II Semana de Tecnologia da Informação

E-Commerce com ASP.Net

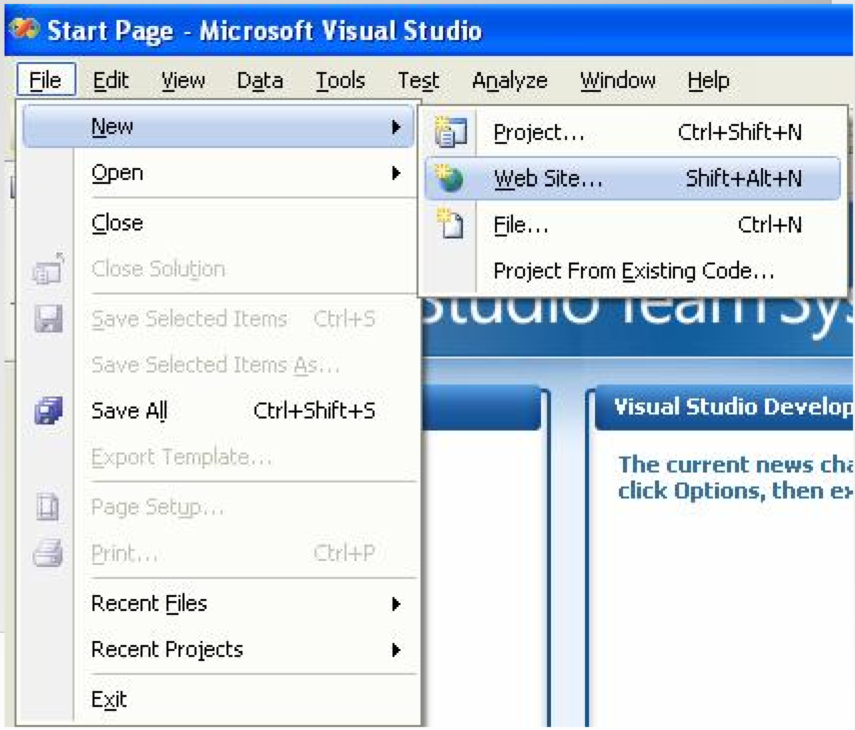
Lucas Simões Maistro
lucassimoes@univem.edu.br

E-Commerce – Tecnologias Utilizadas no Projeto

- Visual Studio 2008
  - ASP.Net 3.5
  - C# 3.0
  - ADO.Net
- SQL Server 2005 Express

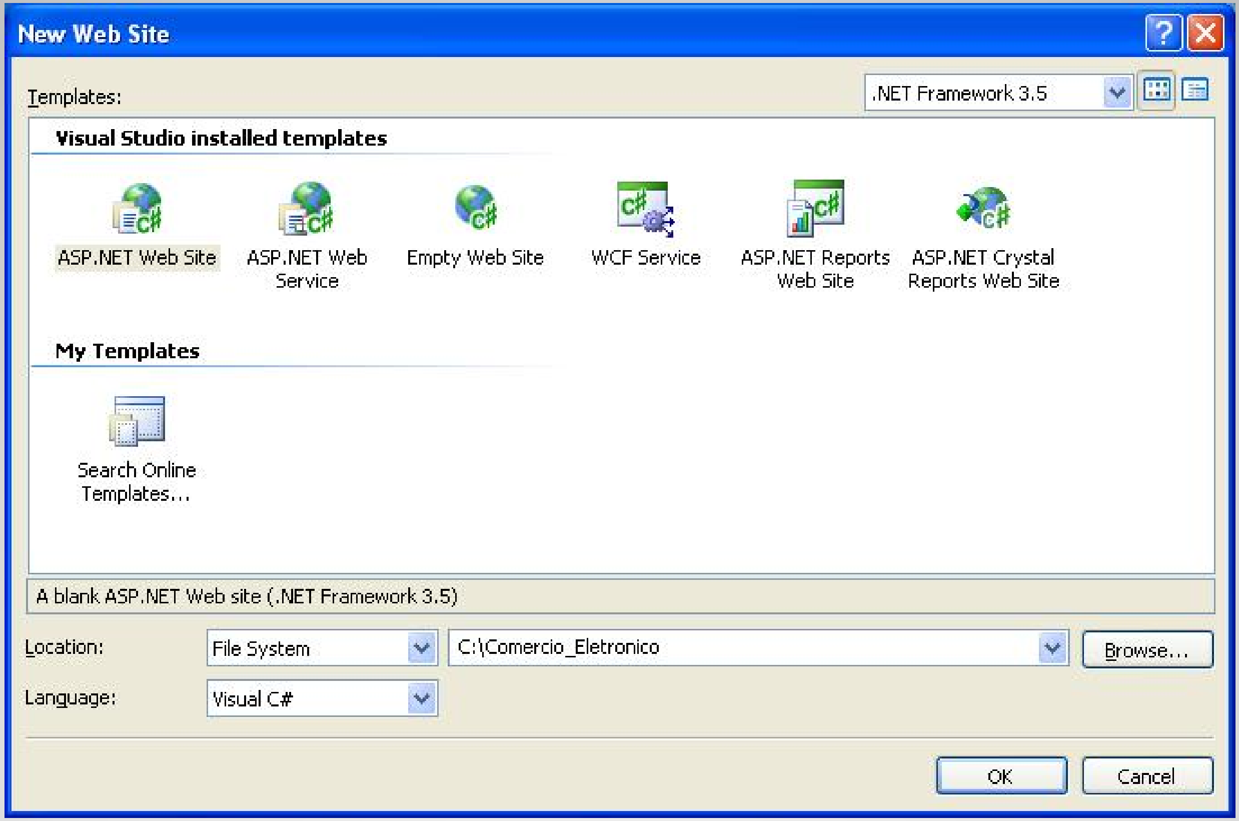
E-Commerce – Iniciando o Projeto

- Configurando o DataList
- Iniciar o serviço do SQL Server 2005 Express
- Criar o Banco de Dados Comercio_EletronicoBD
- Criar Tabelas, View e Stored Procedures
- No Visual Studio, iniciar um novo Web Site



E-Commerce – Iniciando o Projeto

- Escolher a opção ASP.Net Web Site



Templates: .NET Framework 3.5

A blank ASP.NET Web site (.NET Framework 3.5)

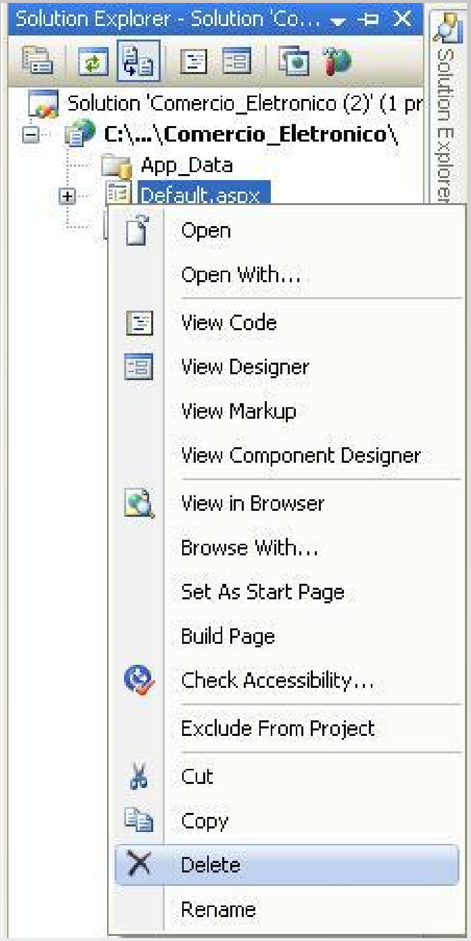
Location: File System C:\Comercio_Eletronico

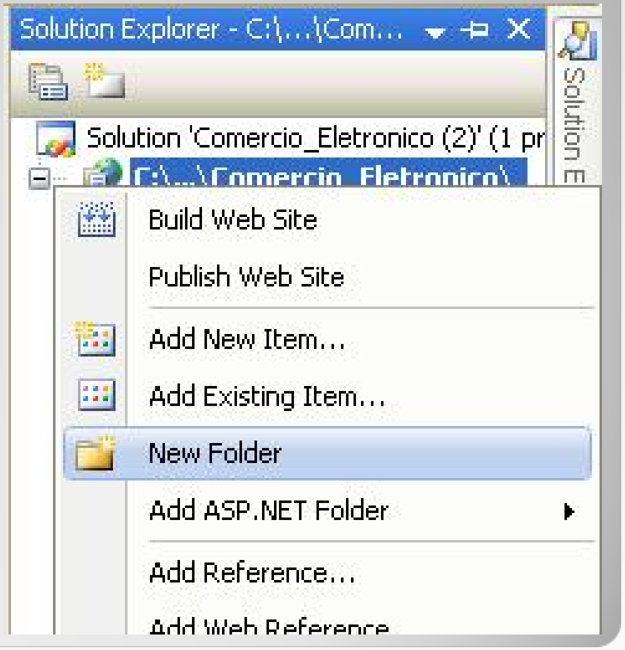
Language: Visual C#

E-Commerce – Iniciando o Projeto

- Excluir a página Default.aspx criada pelo Visual Studio

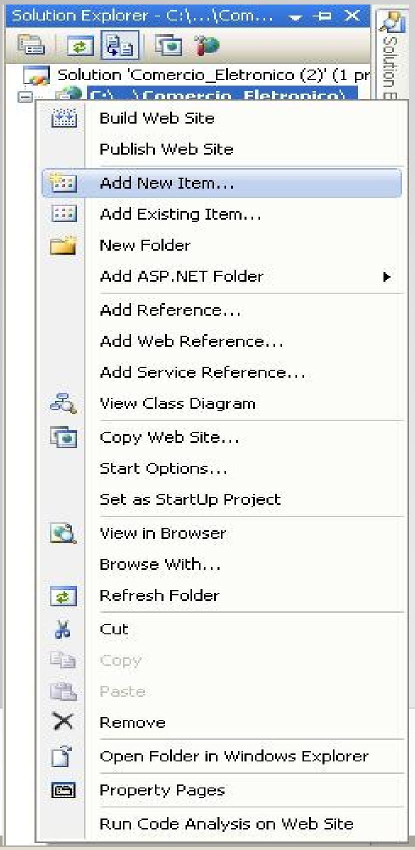
- Criar as pastas:
  - Imagens
  - ImgProduto

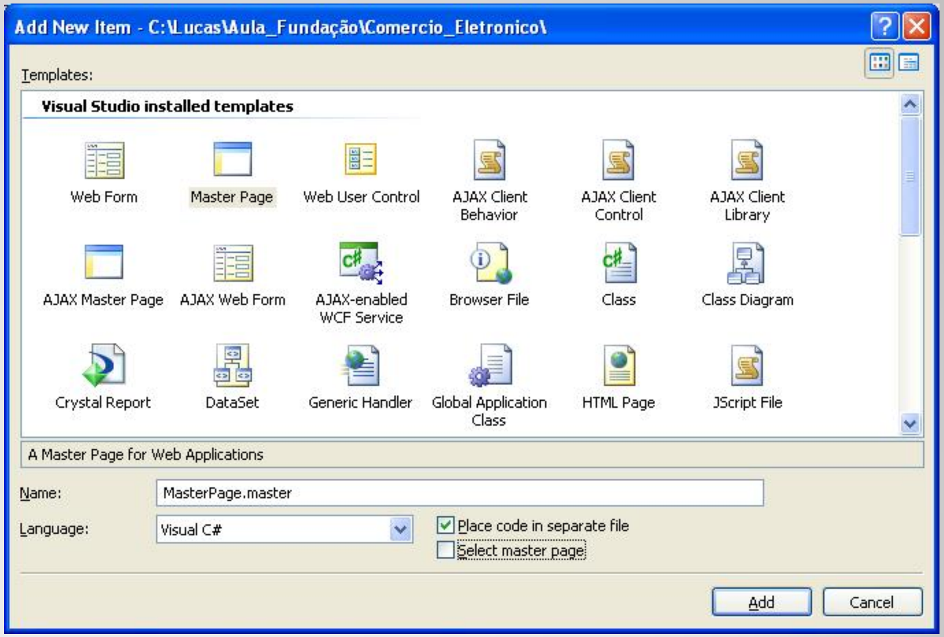




E-Commerce – Master Page

- Permite ao desenvolvedor criar um layout consistente para todas as páginas do projeto.





A Master Page for Web Applications

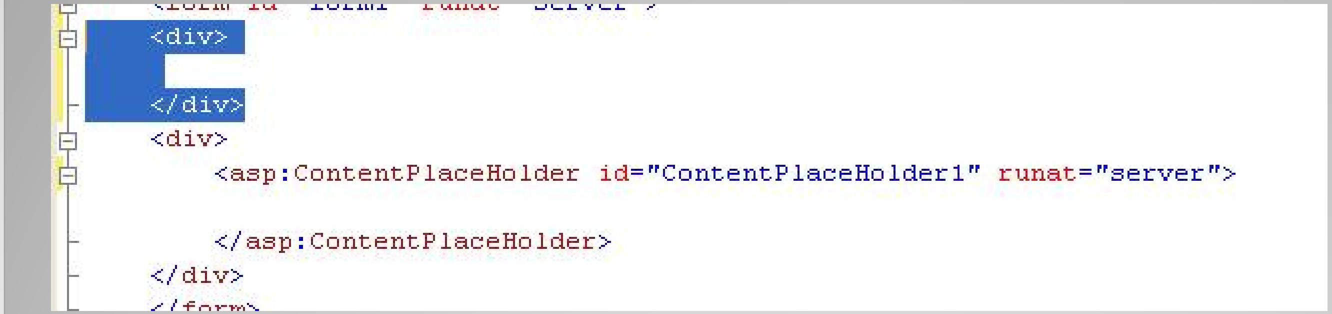
Name: MasterPage.master

Language: Visual C#

☑ Place code in separate file

□ Select master page

E-Commerce – Master Page



```
    <form id="form1" runat="server">
    <div>
    
    </div>
    <div>
        <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
        
        </asp:ContentPlaceHolder>
    </div>
    </form>
```

- Criar uma tag <div> acima da tag já criada por padrão

- Inserir um Image

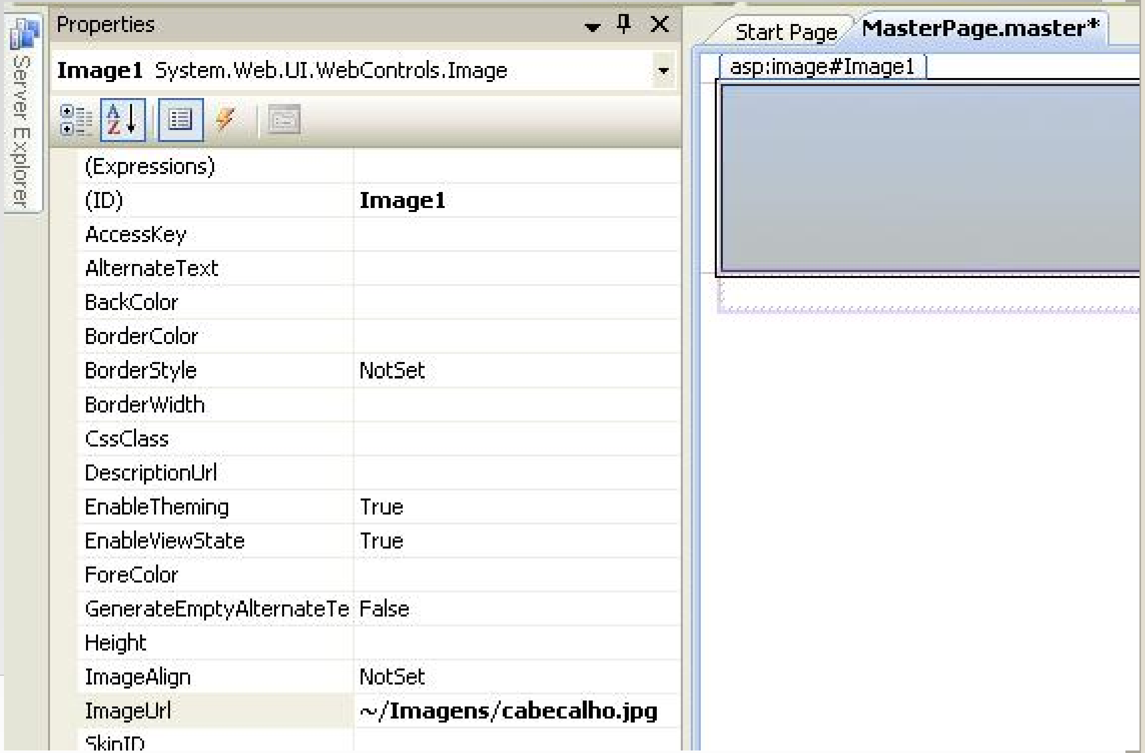
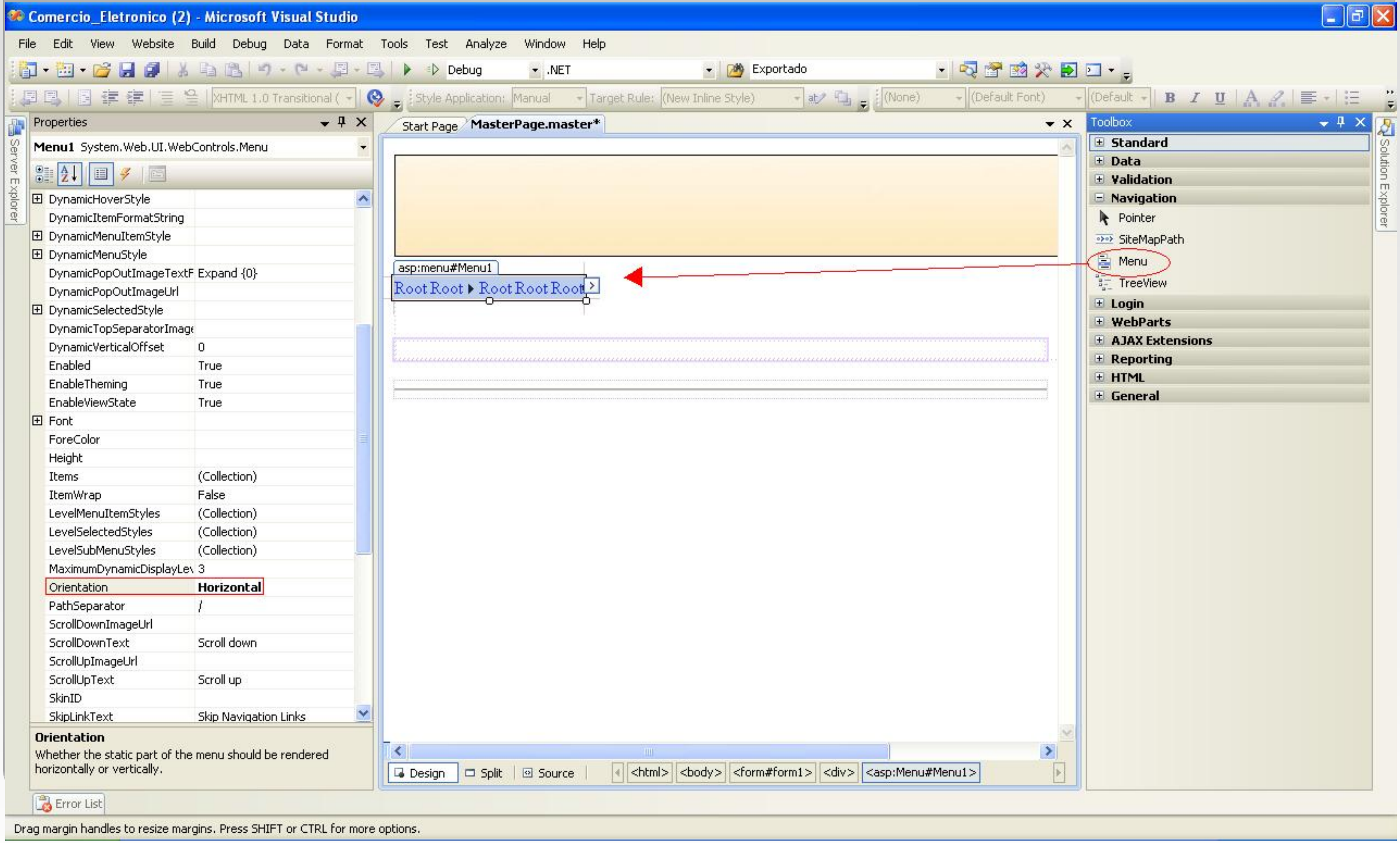


Image1 System.Web.UI.WebControls.Image

(Expressions)	
(ID)	**Image1**
AccessKey	
AlternateText	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
CssClass	
DescriptionUrl	
EnableTheming	True
EnableViewState	True
ForeColor	
GenerateEmptyAlternateTe	False
Height	
ImageAlign	NotSet
ImageUrl	**~/Imagens/cabecalho.jpg**
SkinID	

E-Commerce – Master Page (Menu)



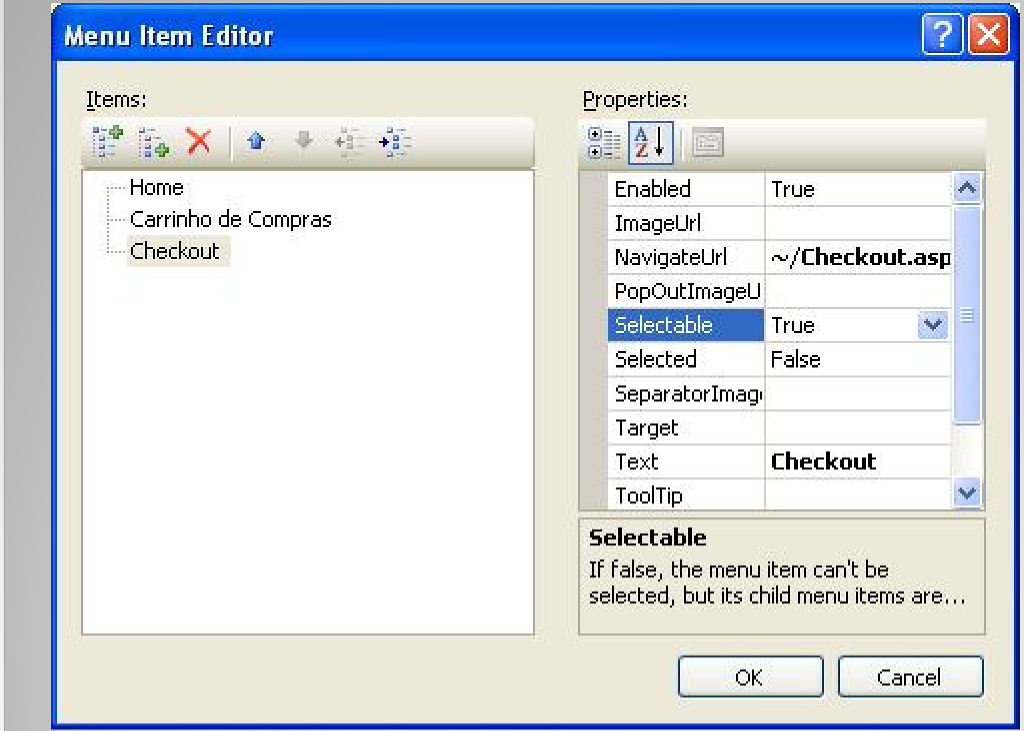
Menu1 System.Web.UI.WebControls.Menu

⊞ DynamicHoverStyle	
DynamicItemFormatString	
⊞ DynamicMenuItemStyle	
⊞ DynamicMenuStyle	
DynamicPopOutImageTextF	Expand {0}
DynamicPopOutImageUrl	
⊞ DynamicSelectedStyle	
DynamicTopSeparatorImage	
DynamicVerticalOffset	0
Enabled	True
EnableTheming	True
EnableViewState	True
⊞ Font	
ForeColor	
Height	
Items	(Collection)
ItemWrap	False
LevelMenuItemStyles	(Collection)
LevelSelectedStyles	(Collection)
LevelSubMenuStyles	(Collection)
MaximumDynamicDisplayLev	3
Orientation	**Horizontal**
PathSeparator	/
ScrollDownImageUrl	
ScrollDownText	Scroll down
ScrollUpImageUrl	
ScrollUpText	Scroll up
SkinID	
SkipLinkText	Skip Navigation Links

Orientation
Whether the static part of the menu should be rendered horizontally or vertically.

