



Cadeira de Tecnologias de Informação

Ano lectivo 2009/2010

Sites dinâmicos **Com Expression Web**

.ASPX vs .HTML

HTML:

- HTML é uma linguagem para descrever páginas web
- HTML significa *Hyper Text Markup Language*
- HTML não é uma linguagem de programação, é uma linguagem de marcação
- HTML usa *markup tags* para descrever páginas web

ASPX:

- ASP.NET é uma tecnologia de *script* que funciona do lado do servidor e que permite criar código (embebido em páginas web) para ser executado num servidor de Internet
- ASP.NET é uma tecnologia Microsoft
- ASP significa *Active Server Pages*
- ASP.NET é um programa que corre "dentro" do IIS
- IIS (*Internet Information Services*) é um servidor de internet da Microsoft

.ASPX

○ O que é um ficheiro ASP.NET?

- Um ficheiro ASP.NET é o mesmo que um ficheiro HTML
- Um ficheiro ASP.NET pode conter HTML, XML, e *scripts*
- O código de um ficheiro ASP.NET é executado no servidor
- Um ficheiro ASP.NET tem a extensão ".aspx"

○ Como funciona o ASP.NET?

- Quando um *browser* faz um pedido de um ficheiro HTML, o servidor retorna o ficheiro sem qualquer modificação
- Quando um *browser* faz um pedido de um ficheiro ASP.NET file, o IIS passa o pedido ao "motor" ASP.NET no servidor
- O "motor" ASP.NET lê o ficheiro, linha a linha, e executa o código do ficheiro, antes de enviar alguma coisa ao *browser*
- Finalmente, o ficheiro ASP.NET é retornado para o *browser* como se fosse um ficheiro puramente HTML

