3.13 Pressione o botão *OK*.

Como resultado, o Visual Studio vai acrescentar, ao formulário, uma barra de ferramentas com uma caixa de texto e um botão para que o utilizador possa introduzir um valor para o parâmetro e depois possa executar a consulta.



Note que, adicionalmente, o Visual Studio também gerou o seguinte código.

```
Private Sub FillByIDClienteToolStripButton_Click(ByVal sender As _
    System.Object, ByVal e As System.EventArgs) Handles _
    FillByIDClienteToolStripButton.Click

    Try
        Me.ClientesEDTableAdapter.FillByIDCliente(Me.EditoraDataSet.ClientesED, _
            CType(IDClienteToolStripTextBox.Text, Integer))
    Catch ex As System.Exception
        System.Windows.Forms.MessageBox.Show(ex.Message)
    End Try

End Sub
```

O código anterior pode ser reescrito do seguinte modo.

```
Private Sub FillByIDClienteToolStripButton_Click(ByVal sender As _
    System.Object, ByVal e As System.EventArgs) Handles _
    FillByIDClienteToolStripButton.Click

    Try
        Dim IDCliente As Integer = CInt(IDClienteToolStripTextBox.Text)
        Me.ClientesEDTableAdapter.FillByIDCliente(Me.EditoraDataSet.ClientesED, _
            IDCliente)
    Catch ex As System.Exception
        MessageBox.Show(ex.Message, ex.GetType.ToString)
    End Try

End Sub
```