Drag margin handles to resize margins. Press SHIFT or CTRL for more options.

E-Commerce – Master Page (Menu)

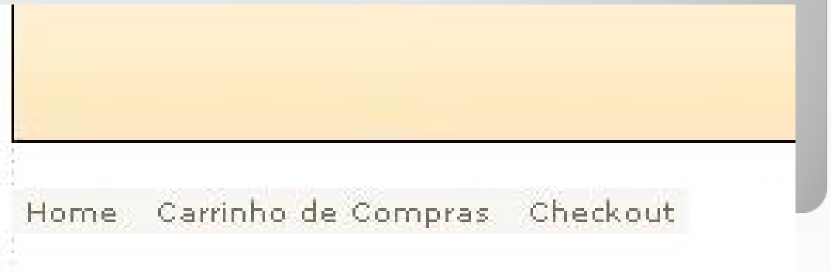


- Home
- Carrinho de Compras
- Checkout

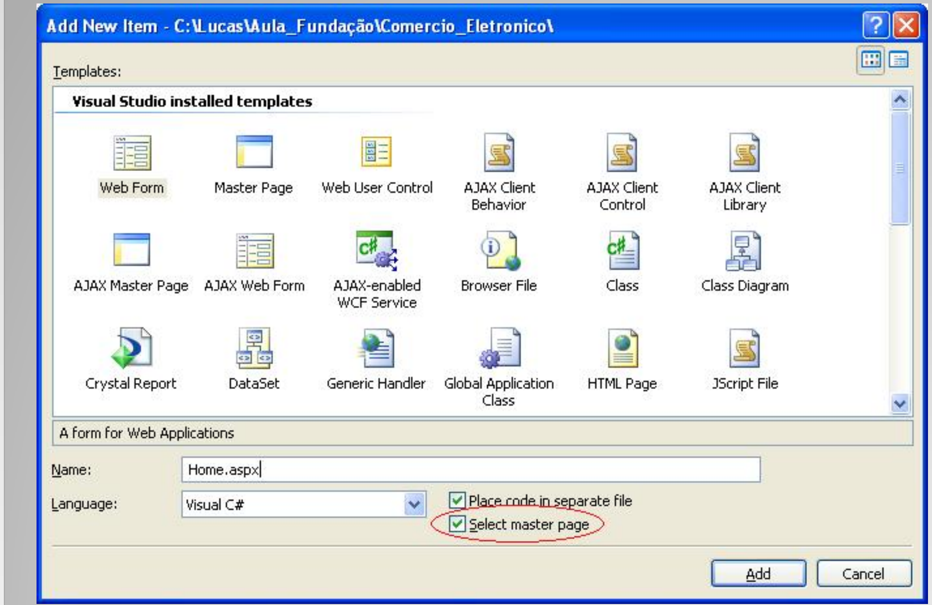
Enabled	True
ImageUrl	
NavigateUrl	**~/Checkout.asp**
PopOutImageU	
Selectable	True
Selected	False
SeparatorImag	
Target	
Text	**Checkout**
ToolTip	

Selectable
If false, the menu item can't be selected, but its child menu items are…

- Criar os itens ao lado



E-Commerce – Criando a página Home



Add New Item - C:\Lucas\Aula_Fundação\Comercio_Eletronico\

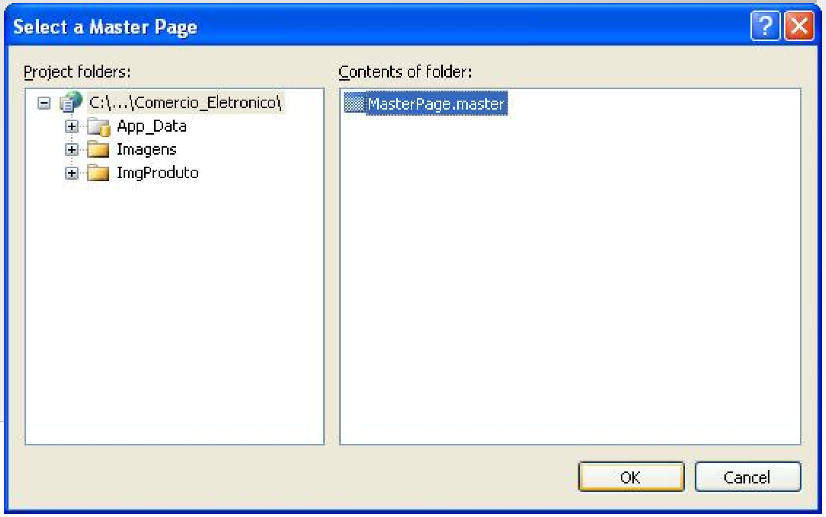
A form for Web Applications

Name: Home.aspx

Language: Visual C#

☑ Place code in separate file

☑ Select master page



Select a Master Page

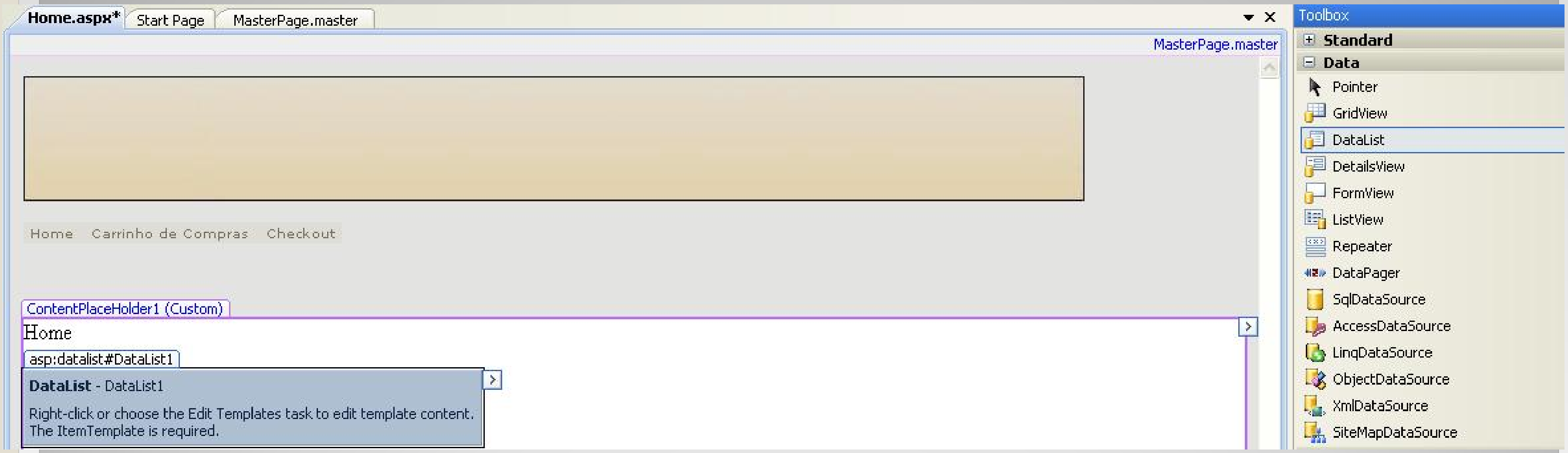
Project folders:

- C:\...\Comercio_Eletronico\
  - App_Data
  - Imagens
  - ImgProduto

Contents of folder:

MasterPage.master

E-Commerce – DataList



DataList - DataList1

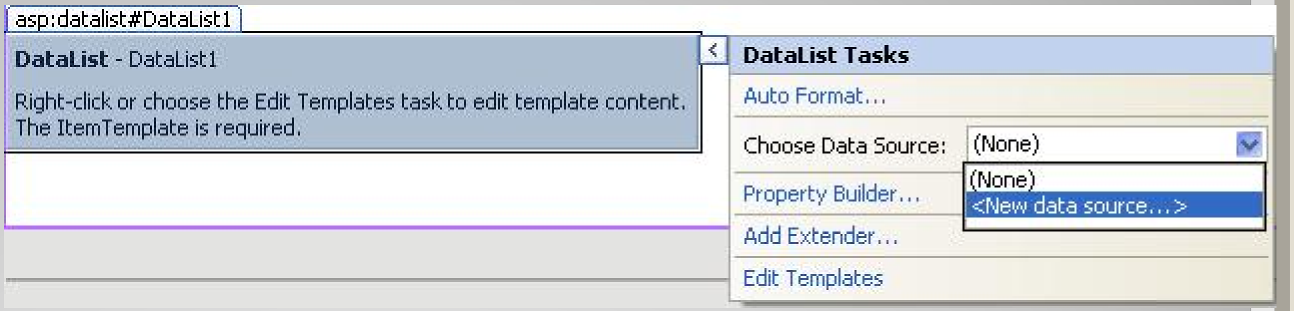
Right-click or choose the Edit Templates task to edit template content.
The ItemTemplate is required.

- DataList renderiza uma lista de controles

E-Commerce – SQLDataSource

- Permite que de forma rápida se selecione uma base de dados

- Selecione o tipo de fonte de dados


asp:datalist#DataList1

DataList - DataList1

Right-click or choose the Edit Templates task to edit template content. The ItemTemplate is required.

DataList Tasks

Auto Format...

Choose Data Source: (None)

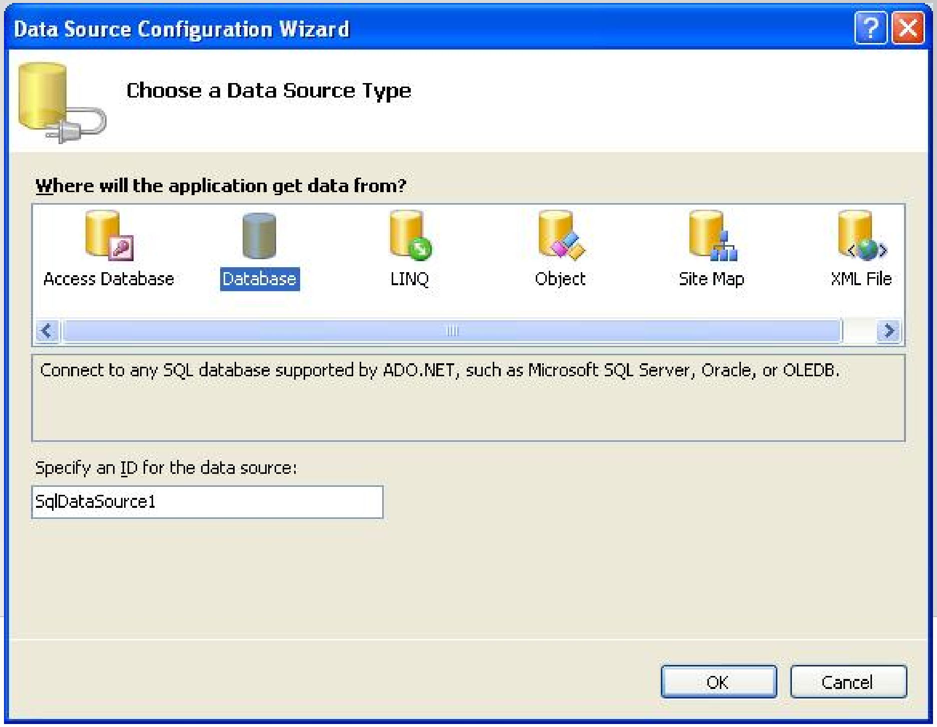
(None)

<New data source...>

Property Builder...

Add Extender...

Edit Templates


Data Source Configuration Wizard

Choose a Data Source Type

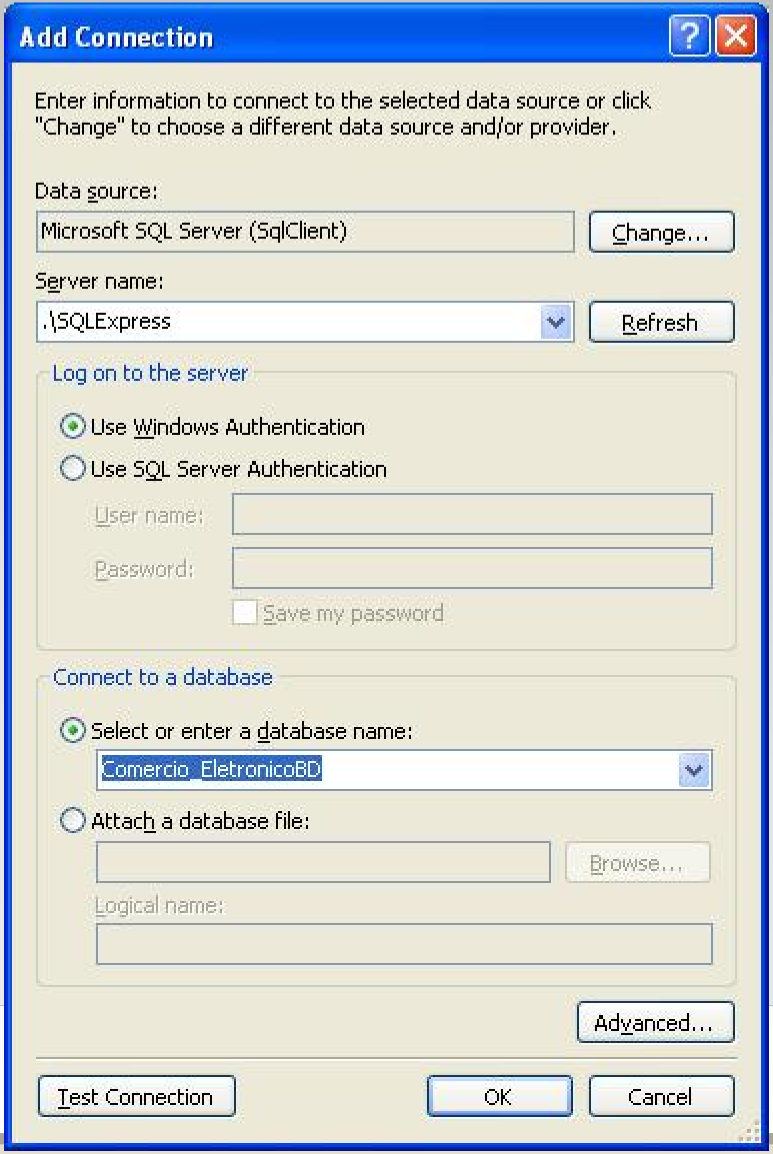
Where will the application get data from?

Connect to any SQL database supported by ADO.NET, such as Microsoft SQL Server, Oracle, or OLEDB.

Specify an ID for the data source: SqlDataSource1

E-Commerce – SQLDataSource

- Criando a conexão com o banco de dados



Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

Server name: .\SQLExpress

Log on to the server

● Use Windows Authentication

○ Use SQL Server Authentication

User name:

Password:

□ Save my password

Connect to a database

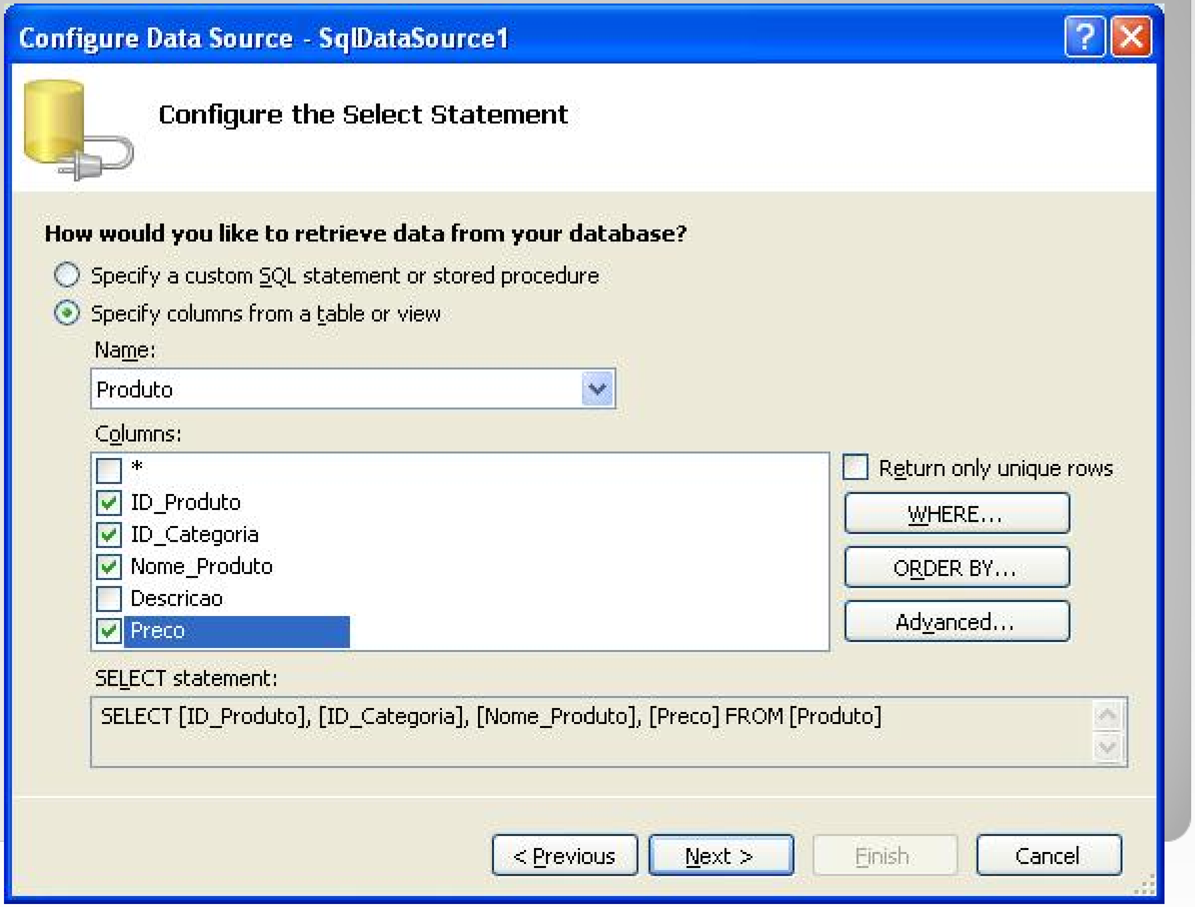
● Select or enter a database name: Comercio_EletronicoBD

○ Attach a database file:

Logical name:

E-Commerce – SQLDataSource

- Montando a SQL para carregar produtos



Configure the Select Statement

How would you like to retrieve data from your database?