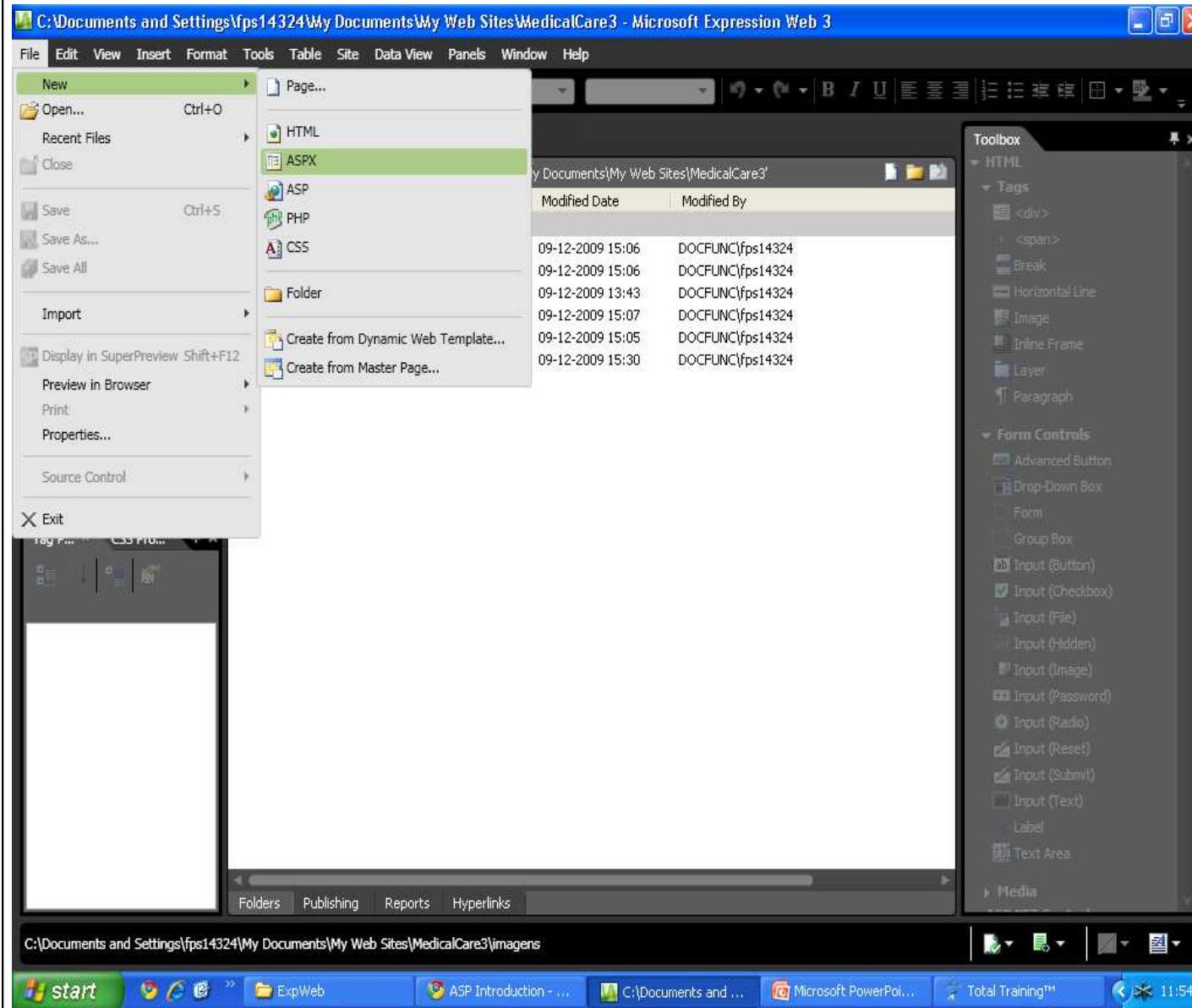
.ASPX

- ASP.NET contém um largo conjunto de controlos HTML. Praticamente todos os controlos HTML numa página podem ser definidos como objectos de controlo ASP.NET, que podem ser controlados por código
- ASP.NET também disponibiliza um novo conjunto de objectos de controlo de *input*, como *list-boxes* programáveis e controlos de validação
- O novo controlo *data grid* suporta ordenação e visualização dos dados, entre outras funcionalidades

O que o ASP.NET pode fazer por nós?

- Editar, alterar ou adicionar conteúdos numa página web de forma dinâmica
- Responder a *queries* do utilizador ou a dados submetidos em formulários HTML
- Aceder a dados ou bases de dados e devolver os resultados ao *browser*
- Personalizar uma página web de forma a torná-la mais útil aos utilizadores
- Fornece alguma segurança, uma vez que o código ASPX não pode ser visto do *browser*
- Pode diminuir o tráfico da rede

Criar uma Página .ASPX (1)



Reparem que a página apresenta uma *tag form* por defeito!

Estas páginas são muitas vezes chamadas de *Web Forms*

Atenção: Não se esqueçam de a salvar, **dentro do site** em que estão a trabalhar!

Criar uma Página .ASPX (2)



Em primeiro lugar, vamos introduzir na nova página, o formato que definimos para as páginas HTML ⁽¹⁾

Como podem verificar, para já não existem grandes diferenças entre esta e as páginas que criámos anteriormente.

⁽¹⁾ Ou repetem o mesmo processo que fizeram para as páginas HTML ou copiam:

- o código relativo aos estilos (`<style..>..</style>`) para o mesmo sítio na nova página
- e o que está dentro do *body*, para dentro das *tags* `<form..>...</form>` da nova página (apagando na vista Design o que for preciso)

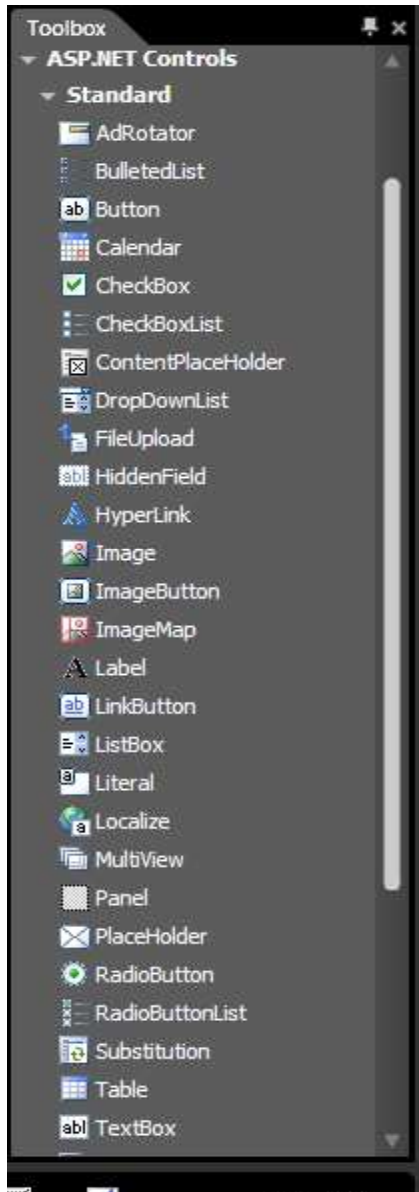
Criar uma Página .ASPX (3)



(1) Ou repetem o mesmo processo que fizeram para as páginas HTML ou copiam:

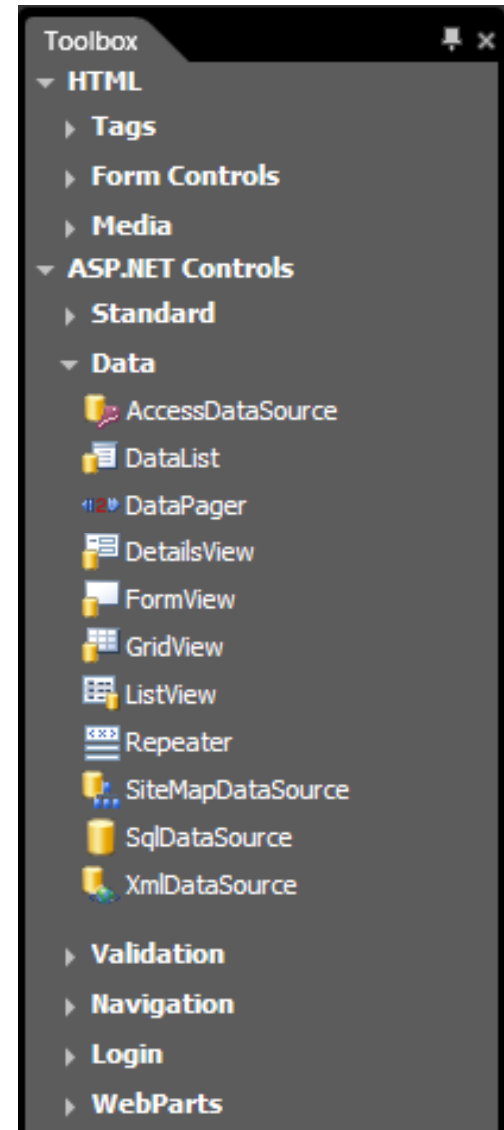
- o código relativo aos estilos (`<style..>..</style>`) para o mesmo sítio na nova página
- e o que está dentro do *body*, para o meio das *tags* `<form..>...</form>` da nova página (apagando depois na vista *Design* o que for preciso)

Objectos de Controlo .ASPX

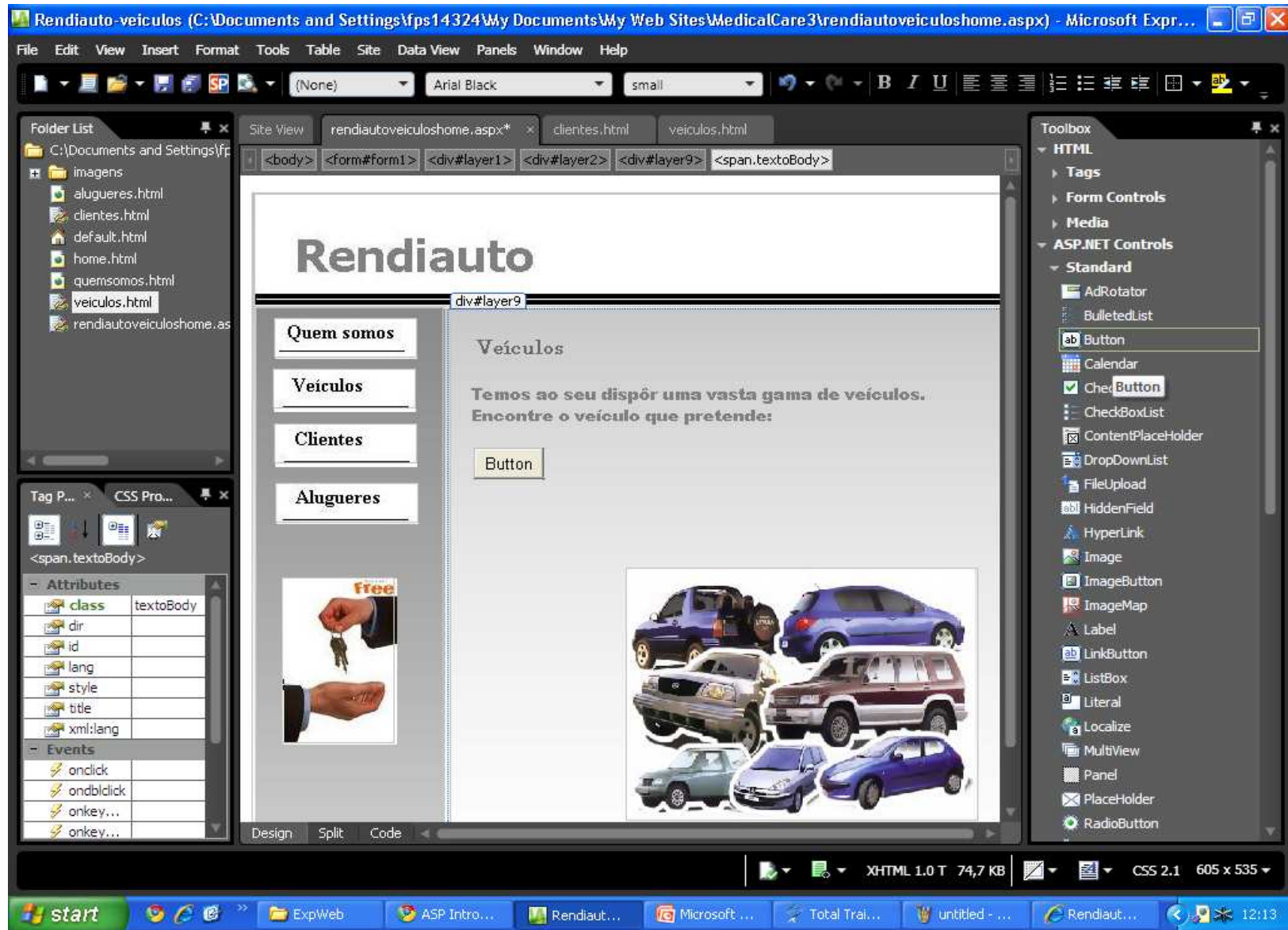


Embora possam utilizar qualquer objecto de controlo HTML, existem, na *Toolbox*, controlos específicos para páginas ASP.NET