○ Specify a custom SQL statement or stored procedure

● Specify columns from a table or view

Name: Produto

Columns:

- □ *
- ☑ ID_Produto
- ☑ ID_Categoria
- ☑ Nome_Produto
- □ Descricao
- ☑ Preco

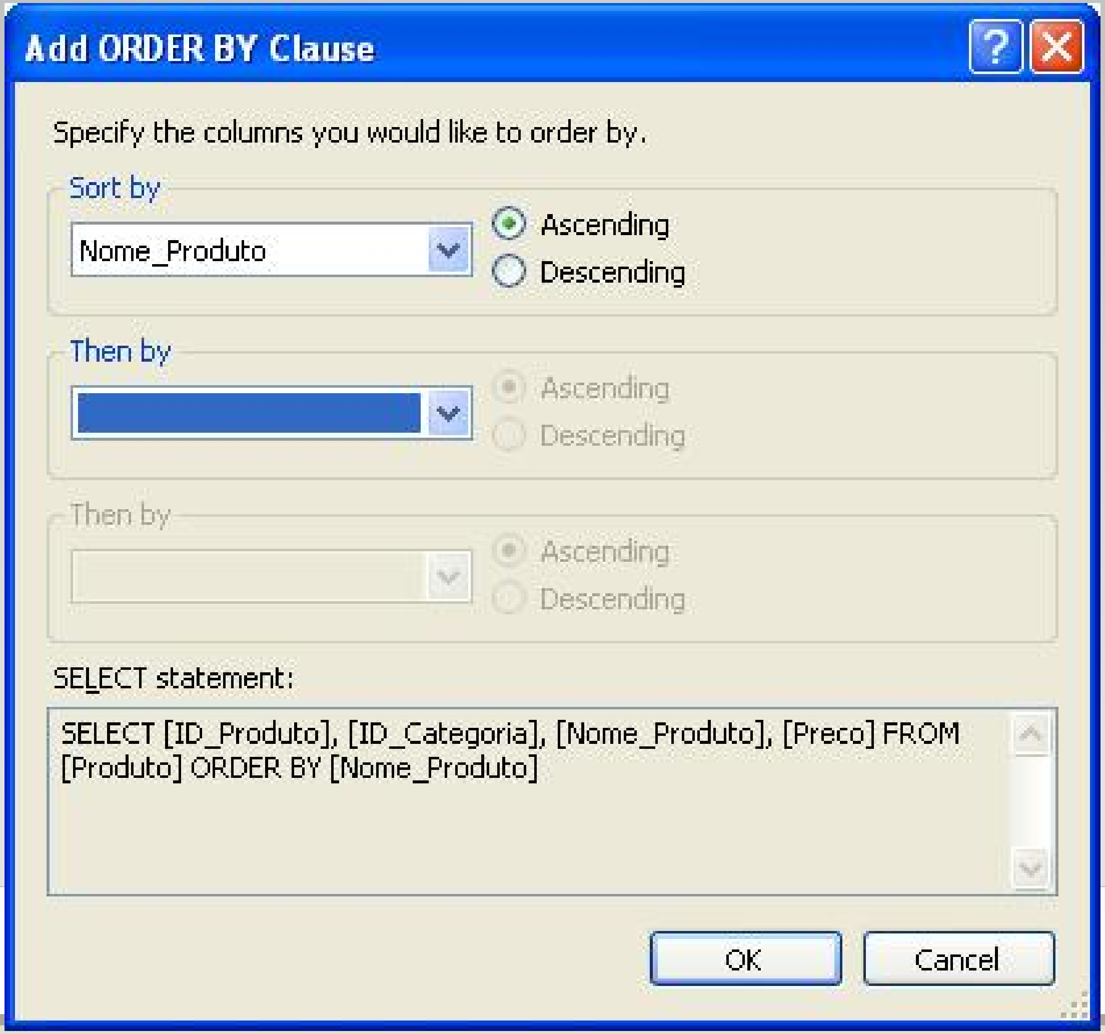
□ Return only unique rows

SELECT statement:

```
SELECT [ID_Produto], [ID_Categoria], [Nome_Produto], [Preco] FROM [Produto]
```

E-Commerce – SQLDataSource

- Ordenando o resultado pelo nome do produto



Sort by: Nome_Produto — ● Ascending ○ Descending

Then by: — ● Ascending ○ Descending

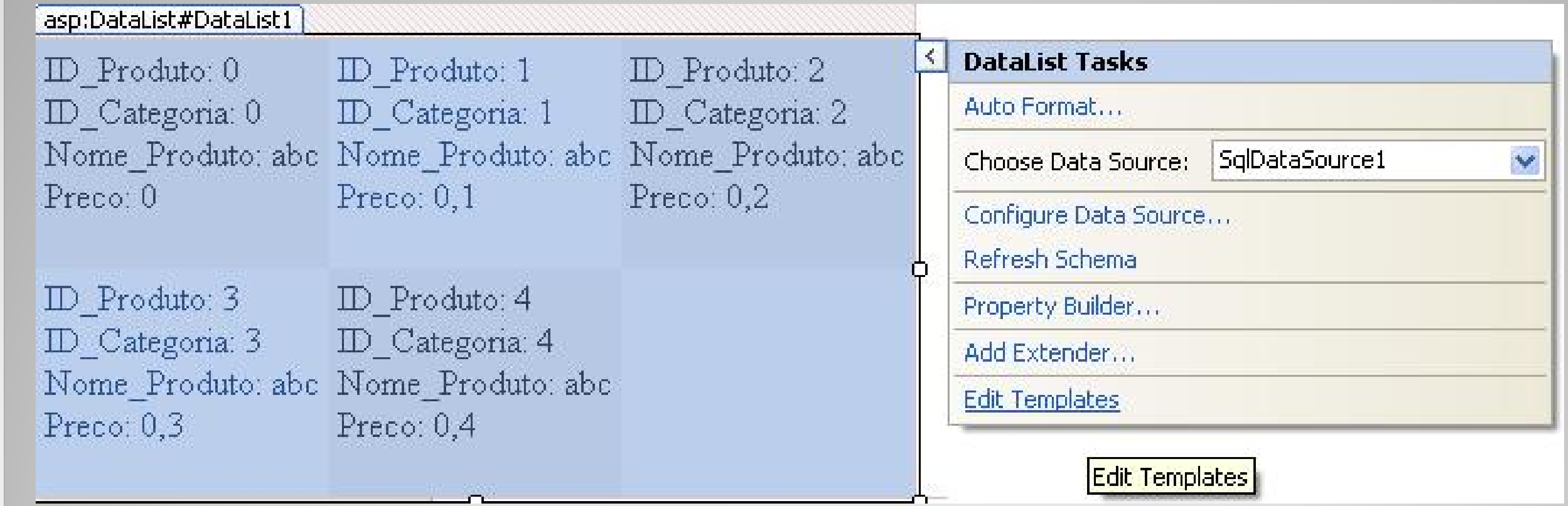
Then by: — ● Ascending ○ Descending

SELECT statement:

```
SELECT [ID_Produto], [ID_Categoria], [Nome_Produto], [Preco] FROM
[Produto] ORDER BY [Nome_Produto]
```

E-Commerce – DataList

- Configurando o DataList



asp:DataList#DataList1

ID_Produto: 0 ID_Categoria: 0 Nome_Produto: abc Preco: 0	ID_Produto: 1 ID_Categoria: 1 Nome_Produto: abc Preco: 0,1	ID_Produto: 2 ID_Categoria: 2 Nome_Produto: abc Preco: 0,2
ID_Produto: 3 ID_Categoria: 3 Nome_Produto: abc Preco: 0,3	ID_Produto: 4 ID_Categoria: 4 Nome_Produto: abc Preco: 0,4	

DataList Tasks

Auto Format...

Choose Data Source: SqlDataSource1

Configure Data Source...

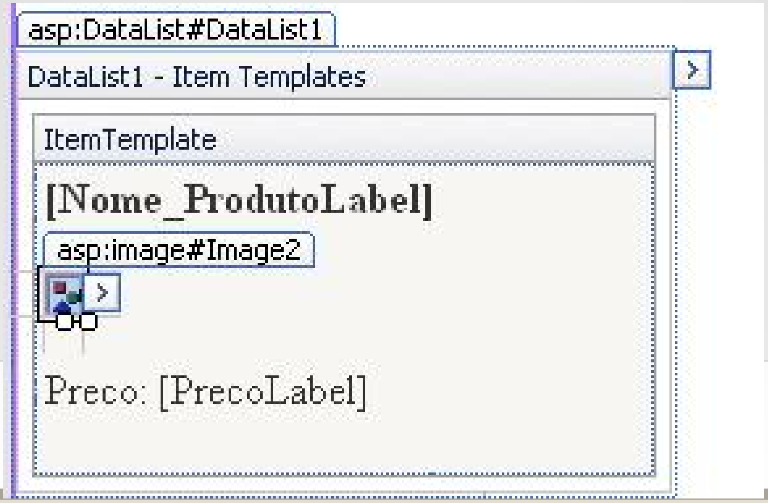
Refresh Schema

Property Builder...

Add Extender...

Edit Templates

Edit Templates



asp:DataList#DataList1

DataList1 - Item Templates

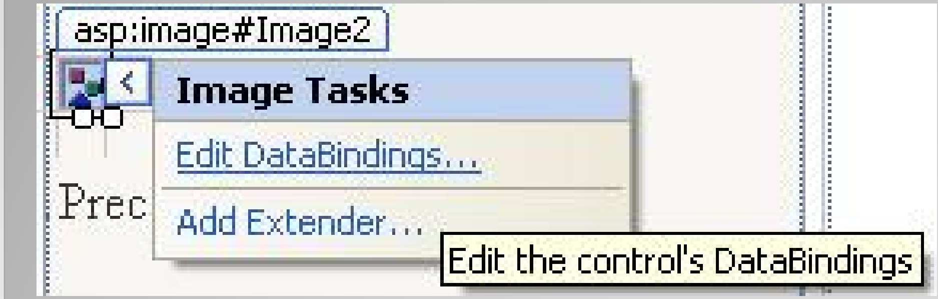
ItemTemplate

[Nome_ProdutoLabel]

asp:image#Image2

Preco: [PrecoLabel]

E-Commerce – Image



- Clicar em Edit DataBindings

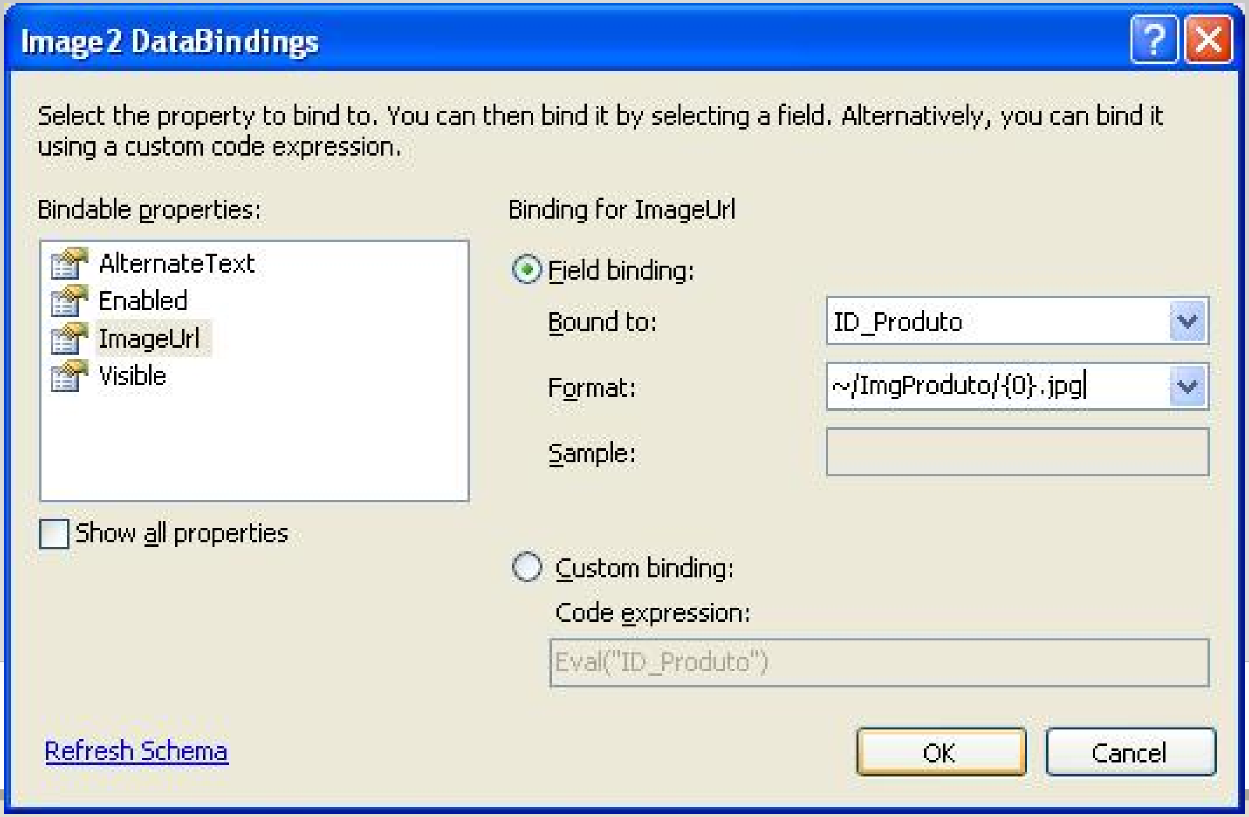


Image2 DataBindings

Select the property to bind to. You can then bind it by selecting a field. Alternatively, you can bind it using a custom code expression.

Bindable properties:

- AlternateText
- Enabled
- ImageUrl
- Visible

□ Show all properties

Binding for ImageUrl

● Field binding:

Bound to: ID_Produto

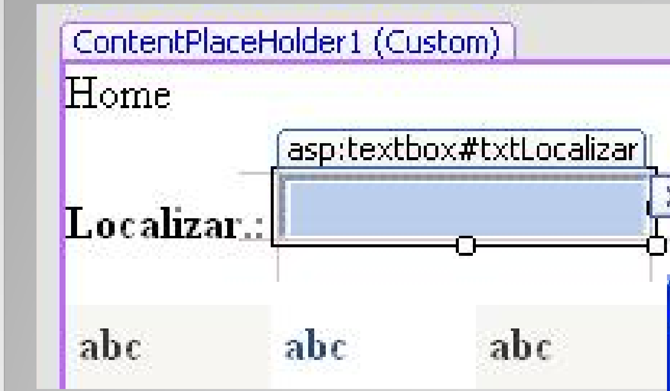
Format: ~/ImgProduto/{0}.jpg

Sample:

○ Custom binding:

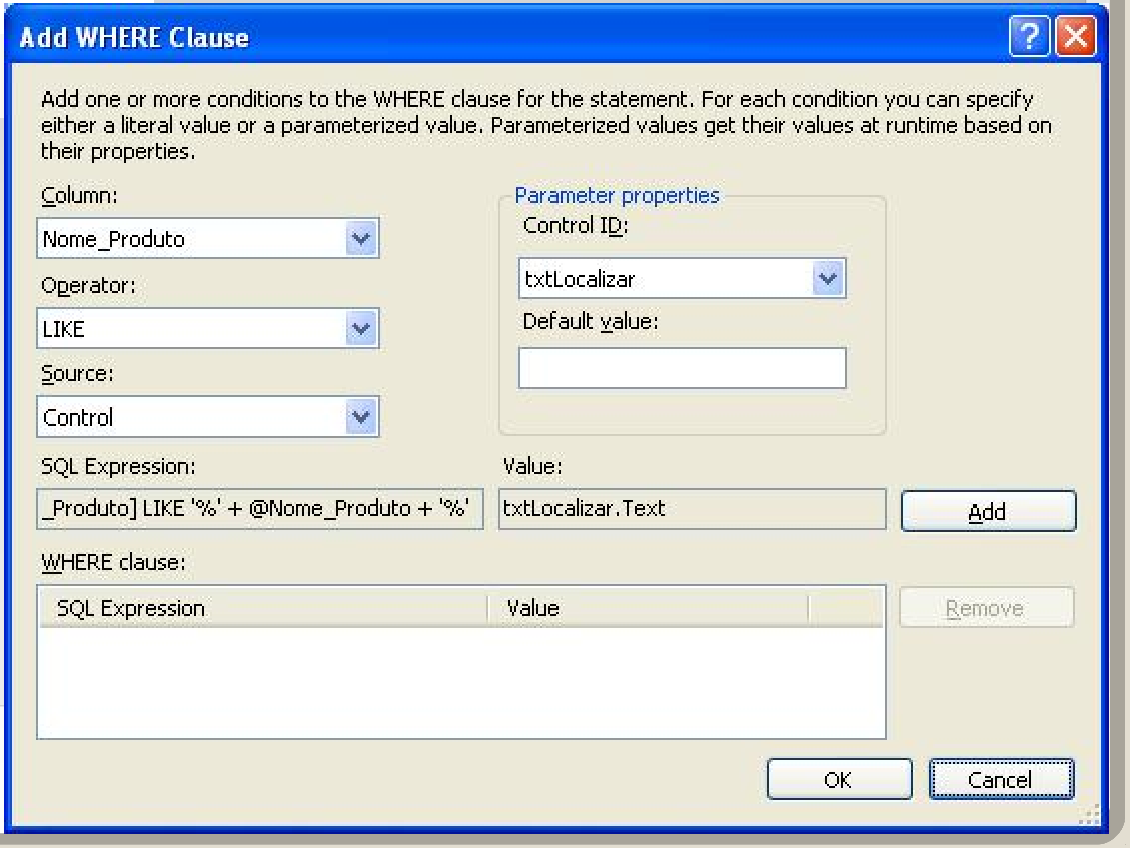
Code expression: Eval("ID_Produto")

E-Commerce – Localizar



- Inserir um TextBox

- No SQLDataSource:
  - Configure DataSource



Add WHERE Clause

Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column: Nome_Produto

Operator: LIKE

Source: Control

Parameter properties

Control ID: txtLocalizar

Default value:

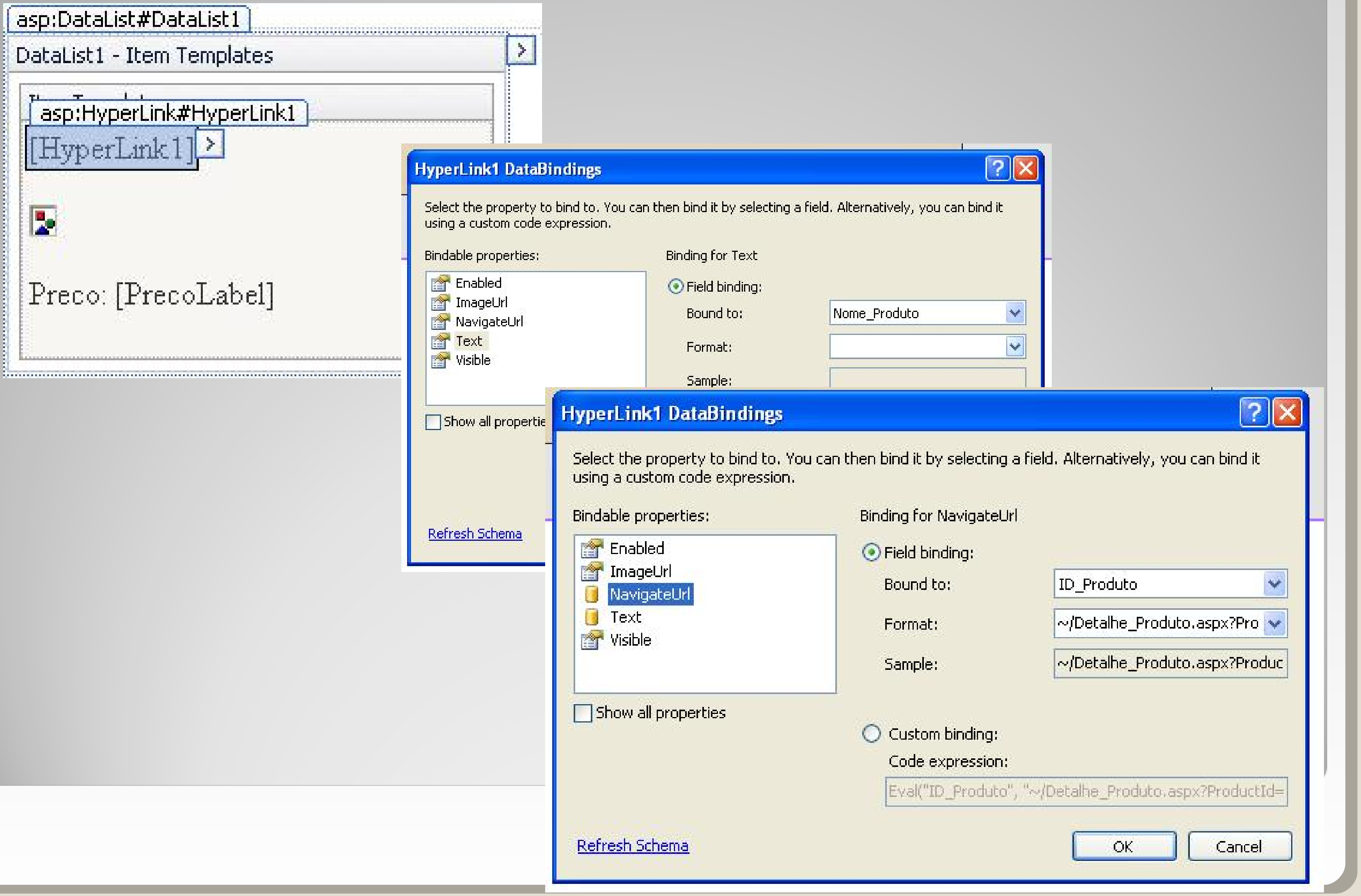
SQL Expression: _Produto] LIKE '%' + @Nome_Produto + '%'

Value: txtLocalizar.Text

WHERE clause:

SQL Expression	Value

E-Commerce – HyperLink nos Produtos



asp:DataList#DataList1

DataList1 - Item Templates

asp:HyperLink#HyperLink1

[HyperLink1]

Preco: [PrecoLabel]

HyperLink1 DataBindings

Select the property to bind to. You can then bind it by selecting a field. Alternatively, you can bind it using a custom code expression.

Bindable properties:

- Enabled
- ImageUrl
- NavigateUrl
- Text
- Visible

Binding for Text

● Field binding:

Bound to: Nome_Produto

Format:

Sample:

□ Show all propertie

Refresh Schema

HyperLink1 DataBindings

Select the property to bind to. You can then bind it by selecting a field. Alternatively, you can bind it using a custom code expression.

Bindable properties:

- Enabled
- ImageUrl
- NavigateUrl
- Text
- Visible

□ Show all properties

Binding for NavigateUrl

● Field binding:

Bound to: ID_Produto

Format: ~/Detalhe_Produto.aspx?Pro

Sample: ~/Detalhe_Produto.aspx?Produc

○ Custom binding:

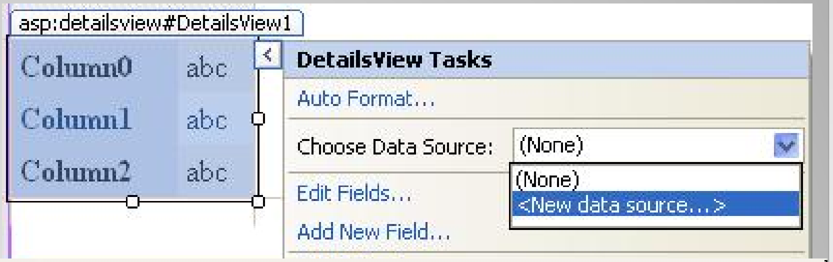
Code expression:

```
Eval("ID_Produto", "~/Detalhe_Produto.aspx?ProductId=
```

Refresh Schema

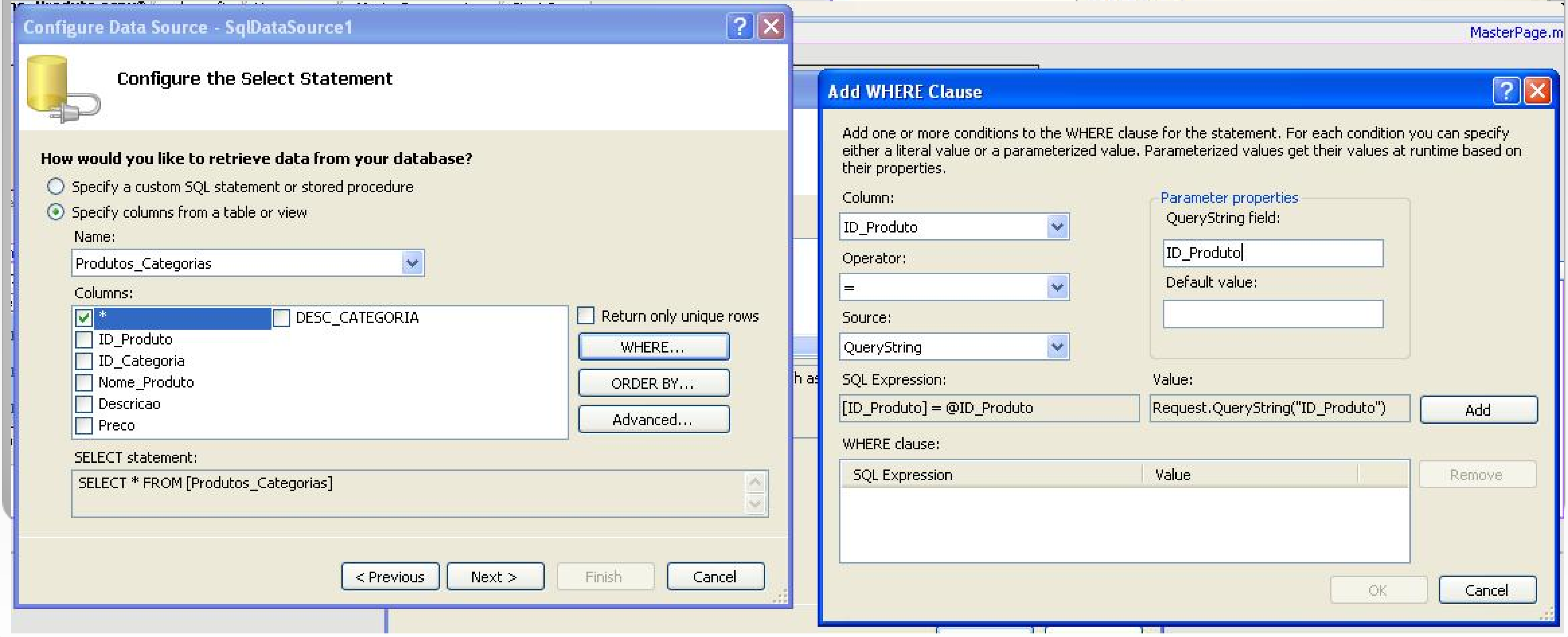
E-Commerce – Página Detalhes

- Adicionar um novo Item
  - Salvar como Detalhe_Produto.aspx
  - Selecionar a Master Page
- Inserir um DetailsView



asp:detailsview#DetailsView1

Column0	abc
Column1	abc
Column2	abc



Configure Data Source - SqlDataSource1

Configure the Select Statement

How would you like to retrieve data from your database?

○ Specify a custom SQL statement or stored procedure

● Specify columns from a table or view

Name: Produtos_Categorias

Columns:

☑ *
□ DESC_CATEGORIA
□ ID_Produto
□ ID_Categoria
□ Nome_Produto
□ Descricao
□ Preco

□ Return only unique rows

SELECT statement:

```
SELECT * FROM [Produtos_Categorias]
```

Add WHERE Clause

Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column: ID_Produto

Operator: =

Source: QueryString

Parameter properties

QueryString field: ID_Produto

Default value:

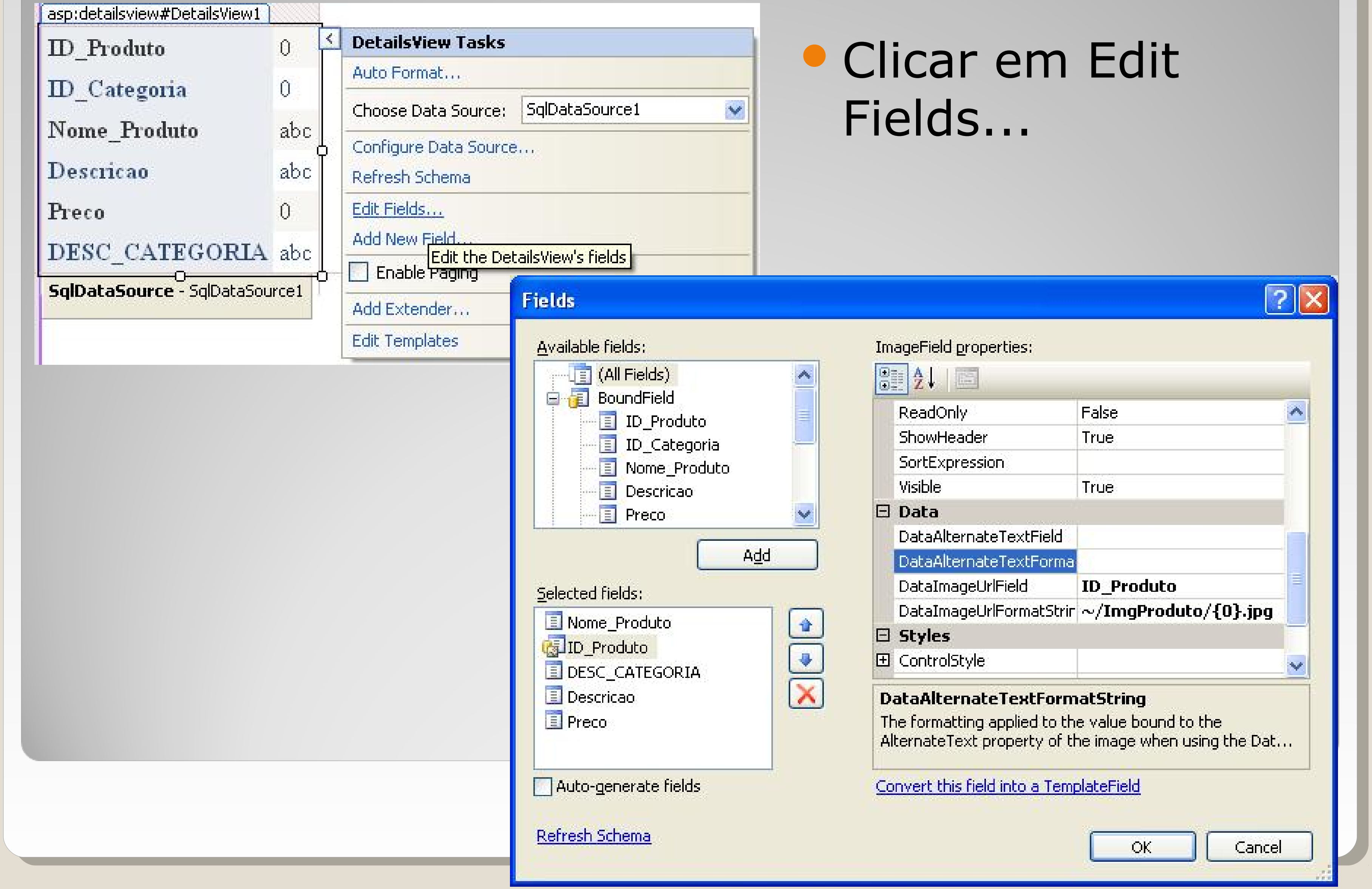
SQL Expression: [ID_Produto] = @ID_Produto

Value: Request.QueryString("ID_Produto")

WHERE clause:

SQL Expression	Value

E-Commerce – DetailsView



asp:detailsview#DetailsView1

ID_Produto	0
ID_Categoria	0
Nome_Produto	abc
Descricao	abc
Preco	0
DESC_CATEGORIA	abc

SqlDataSource - SqlDataSource1

Choose Data Source: SqlDataSource1

□ Enable Paging

Edit the DetailsView's fields

Available fields:

- (All Fields)
- BoundField
  - ID_Produto
  - ID_Categoria
  - Nome_Produto
  - Descricao
  - Preco

Selected fields:

- Nome_Produto
- ID_Produto
- DESC_CATEGORIA
- Descricao
- Preco

□ Auto-generate fields

ImageField properties:

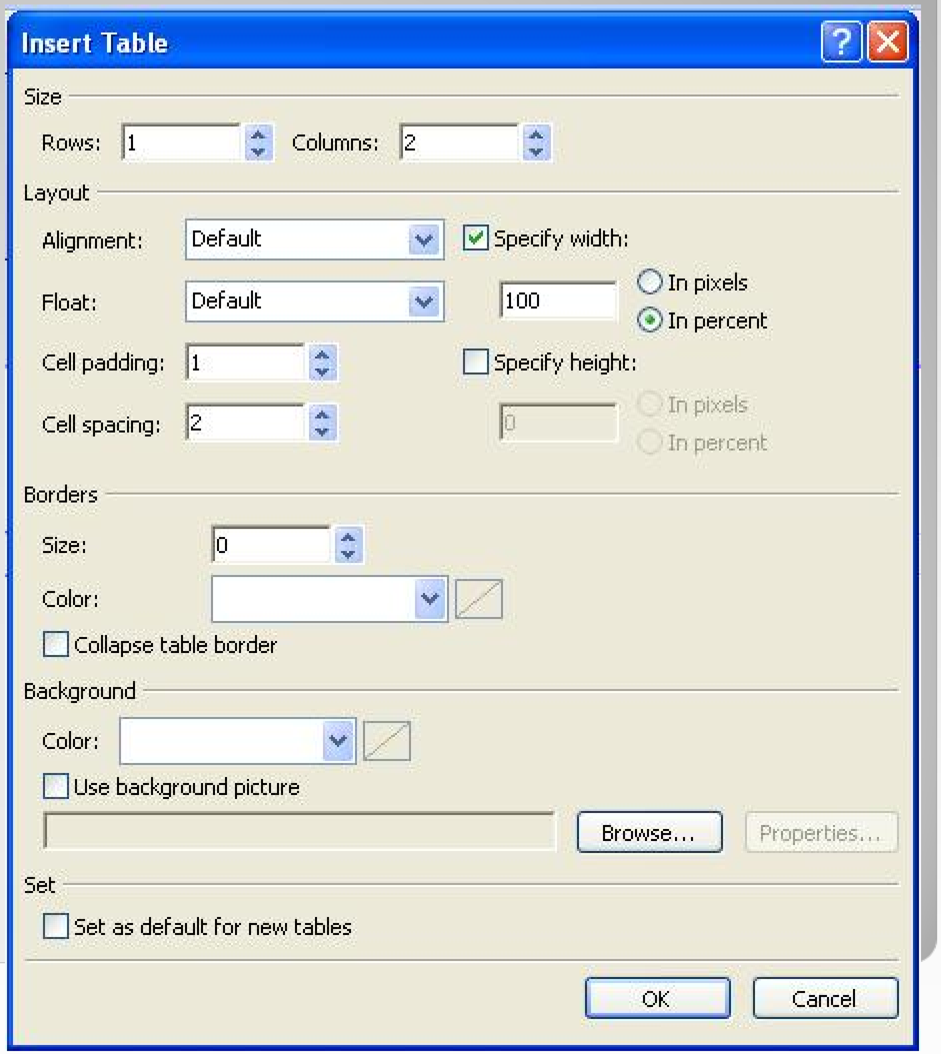
ReadOnly	False
ShowHeader	True
SortExpression	
Visible	True
⊟ **Data**	
DataAlternateTextField	
DataAlternateTextForma	
DataImageUrlField	**ID_Produto**
DataImageUrlFormatStrir	**~/ImgProduto/{0}.jpg**
⊟ **Styles**	
⊞ ControlStyle	

DataAlternateTextFormatString
The formatting applied to the value bound to the AlternateText property of the image when using the Dat...

- Clicar em Edit Fields...

E-Commerce – Pesquisar por Categorias

- Na página Home.aspx
  - Inserir tabela com 1 Linha e 2 Colunas



Insert Table

Size

Rows: 1 Columns: 2

Layout

Alignment: Default

☑ Specify width:

Float: Default

100

○ In pixels

● In percent

Cell padding: 1

□ Specify height:

Cell spacing: 2

0

○ In pixels

○ In percent

Borders

Size: 0

Color:

□ Collapse table border

Background

Color:

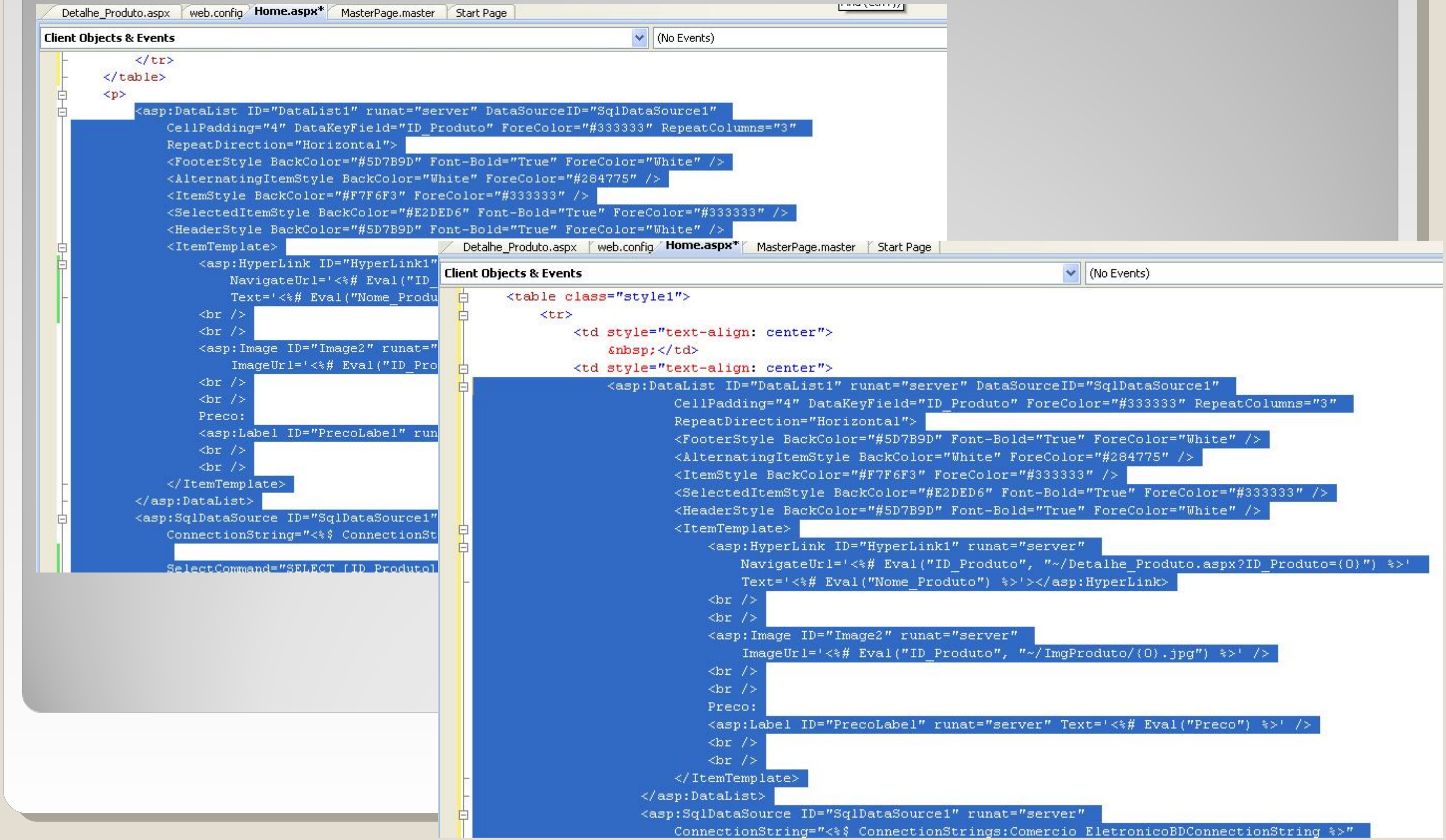
□ Use background picture

Set

□ Set as default for new tables

E-Commerce – Pesquisar por Categorias

- Mover o DataList para dentro da 2ª célula da tabela



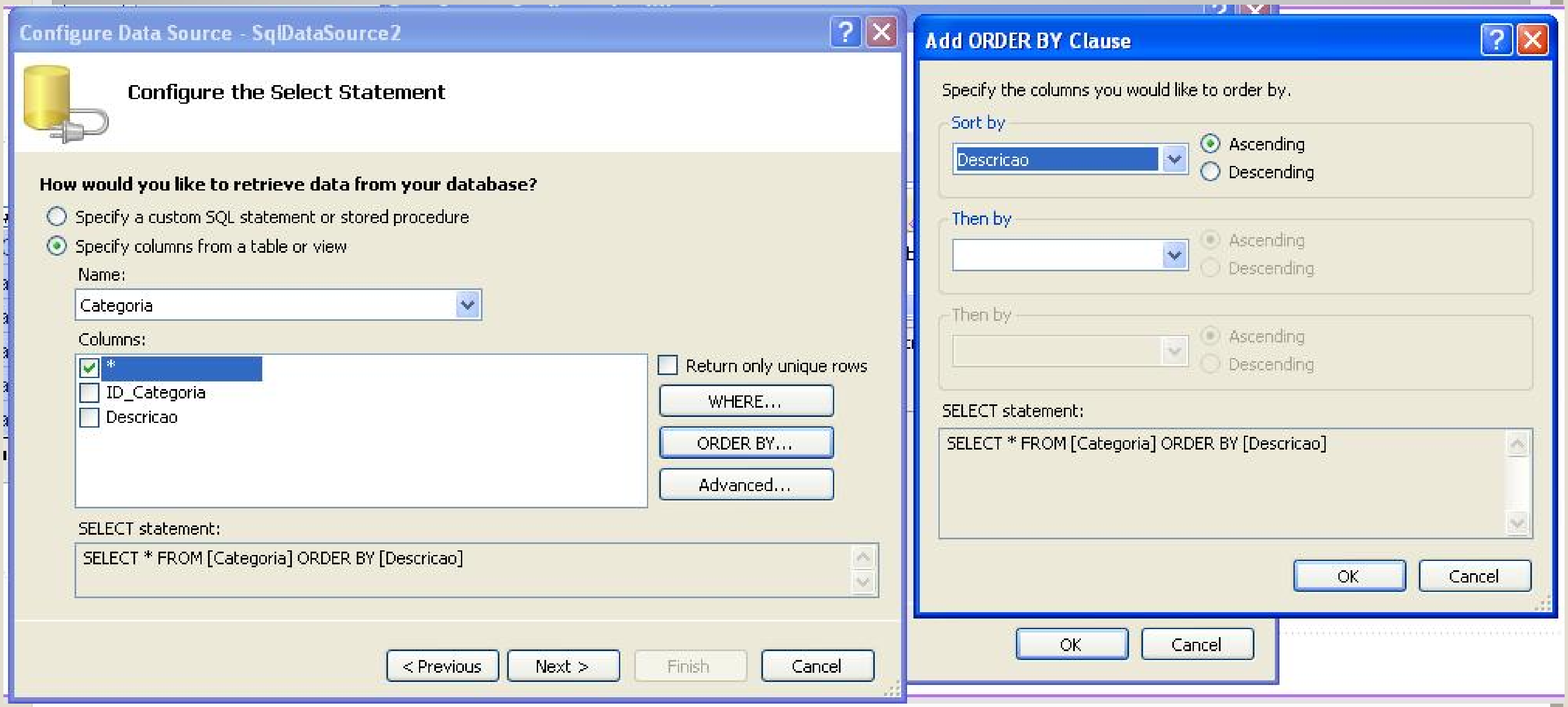
```
            </tr>
        </table>
        <p>
            <asp:DataList ID="DataList1" runat="server" DataSourceID="SqlDataSource1"
                CellPadding="4" DataKeyField="ID_Produto" ForeColor="#333333" RepeatColumns="3"
                RepeatDirection="Horizontal">
                <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
                <AlternatingItemStyle BackColor="White" ForeColor="#284775" />
                <ItemStyle BackColor="#F7F6F3" ForeColor="#333333" />
                <SelectedItemStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333" />
                <HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
                <ItemTemplate>
                    <asp:HyperLink ID="HyperLink1"
                        NavigateUrl='<%# Eval("ID_
                        Text='<%# Eval("Nome_Produ
                    <br />
                    <br />
                    <asp:Image ID="Image2" runat="
                        ImageUrl='<%# Eval("ID_Pro
                    <br />
                    <br />
                    Preco:
                    <asp:Label ID="PrecoLabel" run
                    <br />
                    <br />
                </ItemTemplate>
            </asp:DataList>
            <asp:SqlDataSource ID="SqlDataSource1"
                ConnectionString="<%$ ConnectionSt

                SelectCommand="SELECT [ID_Produto]
```

```
        <table class="style1">
            <tr>
                <td style="text-align: center">
                    &nbsp;</td>
                <td style="text-align: center">
                    <asp:DataList ID="DataList1" runat="server" DataSourceID="SqlDataSource1"
                        CellPadding="4" DataKeyField="ID_Produto" ForeColor="#333333" RepeatColumns="3"
                        RepeatDirection="Horizontal">
                        <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
                        <AlternatingItemStyle BackColor="White" ForeColor="#284775" />
                        <ItemStyle BackColor="#F7F6F3" ForeColor="#333333" />
                        <SelectedItemStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333" />
                        <HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
                        <ItemTemplate>
                            <asp:HyperLink ID="HyperLink1" runat="server"
                                NavigateUrl='<%# Eval("ID_Produto", "~/Detalhe_Produto.aspx?ID_Produto={0}") %>'
                                Text='<%# Eval("Nome_Produto") %>'></asp:HyperLink>
                            <br />
                            <br />
                            <asp:Image ID="Image2" runat="server"
                                ImageUrl='<%# Eval("ID_Produto", "~/ImgProduto/{0}.jpg") %>' />
                            <br />
                            <br />
                            Preco:
                            <asp:Label ID="PrecoLabel" runat="server" Text='<%# Eval("Preco") %>' />
                            <br />
                            <br />
                        </ItemTemplate>
                    </asp:DataList>
                    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
                        ConnectionString="<%$ ConnectionStrings:Comercio_EletronicoBDConnectionString %>"
```

E-Commerce – Pesquisar por Categorias

- Na 1ª célula da tabela colocar um GridView e adicionar um novo SQLDataSource



Configure Data Source - SqlDataSource2

Configure the Select Statement

How would you like to retrieve data from your database?

○ Specify a custom SQL statement or stored procedure

● Specify columns from a table or view

Name: Categoria

Columns:

☑ *

□ ID_Categoria

□ Descricao

□ Return only unique rows

SELECT statement:

```
SELECT * FROM [Categoria] ORDER BY [Descricao]
```

Add ORDER BY Clause

Specify the columns you would like to order by.