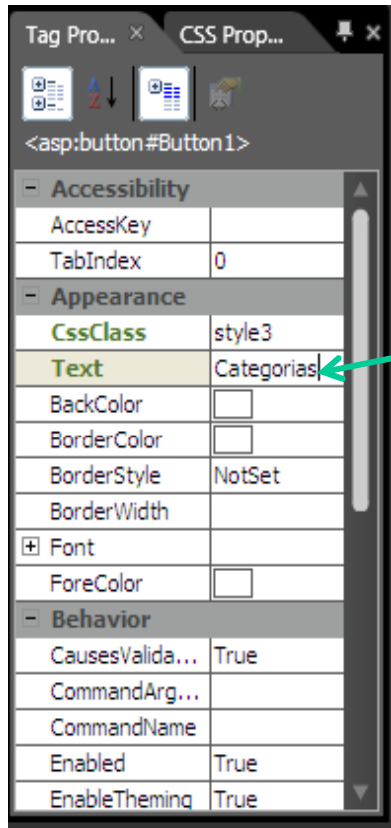
Para além destes, também estão disponíveis objectos de controlo de dados que nos permitem criar ligações a Base de Dados e visualizar, editar e inserir novos dados



Criar um botão na página .ASPX



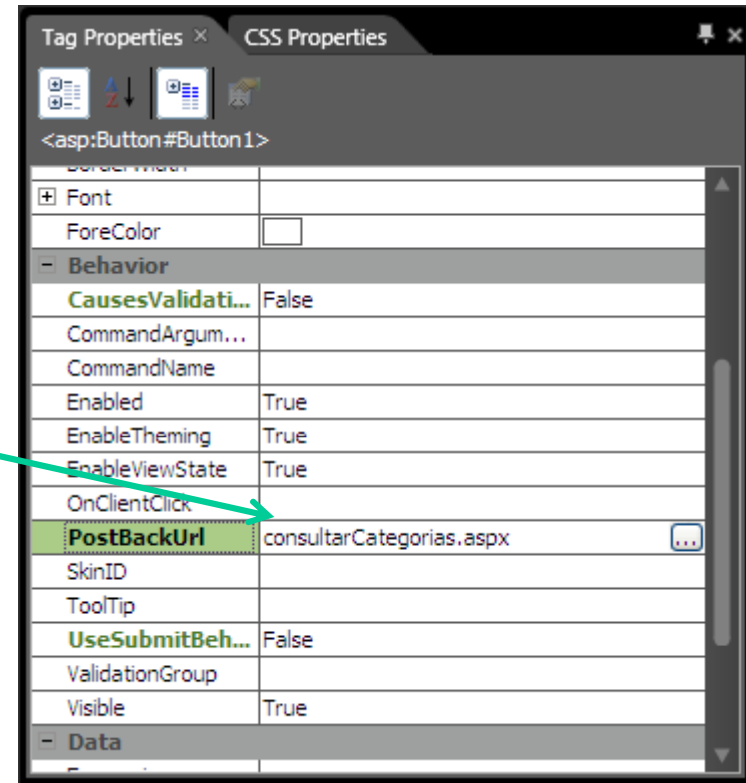
Criar um botão na página .ASPX (2)



Agora alterem algumas propriedades das CSS para este botão

Coloquem o nome na propriedade *Text*

Coloquem esta linha de código na propriedade *PostBackUrl* que vai determinar o comportamento do botão, depois de alguém carregar