Sort by: Descricao ● Ascending ○ Descending

Then by: ● Ascending ○ Descending

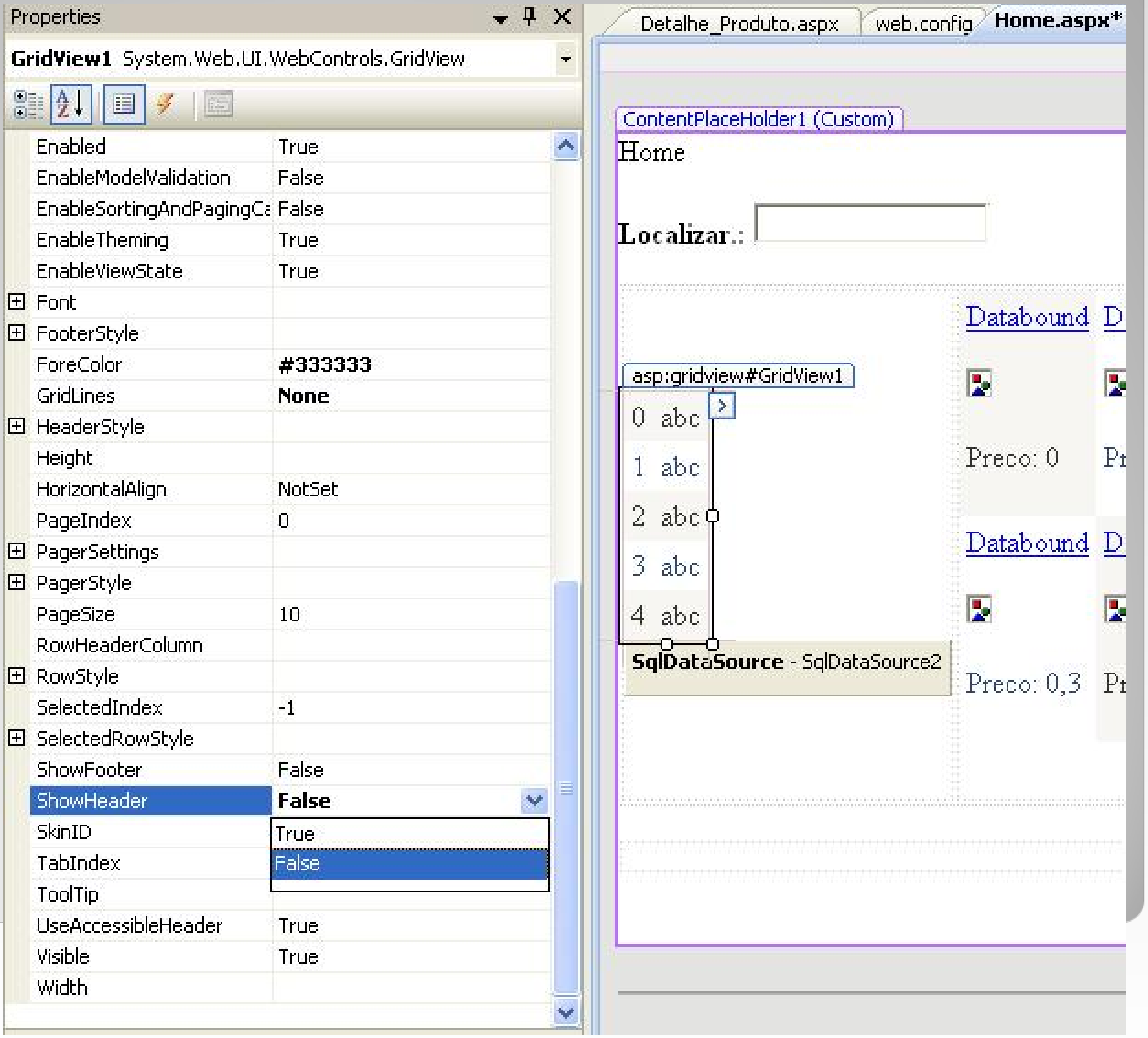
Then by: ● Ascending ○ Descending

SELECT statement:

```
SELECT * FROM [Categoria] ORDER BY [Descricao]
```

E-Commerce – Pesquisar por Categorias

- Configurar o GridView conforme a figura



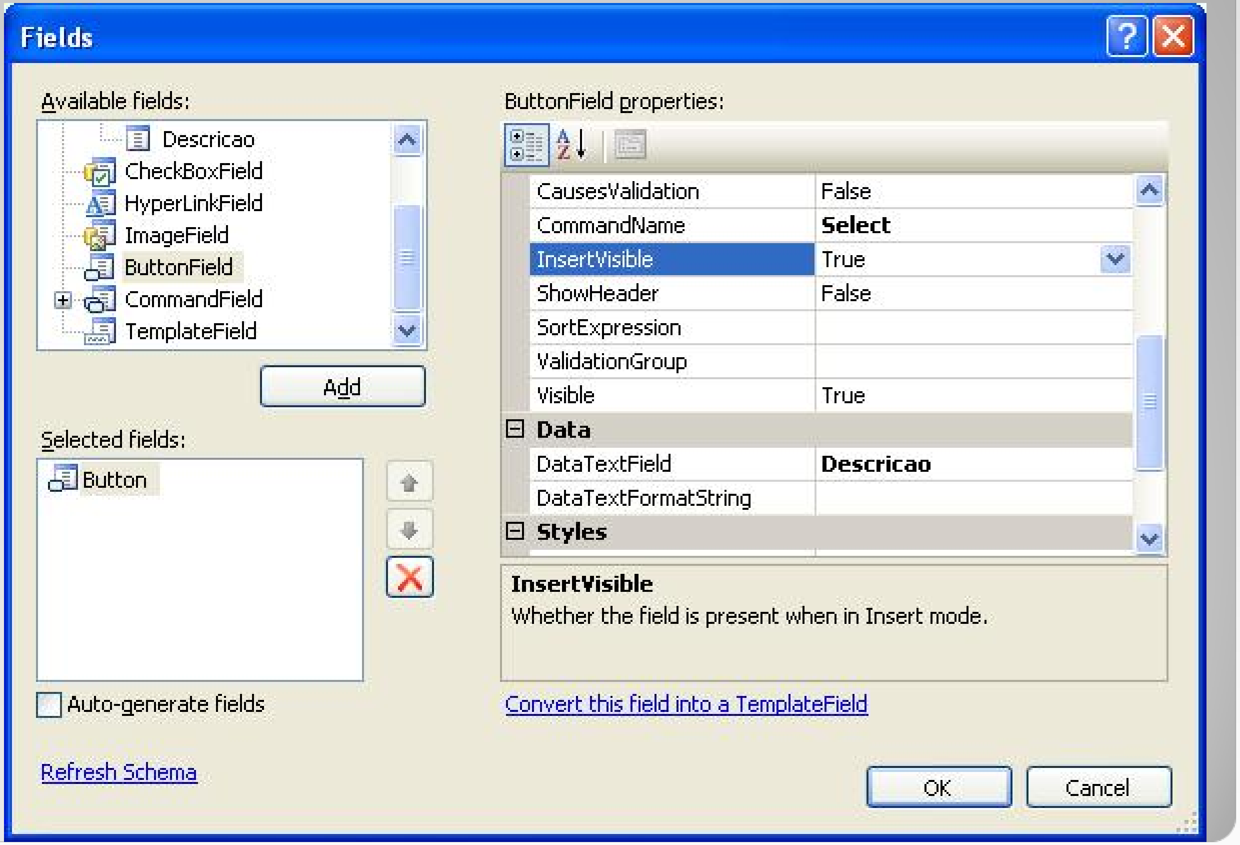
GridView1 System.Web.UI.WebControls.GridView

Enabled	True
EnableModelValidation	False
EnableSortingAndPagingCa	False
EnableTheming	True
EnableViewState	True
⊞ Font	
⊞ FooterStyle	
ForeColor	**#333333**
GridLines	**None**
⊞ HeaderStyle	
Height	
HorizontalAlign	NotSet
PageIndex	0
⊞ PagerSettings	
⊞ PagerStyle	
PageSize	10
RowHeaderColumn	
⊞ RowStyle	
SelectedIndex	-1
⊞ SelectedRowStyle	
ShowFooter	False
ShowHeader	**False**
SkinID	
TabIndex	
ToolTip	
UseAccessibleHeader	True
Visible	True
Width	

True
False

E-Commerce – Pesquisar por Categorias

- Configurar os campos do GridView conforme a figura

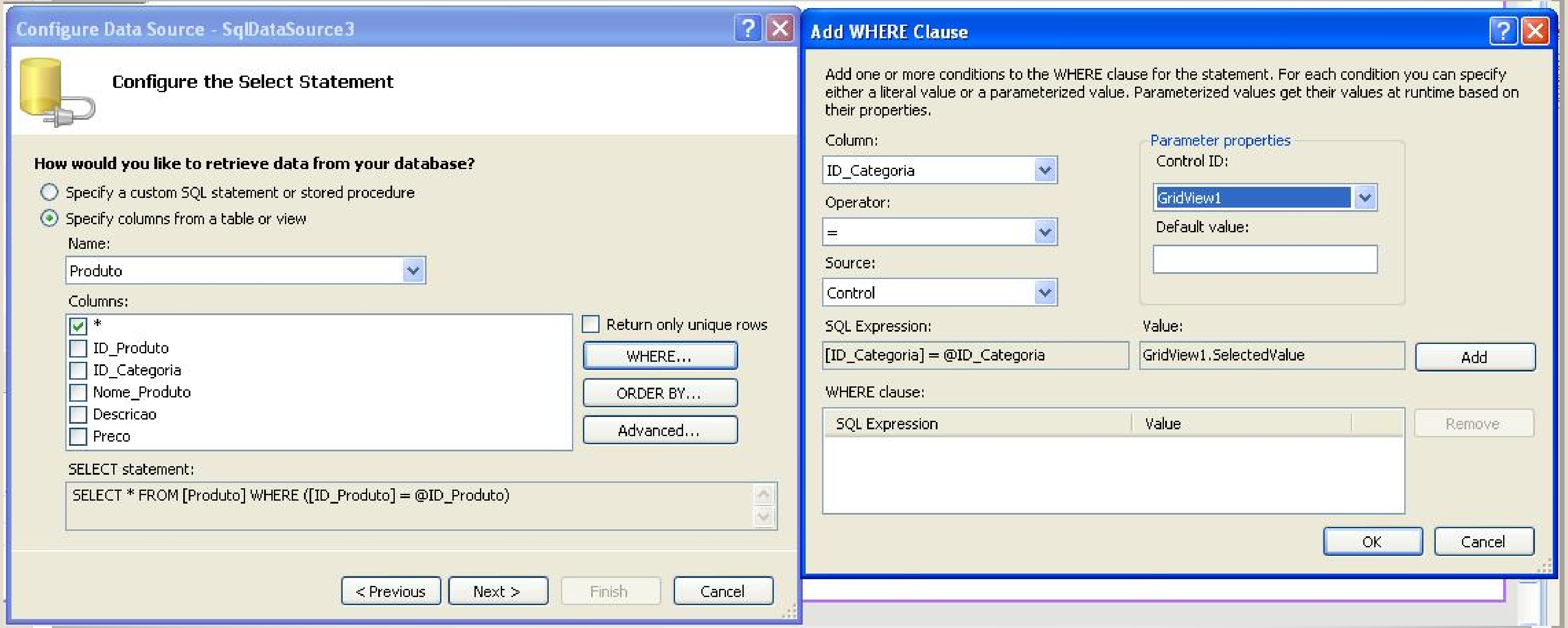


CausesValidation	False
CommandName	**Select**
InsertVisible	True
ShowHeader	False
SortExpression	
ValidationGroup	
Visible	True
Data	
DataTextField	**Descricao**
DataTextFormatString	
Styles	

InsertVisible
Whether the field is present when in Insert mode.

E-Commerce – Pesquisar por Categorias

- Adicionar um novo SQLDataSource



Configure Data Source - SqlDataSource3

Configure the Select Statement

How would you like to retrieve data from your database?

○ Specify a custom SQL statement or stored procedure

● Specify columns from a table or view

Name: Produto

Columns:

☑ *
☐ ID_Produto
☐ ID_Categoria
☐ Nome_Produto
☐ Descricao
☐ Preco

☐ Return only unique rows

SELECT statement:

```
SELECT * FROM [Produto] WHERE ([ID_Produto] = @ID_Produto)
```

Add WHERE Clause

Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column: ID_Categoria

Operator: =

Source: Control

Parameter properties

Control ID: GridView1

Default value:

SQL Expression: [ID_Categoria] = @ID_Categoria

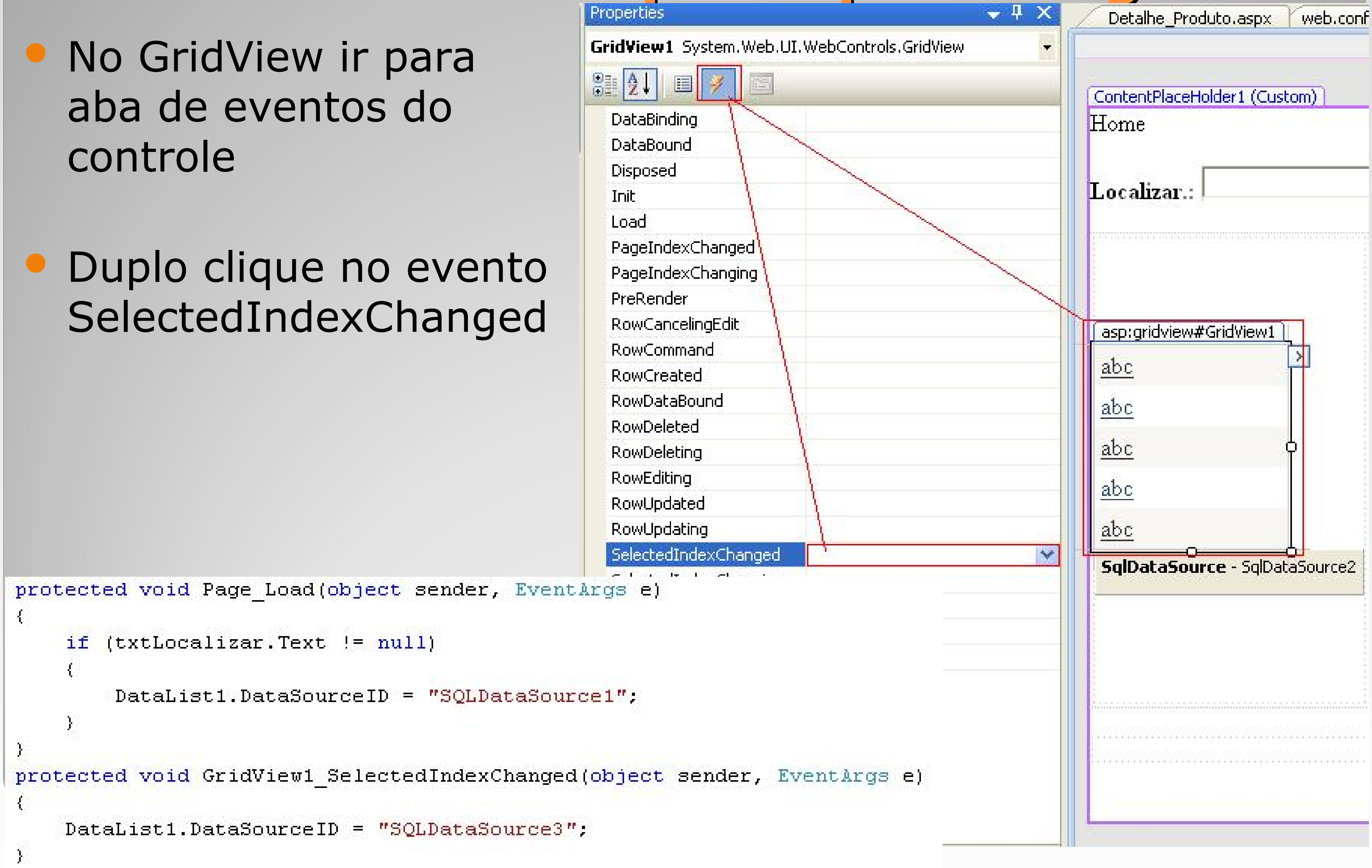
Value: GridView1.SelectedValue

WHERE clause:

SQL Expression	Value

E-Commerce – Pesquisar por Categorias

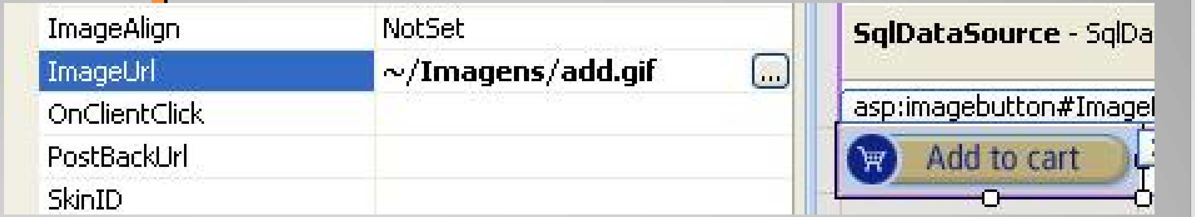
- No GridView ir para aba de eventos do controle
- Duplo clique no evento SelectedIndexChanged



```
protected void Page_Load(object sender, EventArgs e)
{
    if (txtLocalizar.Text != null)
    {
        DataList1.DataSourceID = "SQLDataSource1";
    }
}
protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
    DataList1.DataSourceID = "SQLDataSource3";
}
```

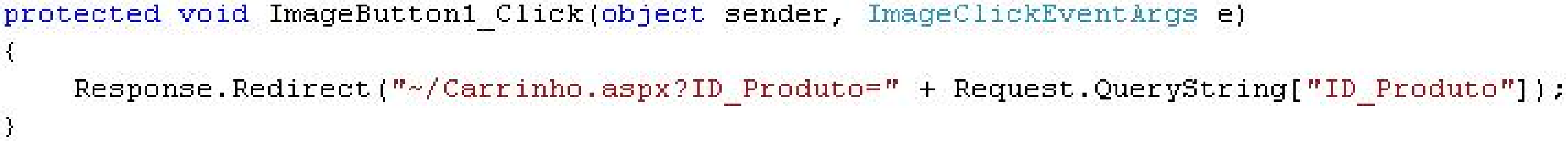
E-Commerce – Adicionado Produto ao Carrinho de compras

- Inserir um ImageButton na página Detalhe_Produto.aspx



ImageAlign	NotSet
ImageUrl	~/Imagens/add.gif
OnClientClick	
PostBackUrl	
SkinID	

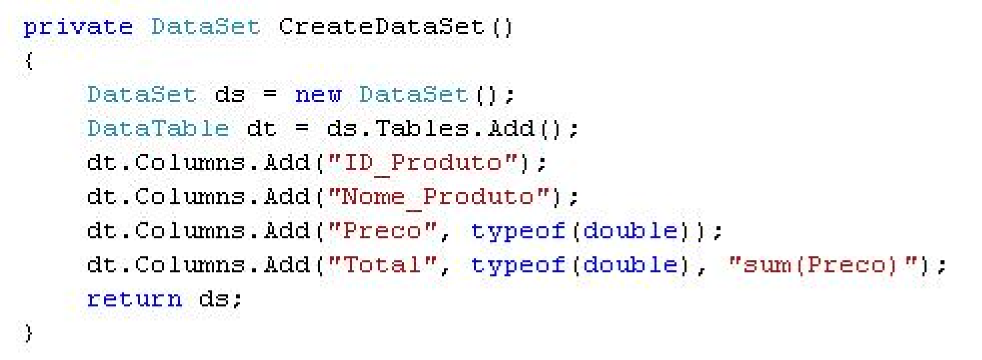
- Programar evento Click



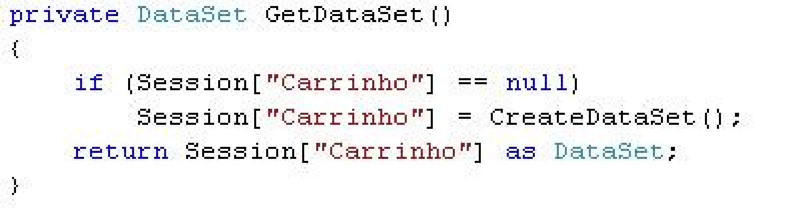
```
protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect("~/Carrinho.aspx?ID_Produto=" + Request.QueryString["ID_Produto"]);
}
```

E-Commerce – Carrinho de Compras

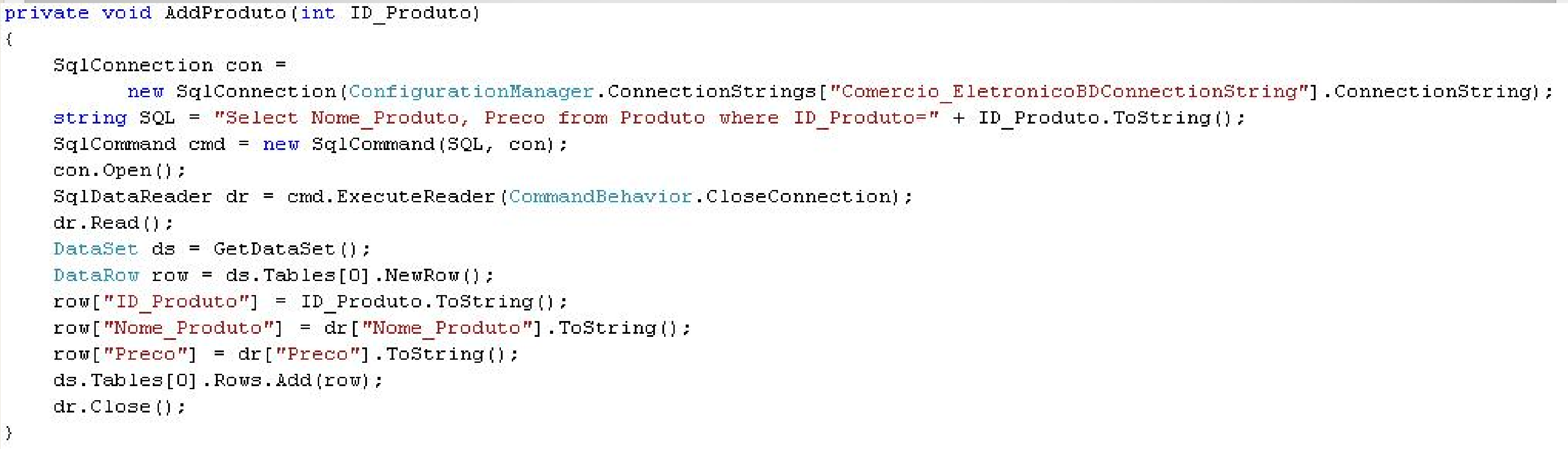
- Criar a página Carrinho.aspx
  - Criar os métodos:
    - CreateDataSet()
    - GetDataSet()
    - AddProduto()



```
private DataSet CreateDataSet()
{
    DataSet ds = new DataSet();
    DataTable dt = ds.Tables.Add();
    dt.Columns.Add("ID_Produto");
    dt.Columns.Add("Nome_Produto");
    dt.Columns.Add("Preco", typeof(double));
    dt.Columns.Add("Total", typeof(double), "sum(Preco)");
    return ds;
}
```



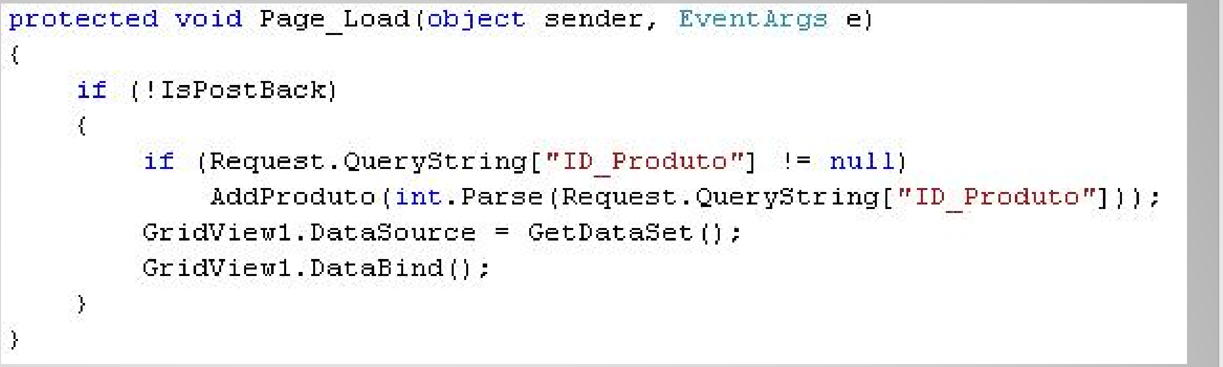
```
private DataSet GetDataSet()
{
    if (Session["Carrinho"] == null)
        Session["Carrinho"] = CreateDataSet();
    return Session["Carrinho"] as DataSet;
}
```



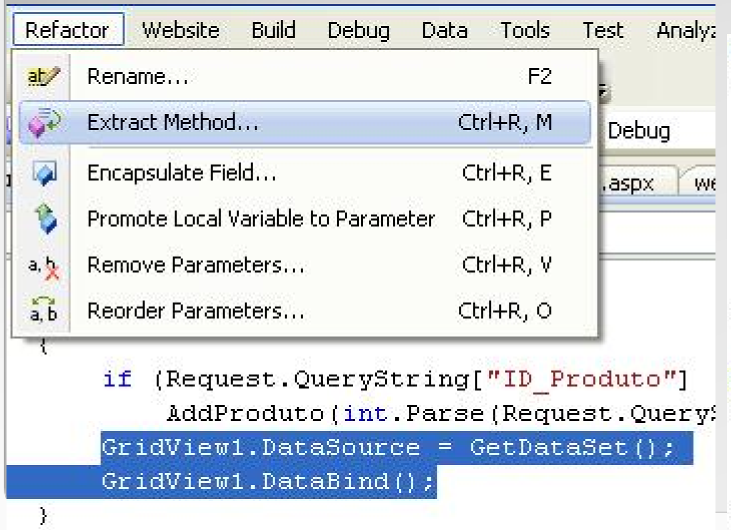
```
private void AddProduto(int ID_Produto)
{
    SqlConnection con =
          new SqlConnection(ConfigurationManager.ConnectionStrings["Comercio_EletronicoBDConnectionString"].ConnectionString);
    string SQL = "Select Nome_Produto, Preco from Produto where ID_Produto=" + ID_Produto.ToString();
    SqlCommand cmd = new SqlCommand(SQL, con);
    con.Open();
    SqlDataReader dr = cmd.ExecuteReader(CommandBehavior.CloseConnection);
    dr.Read();
    DataSet ds = GetDataSet();
    DataRow row = ds.Tables[0].NewRow();
    row["ID_Produto"] = ID_Produto.ToString();
    row["Nome_Produto"] = dr["Nome_Produto"].ToString();
    row["Preco"] = dr["Preco"].ToString();
    ds.Tables[0].Rows.Add(row);
    dr.Close();
}
```

E-Commerce – Carrinho de Compras

- Programar o método PageLoad

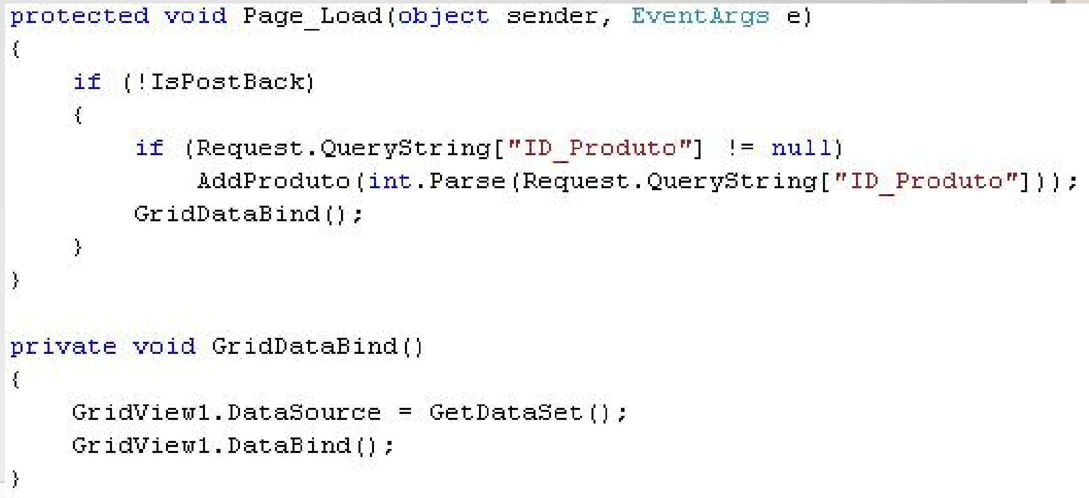


```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        if (Request.QueryString["ID_Produto"] != null)
            AddProduto(int.Parse(Request.QueryString["ID_Produto"]));
        GridView1.DataSource = GetDataSet();
        GridView1.DataBind();
    }
}
```



Refactor
Website
Build
Debug
Data
Tools
Test
Analyz
Rename... F2
Extract Method... Ctrl+R, M
Encapsulate Field... Ctrl+R, E
Promote Local Variable to Parameter Ctrl+R, P
Remove Parameters... Ctrl+R, V
Reorder Parameters... Ctrl+R, O
Debug
.aspx
we

```
{
    if (Request.QueryString["ID_Produto"]
        AddProduto(int.Parse(Request.Query
    GridView1.DataSource = GetDataSet();
    GridView1.DataBind();
}
```

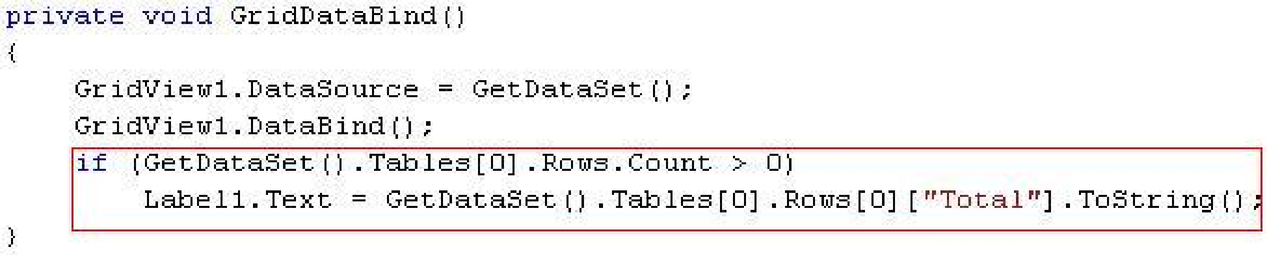


```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        if (Request.QueryString["ID_Produto"] != null)
            AddProduto(int.Parse(Request.QueryString["ID_Produto"]));
        GridDataBind();
    }
}

private void GridDataBind()
{
    GridView1.DataSource = GetDataSet();
    GridView1.DataBind();
}
```

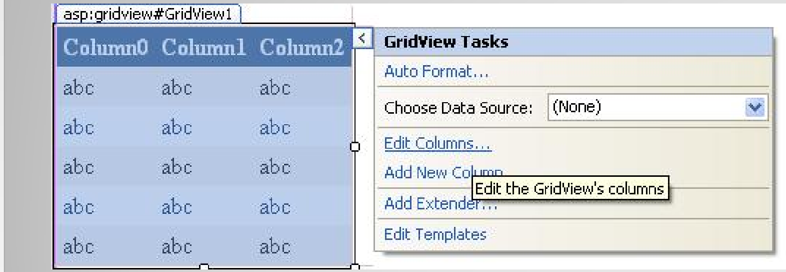
E-Commerce – Carrinho de Compras

- No método GridDataBind()



```
private void GridDataBind()
{
    GridView1.DataSource = GetDataSet();
    GridView1.DataBind();
    if (GetDataSet().Tables[0].Rows.Count > 0)
        Label1.Text = GetDataSet().Tables[0].Rows[0]["Total"].ToString();
}
```

- No GridView



asp:gridview#GridView1

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

GridView Tasks

Auto Format...

Choose Data Source: (None)

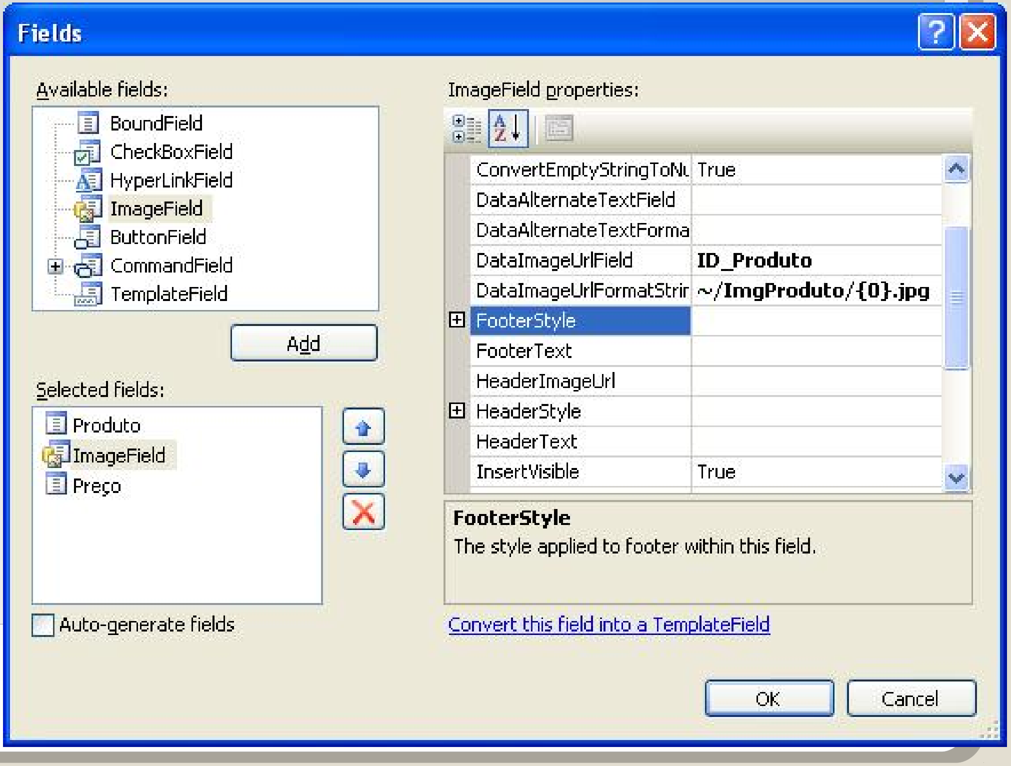
Edit Columns...

Add New Column...

Edit the GridView's columns

Add Extender...

Edit Templates



Fields

Available fields:

- BoundField
- CheckBoxField
- HyperLinkField
- ImageField
- ButtonField
- CommandField
- TemplateField

Add

Selected fields:

- Produto
- ImageField
- Preço

□ Auto-generate fields

ImageField properties:

ConvertEmptyStringToNu	True
DataAlternateTextField	
DataAlternateTextForma	
DataImageUrlField	**ID_Produto**
DataImageUrlFormatStrir	**~/ImgProduto/{0}.jpg**
⊞ FooterStyle	
FooterText	
HeaderImageUrl	
⊞ HeaderStyle	
HeaderText	
InsertVisible	True

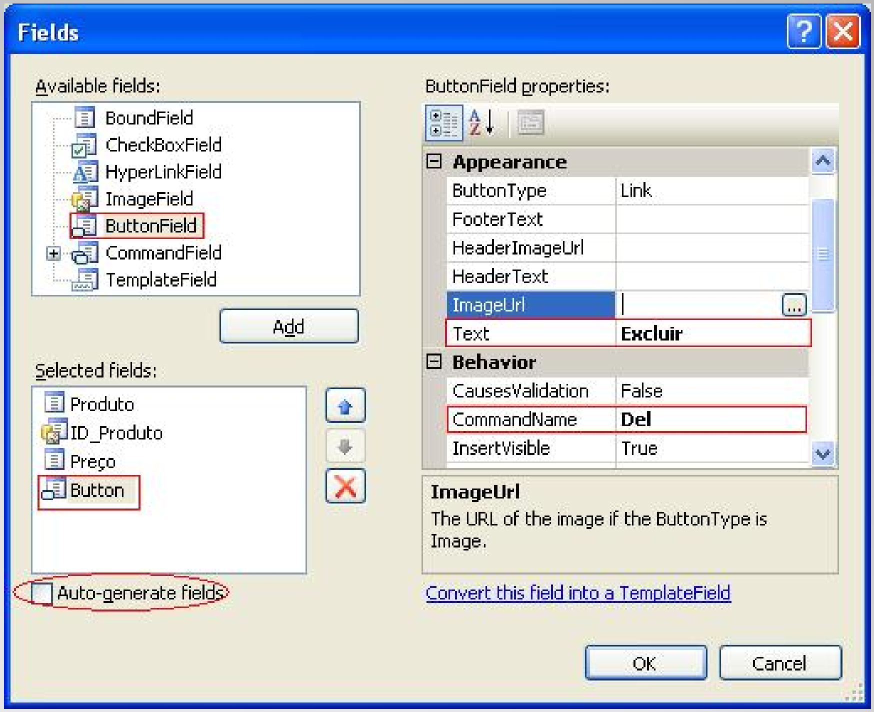
FooterStyle
The style applied to footer within this field.

Convert this field into a TemplateField

OK Cancel

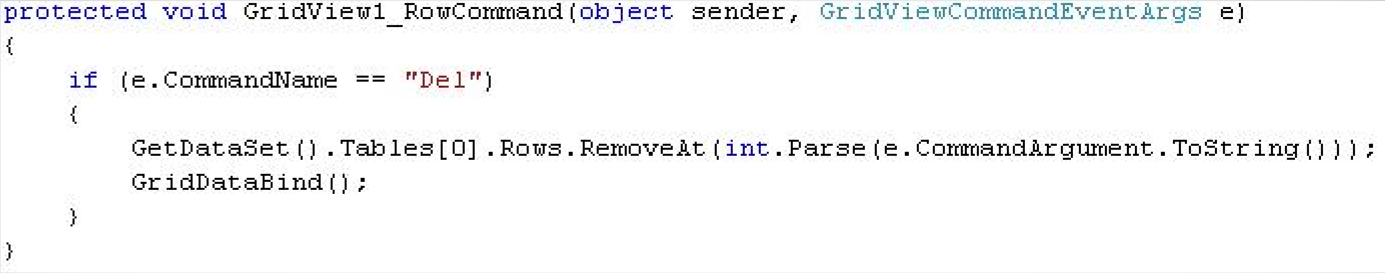
E-Commerce – Carrinho de Compras

- No GridView



⊟ **Appearance**	
ButtonType	Link
FooterText	
HeaderImageUrl	
HeaderText	
ImageUrl	
Text	**Excluir**
⊟ **Behavior**	
CausesValidation	False
CommandName	**Del**
InsertVisible	True

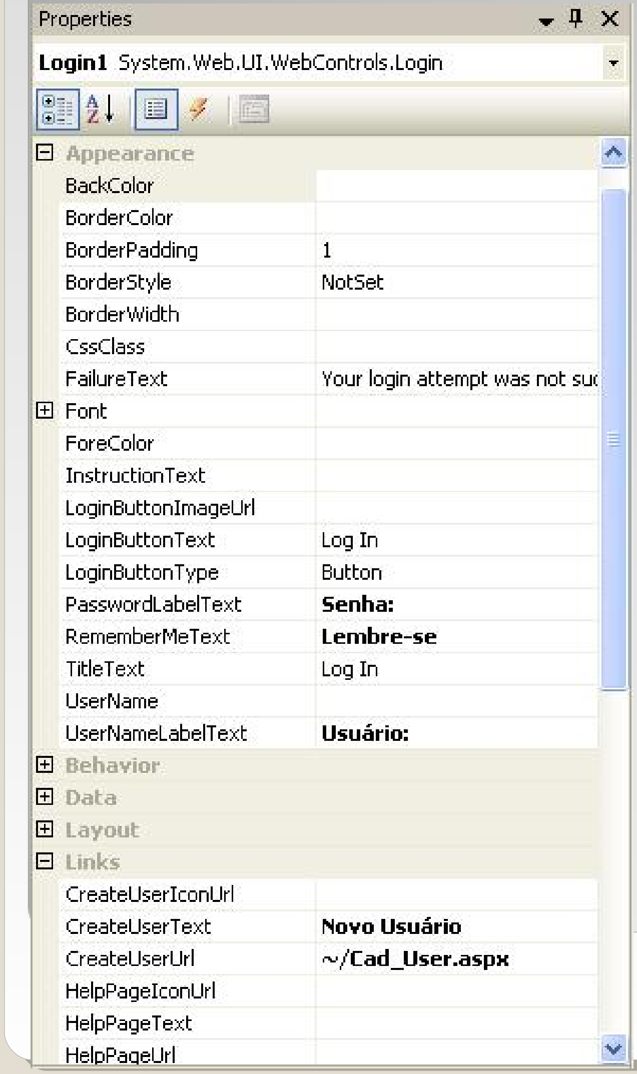
ImageUrl
The URL of the image if the ButtonType is Image.



```
protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if (e.CommandName == "Del")
    {
        GetDataSet().Tables[0].Rows.RemoveAt(int.Parse(e.CommandArgument.ToString()));
        GridDataBind();
    }
}
```

E-Commerce – Login

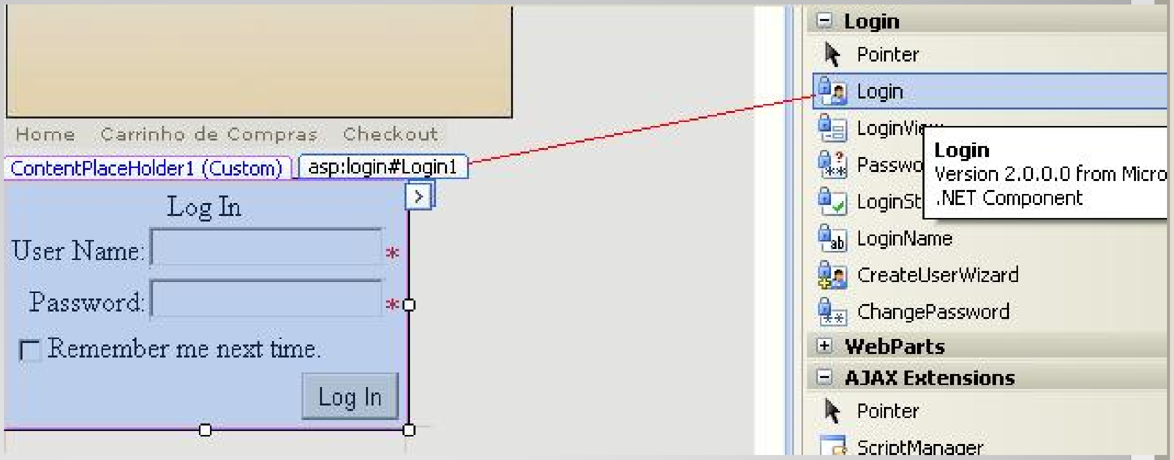
- Criar página Login.aspx



Properties

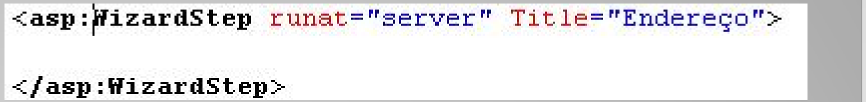
Login1 System.Web.UI.WebControls.Login

⊟ Appearance	
BackColor	
BorderColor	
BorderPadding	1
BorderStyle	NotSet
BorderWidth	
CssClass	
FailureText	Your login attempt was not suc
⊞ Font	
ForeColor	
InstructionText	
LoginButtonImageUrl	
LoginButtonText	Log In
LoginButtonType	Button
PasswordLabelText	**Senha:**
RememberMeText	**Lembre-se**
TitleText	Log In
UserName	
UserNameLabelText	**Usuário:**
⊞ Behavior	
⊞ Data	
⊞ Layout	
⊟ Links	
CreateUserIconUrl	
CreateUserText	**Novo Usuário**
CreateUserUrl	**~/Cad_User.aspx**
HelpPageIconUrl	
HelpPageText	
HelpPageUrl	



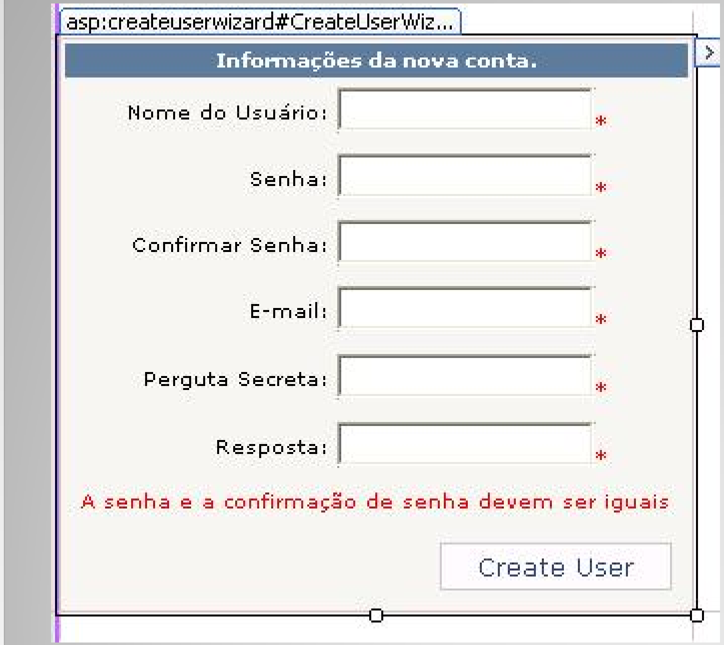
E-Commerce – Criando Usuários

- Criar página Cad_User.aspx



```
<asp:WizardStep runat="server" Title="Endereço">

</asp:WizardStep>
```



asp:createuserwizard#CreateUserWiz...