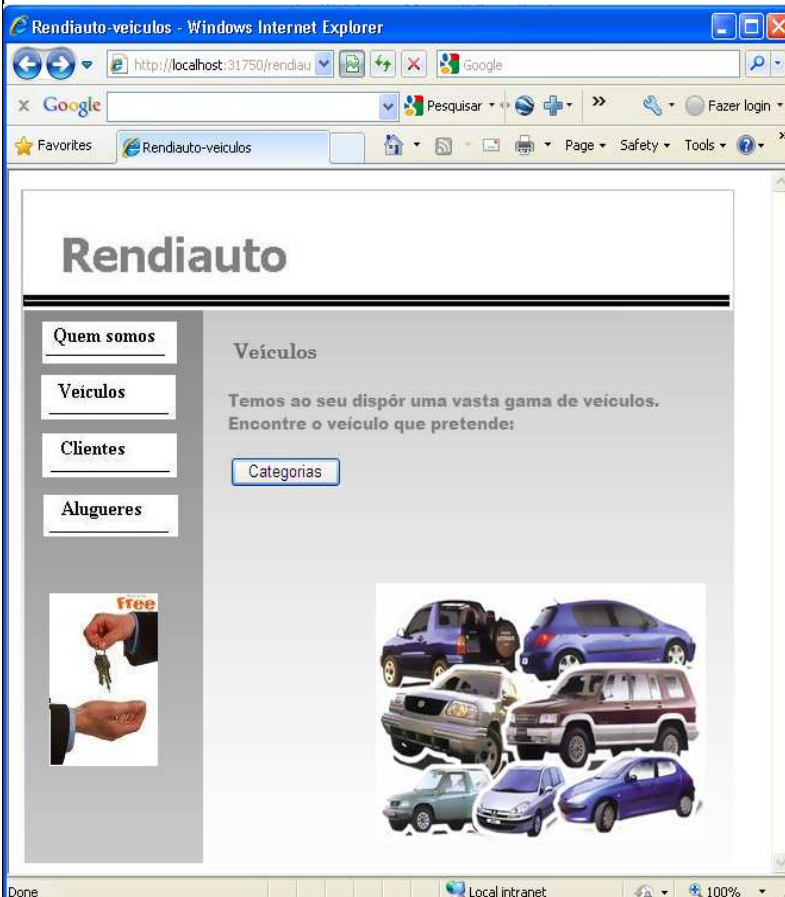
Neste caso estamos a dizer que deve ser aberta a página "consultarCategorias.aspx", como se se tratasse de um *hyperlink* (existem outras formas de fazer isto)

Ajuda: Testem no *browser* para verificar se tudo está a funcionar!

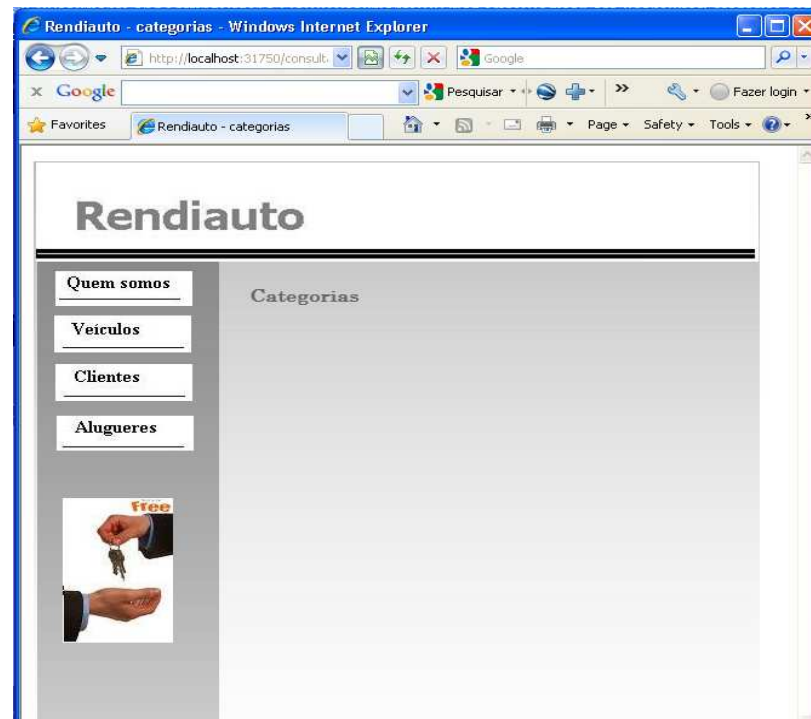
Ponto da situação

Agora vamos criar uma página que nos permita consultar as categorias existentes. Para tal, precisamos de várias coisas:

- Uma nova página com nome "consultarCategorias.aspx", com aspecto similar às anteriores, mas sem nada escrito no corpo



- Uma ligação à base de dados para que seja possível ir buscar os valores a visualizar

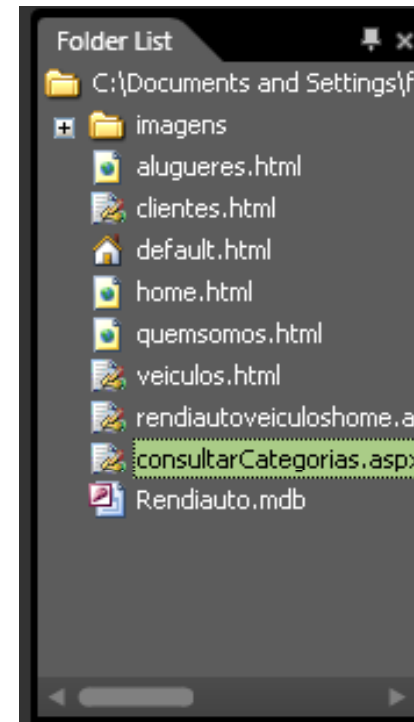
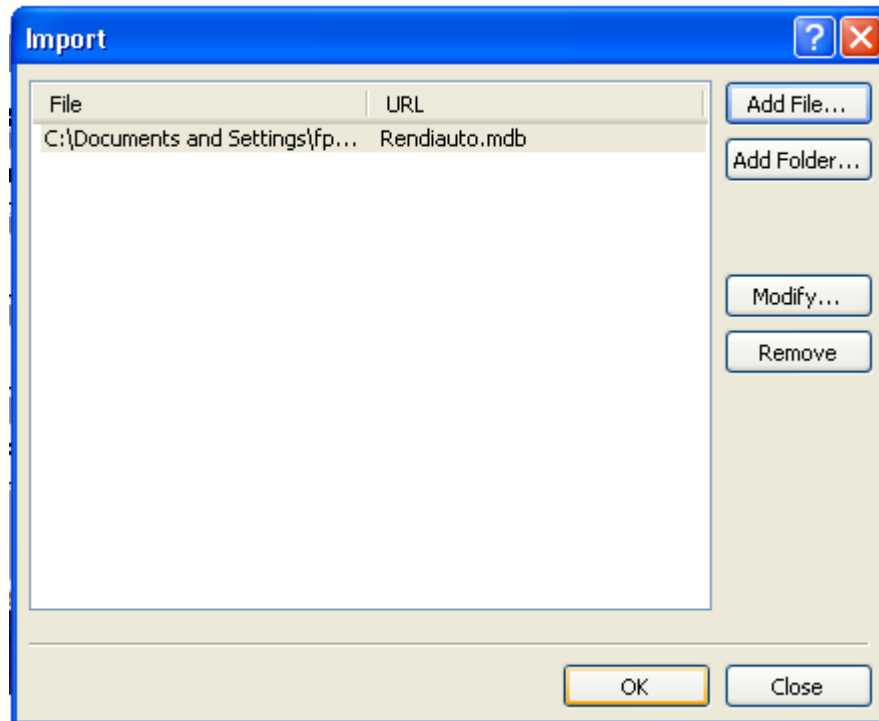


Criar uma ligação à base de dados (1)

Primeiro vamos importar a base de dados para o *site*



Criar uma ligação à base de dados (2)



ATENÇÃO: A base de dados deve estar em formato *.mdb*, ou seja, em formato Access 2003 ou 2000