Informações da nova conta.

Nome do Usuário: *

Senha: *

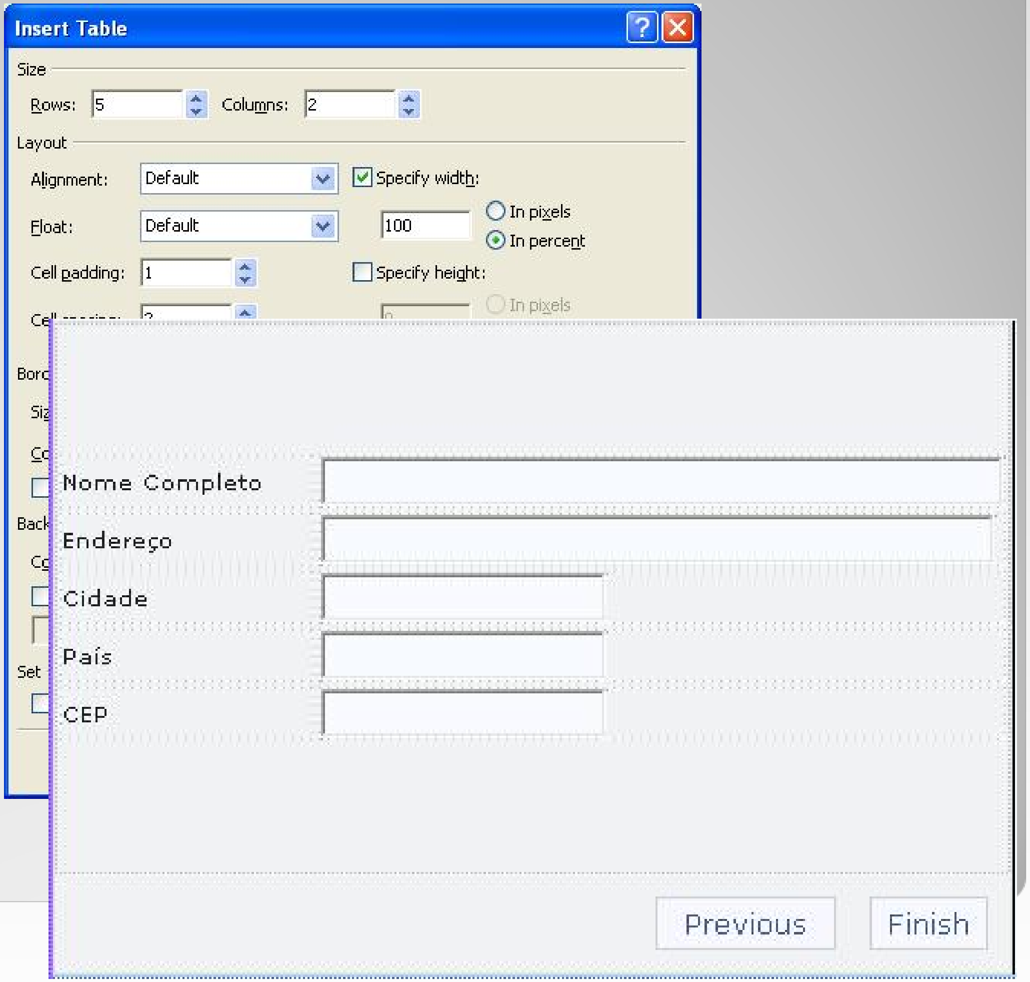
Confirmar Senha: *

E-mail: *

Perguta Secreta: *

Resposta: *

A senha e a confirmação de senha devem ser iguais



Insert Table

Size

Rows: 5 Columns: 2

Layout

Alignment: Default ☑ Specify width:

Float: Default 100 ○ In pixels ● In percent

Cell padding: 1 □ Specify height:

○ In pixels

Nome Completo

Endereço

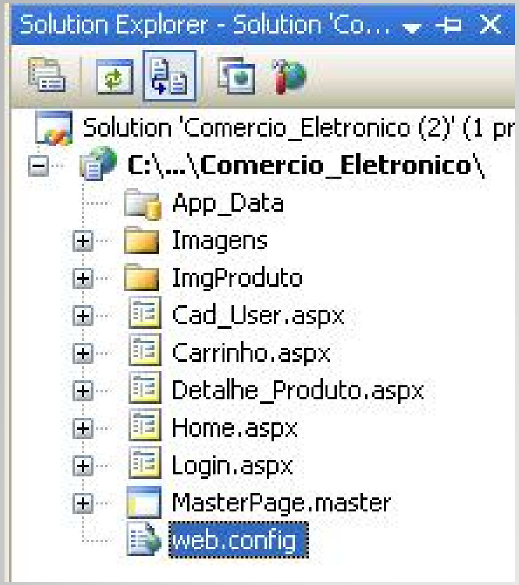
Cidade

País

CEP

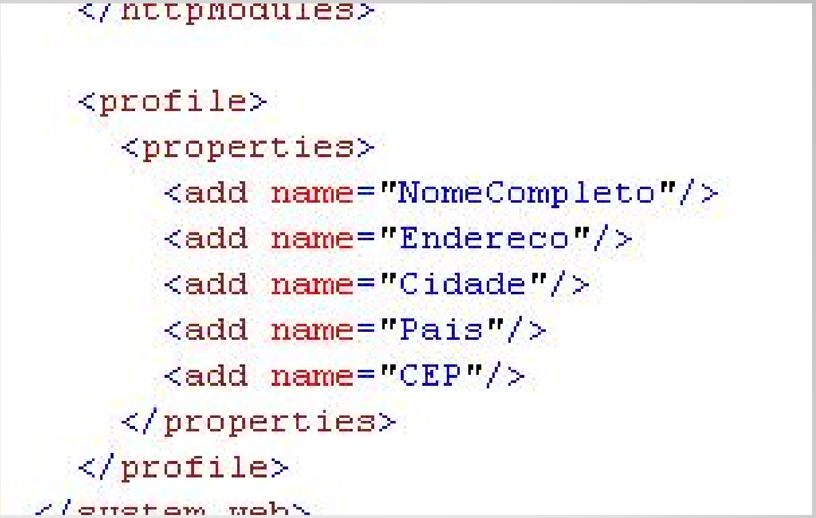
E-Commerce – Criando Usuários (Profile)

- Abrir arquivo web.config



Solution Explorer - Solution 'Co...

```
Solution 'Comercio_Eletronico (2)' (1 pr
⊟ C:\...\Comercio_Eletronico\
     App_Data
  ⊞  Imagens
  ⊞  ImgProduto
  ⊞  Cad_User.aspx
  ⊞  Carrinho.aspx
  ⊞  Detalhe_Produto.aspx
  ⊞  Home.aspx
  ⊞  Login.aspx
  ⊞  MasterPage.master
     web.config
```

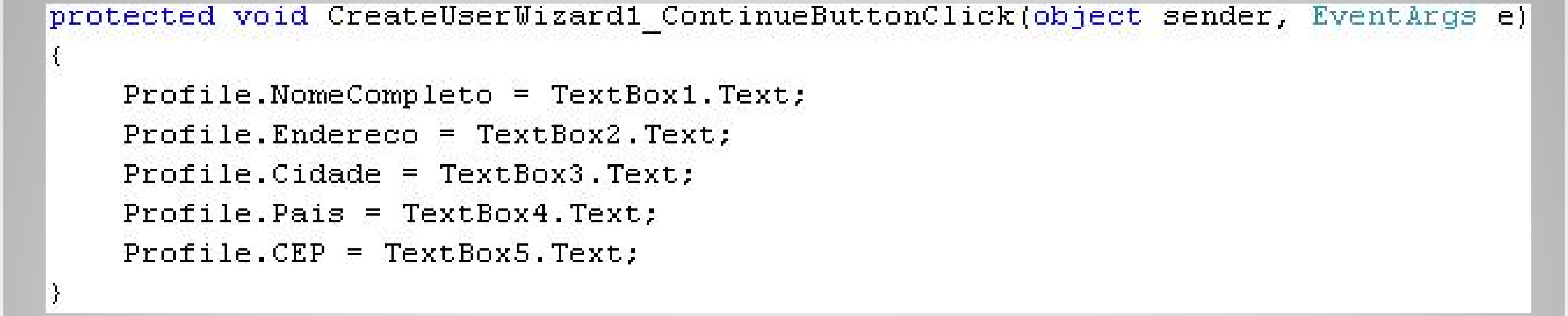


```
    </httpModules>

    <profile>
      <properties>
        <add name="NomeCompleto"/>
        <add name="Endereco"/>
        <add name="Cidade"/>
        <add name="Pais"/>
        <add name="CEP"/>
      </properties>
    </profile>
  </system.web>
```

E-Commerce – Criando Usuários

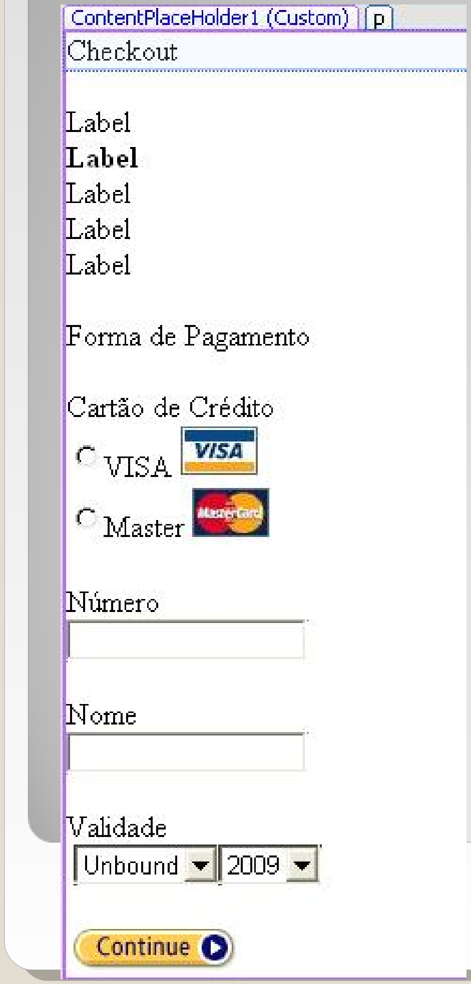
- No controle CreateUserWizard programar o evento ContinueButtonClick



```
protected void CreateUserWizard1_ContinueButtonClick(object sender, EventArgs e)
{
    Profile.NomeCompleto = TextBox1.Text;
    Profile.Endereco = TextBox2.Text;
    Profile.Cidade = TextBox3.Text;
    Profile.Pais = TextBox4.Text;
    Profile.CEP = TextBox5.Text;
}
```

E-Commerce – Fechamento do Pedido

- Criar uma pasta chamada Checkout
  - Dentro da pasta criar a página Checkout.aspx



Checkout

Label
Label
Label
Label
Label

Forma de Pagamento

Cartão de Crédito

○ VISA

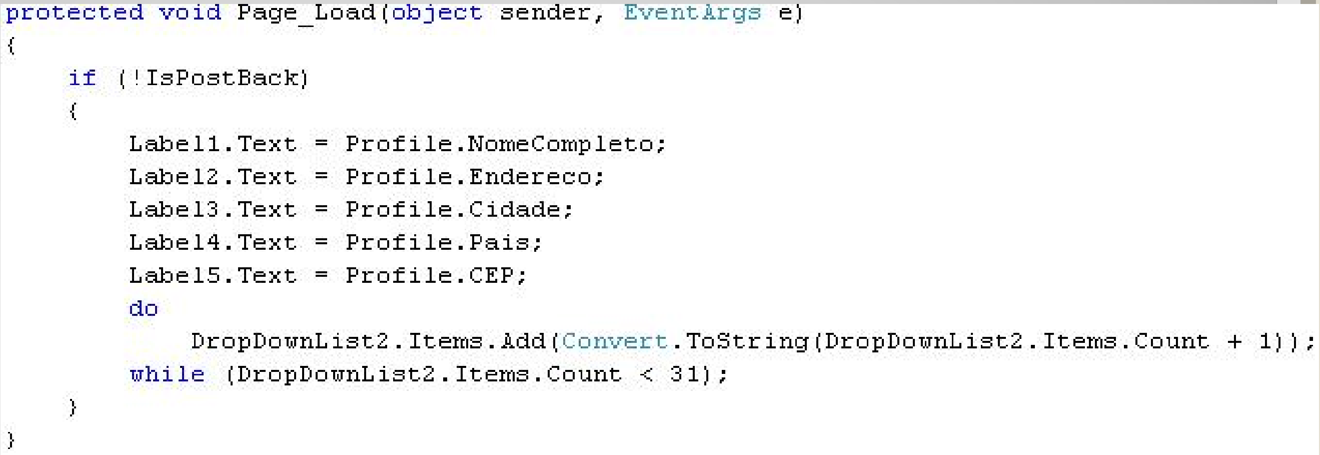
○ Master

Número

Nome

Validade

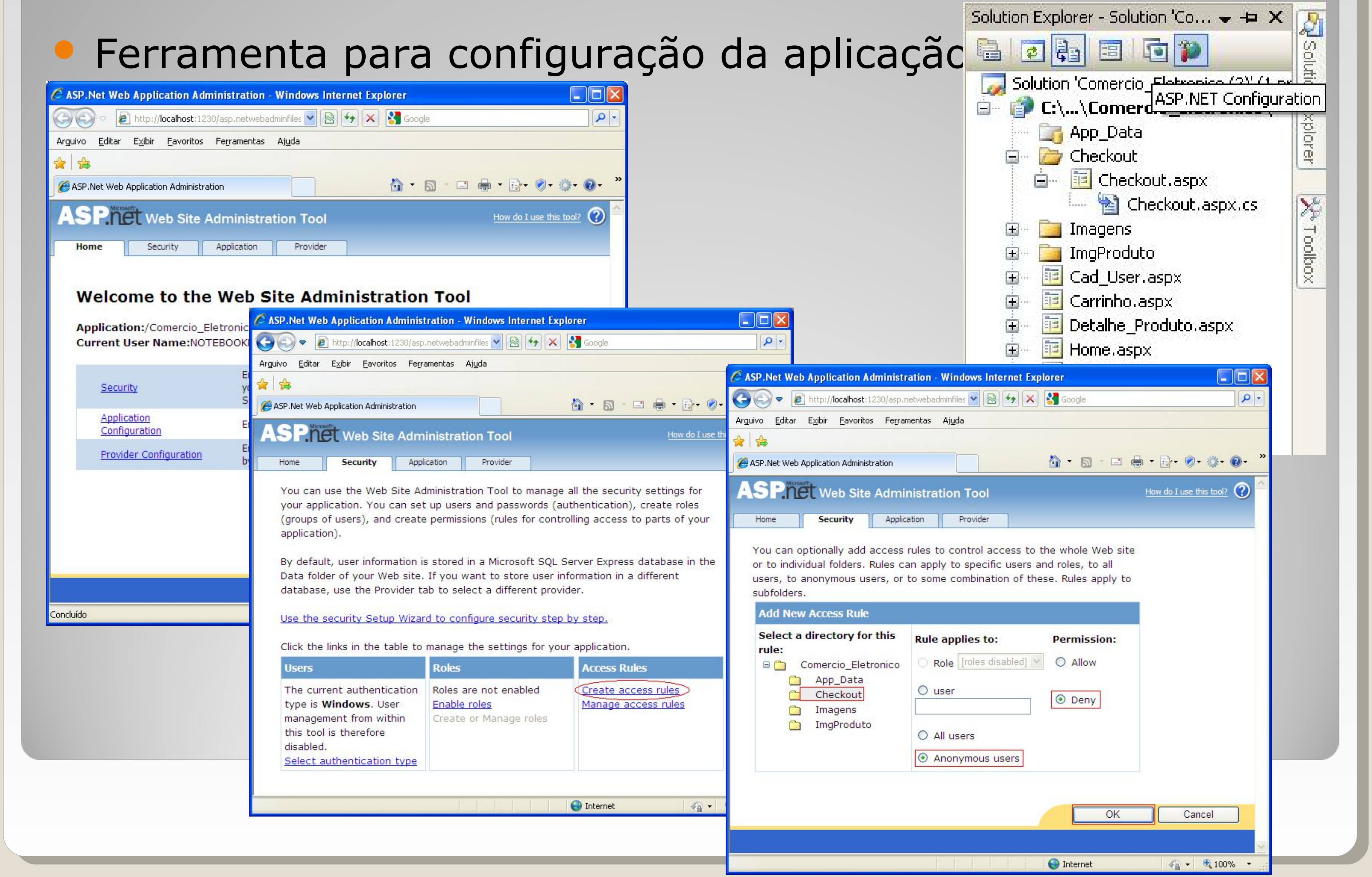
Unbound ▾ 2009 ▾



```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        Label1.Text = Profile.NomeCompleto;
        Label2.Text = Profile.Endereco;
        Label3.Text = Profile.Cidade;
        Label4.Text = Profile.Pais;
        Label5.Text = Profile.CEP;
        do
            DropDownList2.Items.Add(Convert.ToString(DropDownList2.Items.Count + 1));
        while (DropDownList2.Items.Count < 31);
    }
}
```

E-Commerce – ASP.Net Configuration

- Ferramenta para configuração da aplicaçã



Welcome to the Web Site Administration Tool

Application:/Comercio_Eletronic
Current User Name:NOTEBOOK

Security

Application Configuration

Provider Configuration

You can use the Web Site Administration Tool to manage all the security settings for your application. You can set up users and passwords (authentication), create roles (groups of users), and create permissions (rules for controlling access to parts of your application).

By default, user information is stored in a Microsoft SQL Server Express database in the Data folder of your Web site. If you want to store user information in a different database, use the Provider tab to select a different provider.

Use the security Setup Wizard to configure security step by step.

Click the links in the table to manage the settings for your application.

Users	Roles	Access Rules
The current authentication type is **Windows**. User management from within this tool is therefore disabled. Select authentication type	Roles are not enabled Enable roles Create or Manage roles	Create access rules Manage access rules

You can optionally add access rules to control access to the whole Web site or to individual folders. Rules can apply to specific users and roles, to all users, to anonymous users, or to some combination of these. Rules apply to subfolders.

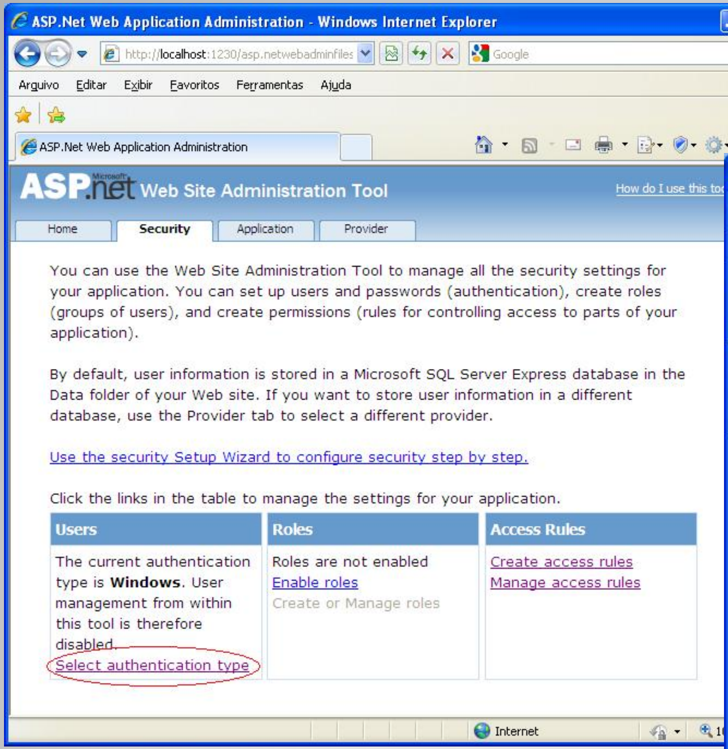
Add New Access Rule

Select a directory for this rule:	Rule applies to:	Permission:
Comercio_Eletronico App_Data Checkout Imagens ImgProduto	○ Role [roles disabled] ○ user ○ All users ● Anonymous users	○ Allow ● Deny

OK Cancel

E-Commerce – ASP.Net Configuration

- Ferramenta para configuração da aplicação


ASP.Net Web Application Administration - Windows Internet Explorer
http://localhost:1230/asp.netwebadminfiles
Google
Arquivo Editar Exibir Favoritos Ferramentas Ajuda
ASP.Net Web Application Administration
ASP.net Web Site Administration Tool
How do I use this to
Home
Security
Application
Provider
Internet

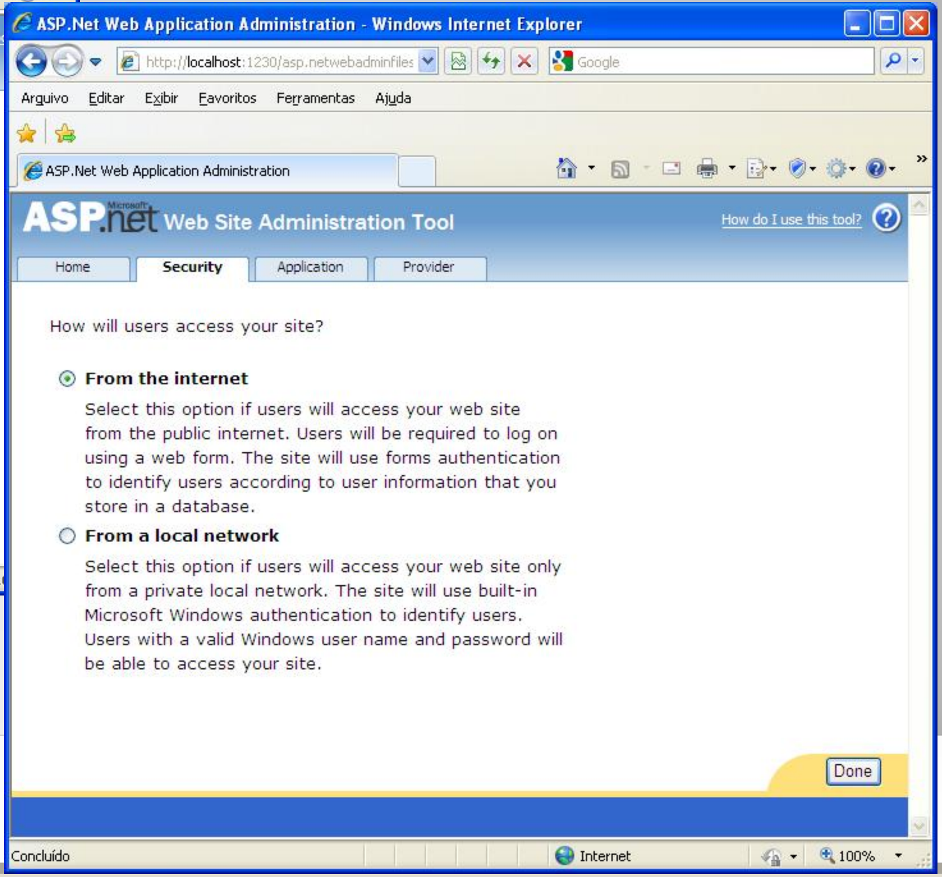
You can use the Web Site Administration Tool to manage all the security settings for your application. You can set up users and passwords (authentication), create roles (groups of users), and create permissions (rules for controlling access to parts of your application).