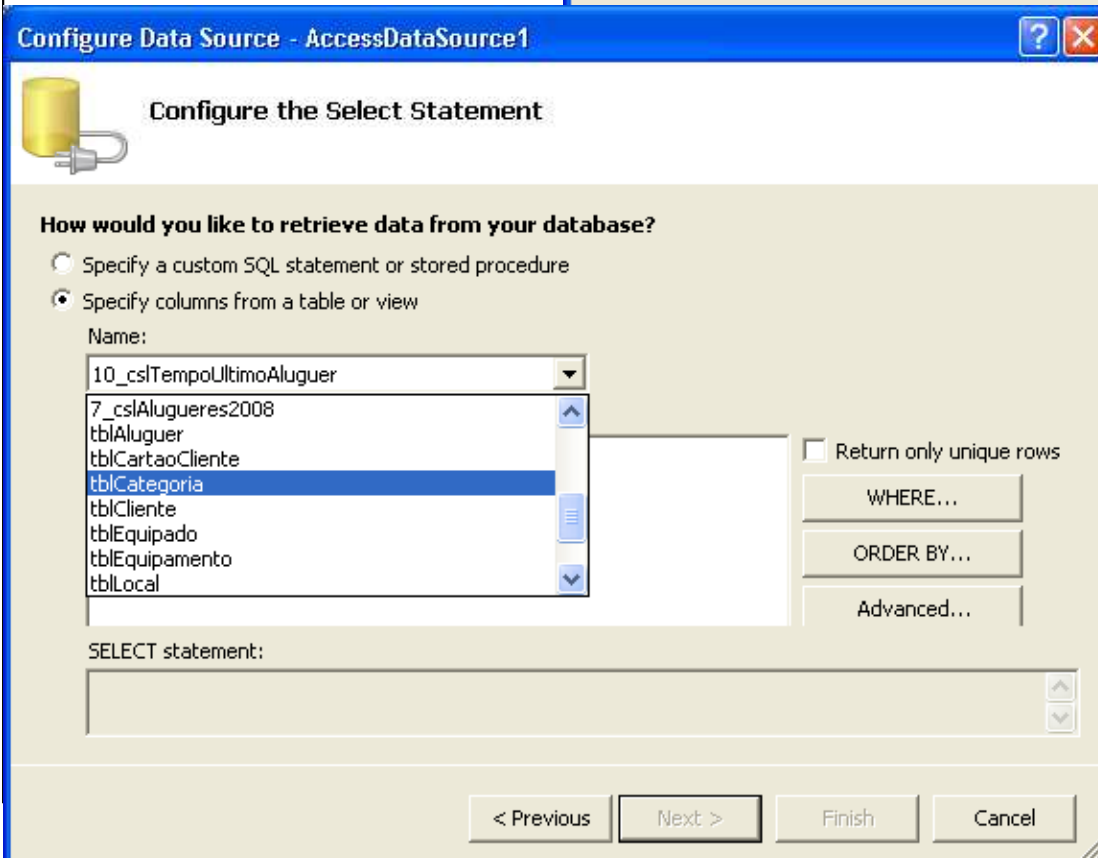
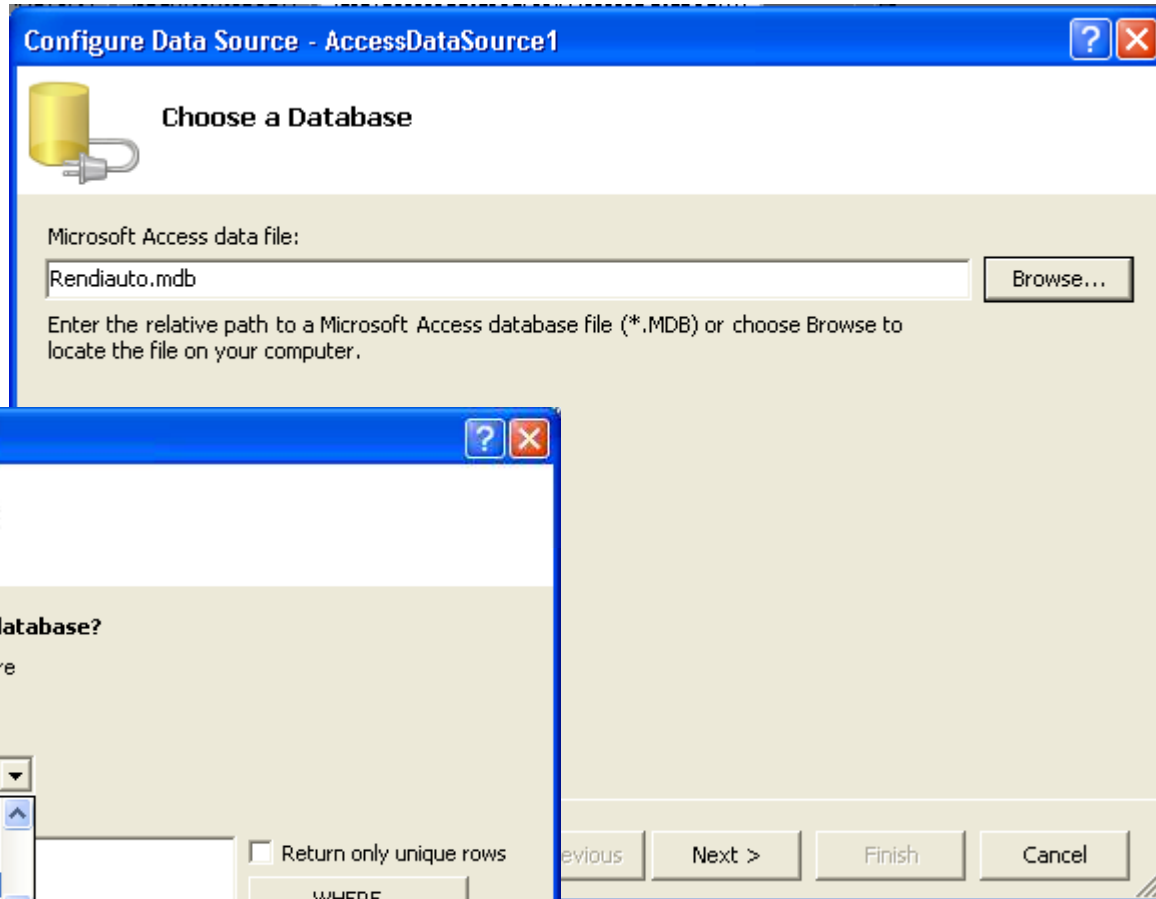
Criar uma ligação à base de dados (3)

The screenshot displays the Microsoft Expression Web 3.0 interface. The main workspace shows a web page titled "Rendiauto" with a sidebar menu containing "Quem somos", "Veiculos", "Clientes", and "Alugueres". The central content area is titled "Categorias" and contains an `AccessDataSource` control. A context menu is open over the control, showing "AccessDataSource Tasks" and "Configure Data Source...". The Toolbox on the right side of the interface has the `AccessDataSource` control highlighted with a yellow box. A tooltip is visible over the `AccessDataSource` control, stating "Connect to an Access database created with Microsoft Office." The status bar at the bottom indicates the page is rendered in XHTML 1.0 T, 27.4 KB, with CSS 2.1 and a resolution of 609 x 535.

Na *Toolbox* escolham *AccessDataSource* e depois cliquem na opção de configurar a fonte de dados (como mostra a imagem)

Criar uma ligação à base de dados (4)

Escolham a base de dados com que estão a trabalhar



Aqui poderão escolher a tabela ou a consulta que servirá de fonte dos dados

Criar uma ligação à base de dados (5)

Ajuda: Devem sempre testar, para verificar se está tudo correcto!