By default, user information is stored in a Microsoft SQL Server Express database in the Data folder of your Web site. If you want to store user information in a different database, use the Provider tab to select a different provider.

Use the security Setup Wizard to configure security step by step.

Click the links in the table to manage the settings for your application.

Users	Roles	Access Rules
The current authentication type is **Windows**. User management from within this tool is therefore disabled. Select authentication type	Roles are not enabled Enable roles Create or Manage roles	Create access rules Manage access rules


ASP.Net Web Application Administration - Windows Internet Explorer
http://localhost:1230/asp.netwebadminfiles
Google
Arquivo Editar Exibir Favoritos Ferramentas Ajuda
ASP.Net Web Application Administration
ASP.net Web Site Administration Tool
How do I use this tool?
Home
Security
Application
Provider
Done
Concluído
Internet
100%

How will users access your site?

● **From the internet**

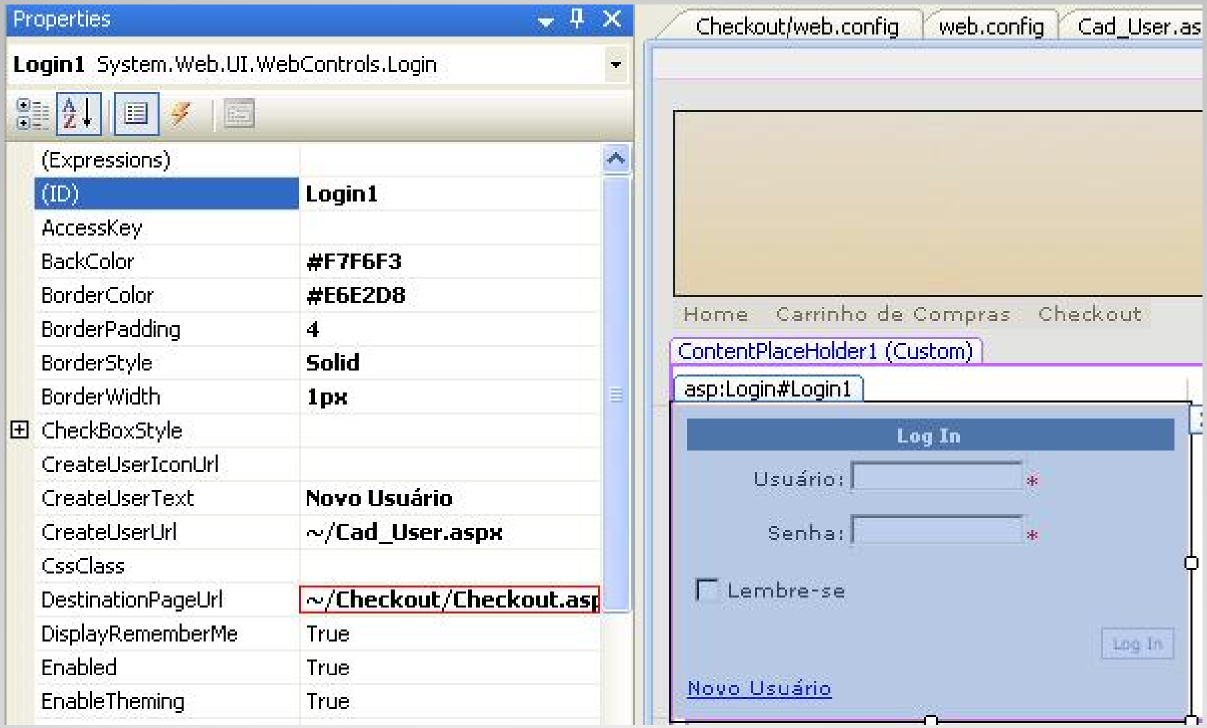
Select this option if users will access your web site from the public internet. Users will be required to log on using a web form. The site will use forms authentication to identify users according to user information that you store in a database.

○ **From a local network**

Select this option if users will access your web site only from a private local network. The site will use built-in Microsoft Windows authentication to identify users. Users with a valid Windows user name and password will be able to access your site.

E-Commerce – Últimos Detalhes

- Na página Login.aspx



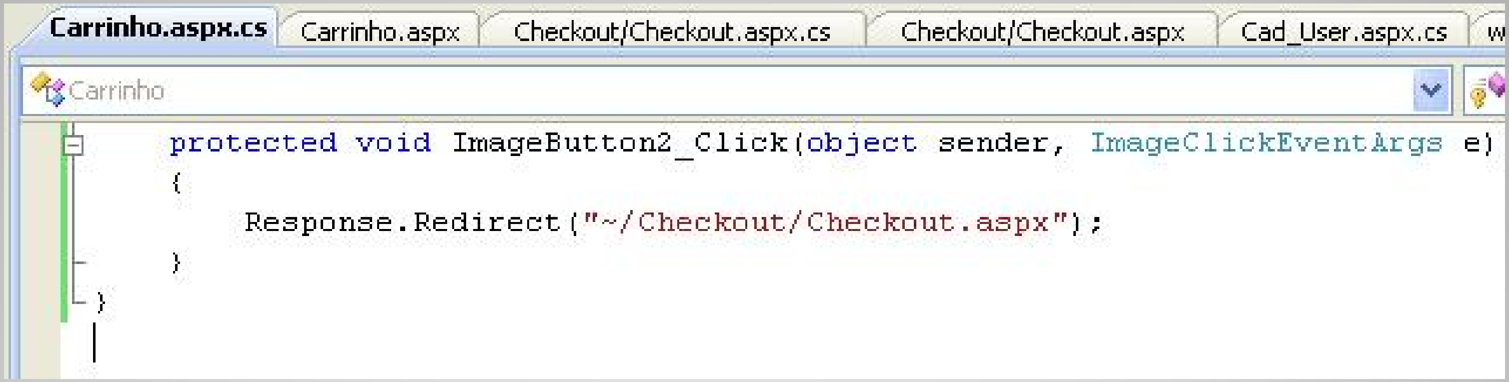
Properties

Login1 System.Web.UI.WebControls.Login

(Expressions)	
(ID)	Login1
AccessKey	
BackColor	#F7F6F3
BorderColor	#E6E2D8
BorderPadding	4
BorderStyle	Solid
BorderWidth	1px
⊞ CheckBoxStyle	
CreateUserIconUrl	
CreateUserText	Novo Usuário
CreateUserUrl	~/Cad_User.aspx
CssClass	
DestinationPageUrl	~/Checkout/Checkout.asp
DisplayRememberMe	True
Enabled	True
EnableTheming	True

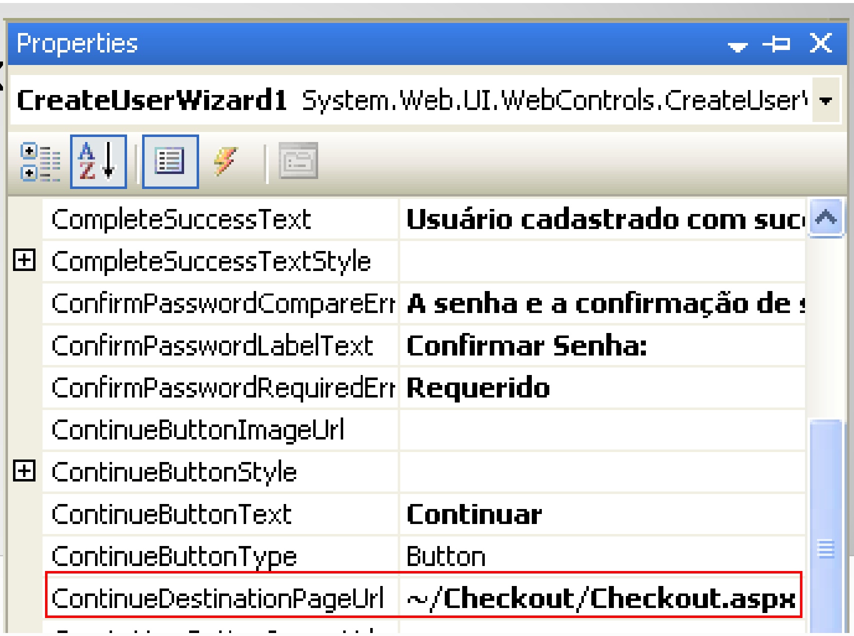
E-Commerce – Últimos Detalhes

- Na página Carrinho.aspx



```
    protected void ImageButton2_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect("~/Checkout/Checkout.aspx");
    }
}
```

- Na página Cad_User.aspx

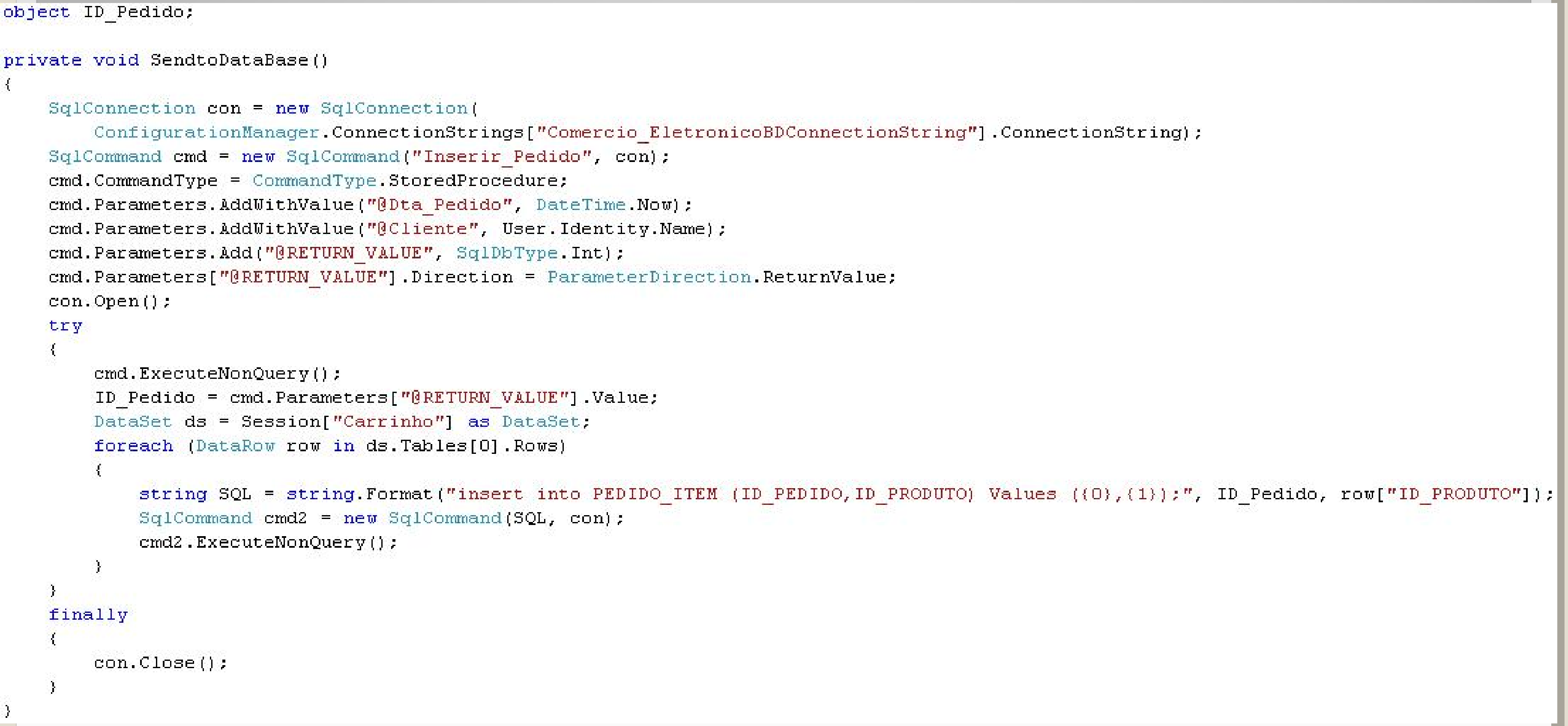


CreateUserWizard1 System.Web.UI.WebControls.CreateUserV

CompleteSuccessText	Usuário cadastrado com suc
⊞ CompleteSuccessTextStyle	
ConfirmPasswordCompareErr	A senha e a confirmação de s
ConfirmPasswordLabelText	Confirmar Senha:
ConfirmPasswordRequiredErr	Requerido
ContinueButtonImageUrl	
⊞ ContinueButtonStyle	
ContinueButtonText	Continuar
ContinueButtonType	Button
ContinueDestinationPageUrl	~/Checkout/Checkout.aspx

E-Commerce – Fechamento do Pedido

- Persistência do Pedido no Banco de Dados

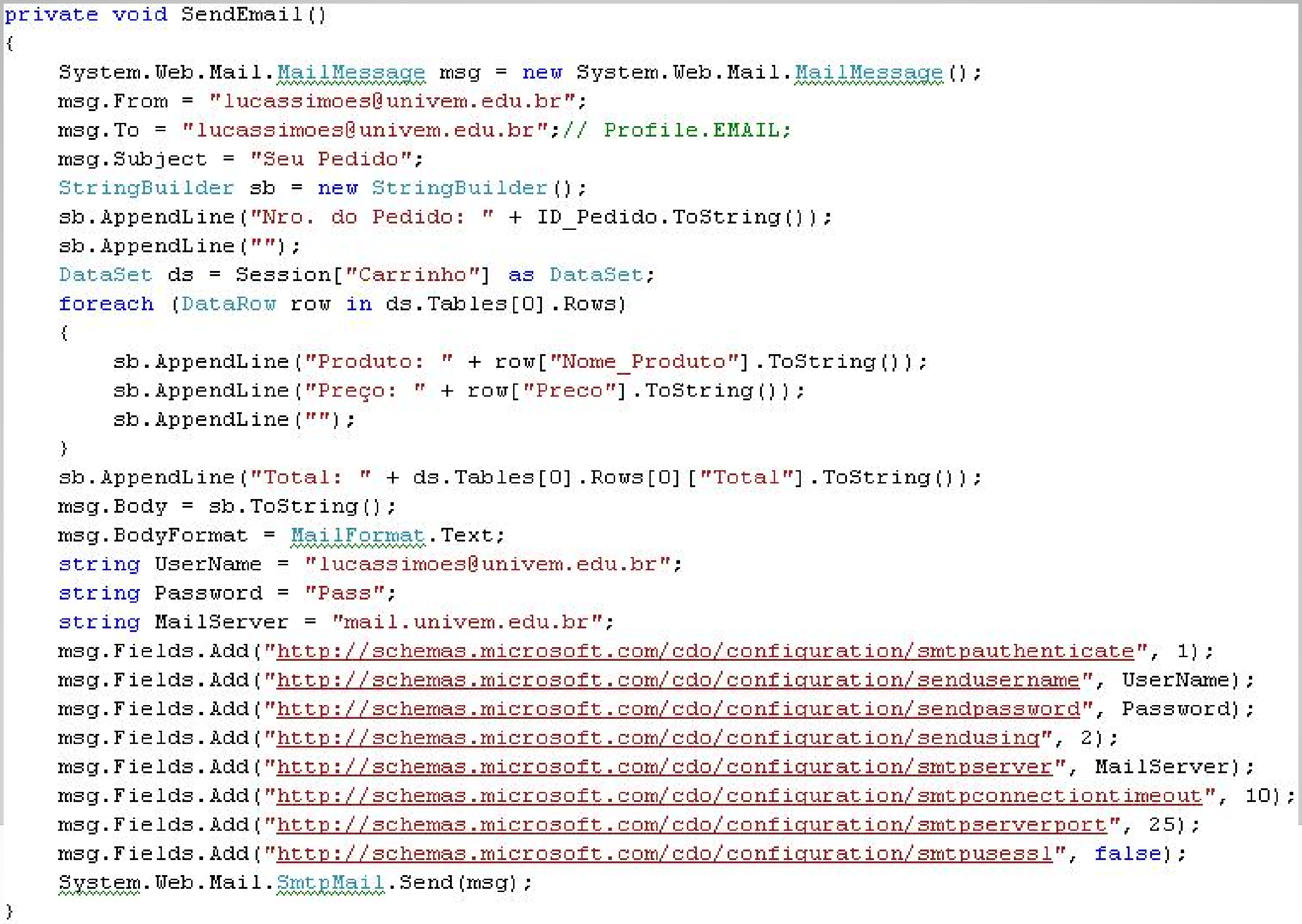


```
object ID_Pedido;

private void SendtoDataBase()
{
    SqlConnection con = new SqlConnection(
        ConfigurationManager.ConnectionStrings["Comercio_EletronicoBDConnectionString"].ConnectionString);
    SqlCommand cmd = new SqlCommand("Inserir_Pedido", con);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("@Dta_Pedido", DateTime.Now);
    cmd.Parameters.AddWithValue("@Cliente", User.Identity.Name);
    cmd.Parameters.Add("@RETURN_VALUE", SqlDbType.Int);
    cmd.Parameters["@RETURN_VALUE"].Direction = ParameterDirection.ReturnValue;
    con.Open();
    try
    {
        cmd.ExecuteNonQuery();
        ID_Pedido = cmd.Parameters["@RETURN_VALUE"].Value;
        DataSet ds = Session["Carrinho"] as DataSet;
        foreach (DataRow row in ds.Tables[0].Rows)
        {
            string SQL = string.Format("insert into PEDIDO_ITEM (ID_PEDIDO,ID_PRODUTO) Values ({0},{1});", ID_Pedido, row["ID_PRODUTO"]);
            SqlCommand cmd2 = new SqlCommand(SQL, con);
            cmd2.ExecuteNonQuery();
        }
    }
    finally
    {
        con.Close();
    }
}
```

E-Commerce – Fechamento do Pedido

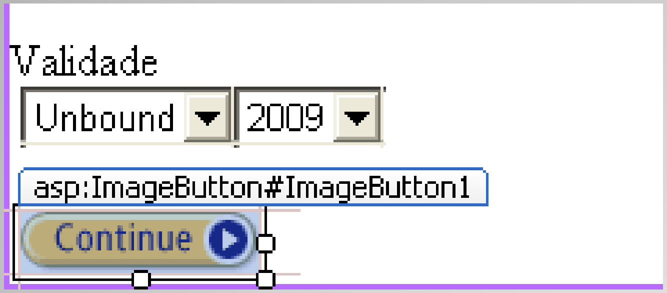
- Enviando Pedido por email Email

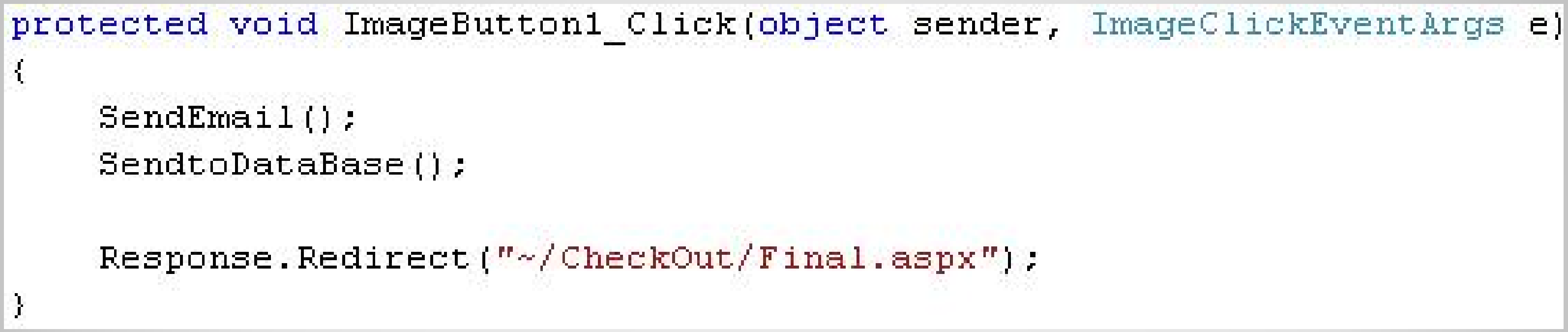


```
private void SendEmail()
{
    System.Web.Mail.MailMessage msg = new System.Web.Mail.MailMessage();
    msg.From = "lucassimoes@univem.edu.br";
    msg.To = "lucassimoes@univem.edu.br";// Profile.EMAIL;
    msg.Subject = "Seu Pedido";
    StringBuilder sb = new StringBuilder();
    sb.AppendLine("Nro. do Pedido: " + ID_Pedido.ToString());
    sb.AppendLine("");
    DataSet ds = Session["Carrinho"] as DataSet;
    foreach (DataRow row in ds.Tables[0].Rows)
    {
        sb.AppendLine("Produto: " + row["Nome_Produto"].ToString());
        sb.AppendLine("Preço: " + row["Preco"].ToString());
        sb.AppendLine("");
    }
    sb.AppendLine("Total: " + ds.Tables[0].Rows[0]["Total"].ToString());
    msg.Body = sb.ToString();
    msg.BodyFormat = MailFormat.Text;
    string UserName = "lucassimoes@univem.edu.br";
    string Password = "Pass";
    string MailServer = "mail.univem.edu.br";
    msg.Fields.Add("http://schemas.microsoft.com/cdo/configuration/smtpauthenticate", 1);
    msg.Fields.Add("http://schemas.microsoft.com/cdo/configuration/sendusername", UserName);
    msg.Fields.Add("http://schemas.microsoft.com/cdo/configuration/sendpassword", Password);
    msg.Fields.Add("http://schemas.microsoft.com/cdo/configuration/sendusing", 2);
    msg.Fields.Add("http://schemas.microsoft.com/cdo/configuration/smtpserver", MailServer);
    msg.Fields.Add("http://schemas.microsoft.com/cdo/configuration/smtpconnectiontimeout", 10);
    msg.Fields.Add("http://schemas.microsoft.com/cdo/configuration/smtpserverport", 25);
    msg.Fields.Add("http://schemas.microsoft.com/cdo/configuration/smtpusessl", false);
    System.Web.Mail.SmtpMail.Send(msg);
}
```

E-Commerce – Fechamento do Pedido

- Programar clique do ImageButton1





```
protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    SendEmail();
    SendtoDataBase();

    Response.Redirect("~/CheckOut/Final.aspx");
}
```