Configure Data Source - AccessDataSource1

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: tblCategoria

Columns:

- *
- categoriaID
- nomeCategoria
- preçoDiario

SELECT statement:
SELECT [nomeCategoria], [preçoDiario] FROM [tblC...

< Previous

Configure Data Source - AccessDataSource1

Test Query

To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.

nomeCategoria	preçoDiario
Alta	50
media	25
comerciais	40
ligeiros utilitários	20

Test Query

SELECT statement:
SELECT [nomeCategoria], [preçoDiario] FROM [tblCategoria]

< Previous Next > Finish Cancel

O Expression Web gera o código SQL correspondente à *query* que estamos a fazer. Não precisam de se preocupar com isto!

Criar uma vista de dados (1)

The screenshot shows the Microsoft Expression Web 3.0 interface. The main window displays a web page in design view. The page has a header with the text "Rendiauto" and a sidebar on the left with navigation links: "Quem somos", "Veiculos", "Clientes", and "Alugueres". The main content area is titled "Categorias" and contains a GridView control. A context menu is open over the GridView, showing "GridView Tasks" with options like "Auto Format...", "Choose Data Source...", "Edit Columns...", "Add New Column...", and "Edit Templates". The "Choose Data Source..." option is selected, and a dropdown menu is open showing "(None)", "AccessDataSource1", and "<New data source...>". The "Repeater" control is highlighted in the Toolbox on the right. A green arrow points from the "Repeater" control in the Toolbox to the "GridView Tasks" menu.

Agora só nos falta criar um controlo que nos permita ver os dados

Para tal vamos escolher uma *GridView* da *Toolbox* e depois vamos escolher a ligação que criamos anteriormente

Criar uma vista de dados (2)

Testem no *browser* para ver o resultado!

The screenshot shows a web browser window titled "Rendiauto - categorias - Windows Internet Explorer". The address bar shows "http://localhost:31750/cc". The page content includes a navigation menu on the left with items like "Quem somos", "Veículos", "Clientes", and "Alugueres". The main content area is titled "Rendiauto" and "Categorias", displaying a table with the following data:

Nome Categoria	Preço Diário (€)
Alta	50
media	25
comerciais	40
ligeiros utilitários	20

Overlaid on the table is a GridView control with a context menu open. The menu items are: "GridView Tasks", "Auto Format...", "Choose Data Source: AccessDataSource1", "Configure Data Source...", "Refresh Schema", "Edit Columns...", "Add New Column...", "Move Column Left", "Remove Column", "Enable Paging", "Enable Sorting", "Enable Selection", and "Edit Templates". A green arrow points to the "Edit Columns..." option.

Para alterar o aspecto da GridView devem editar as colunas

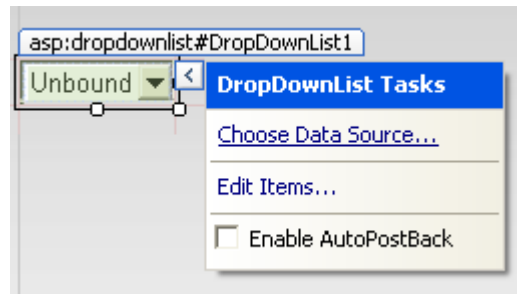
Criar outro tipo de vista de dados (1)

drop down list

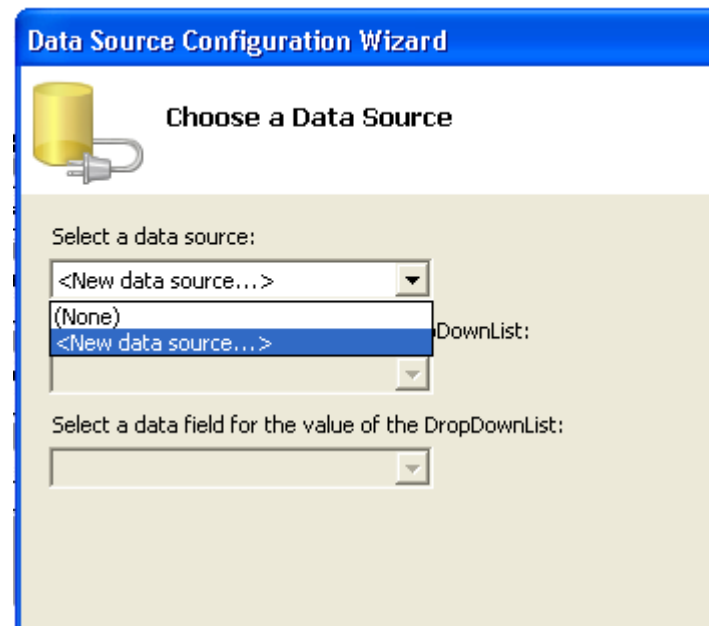


Criar outro tipo de vista de dados (2) *drop down list* – criar nova fonte de dados

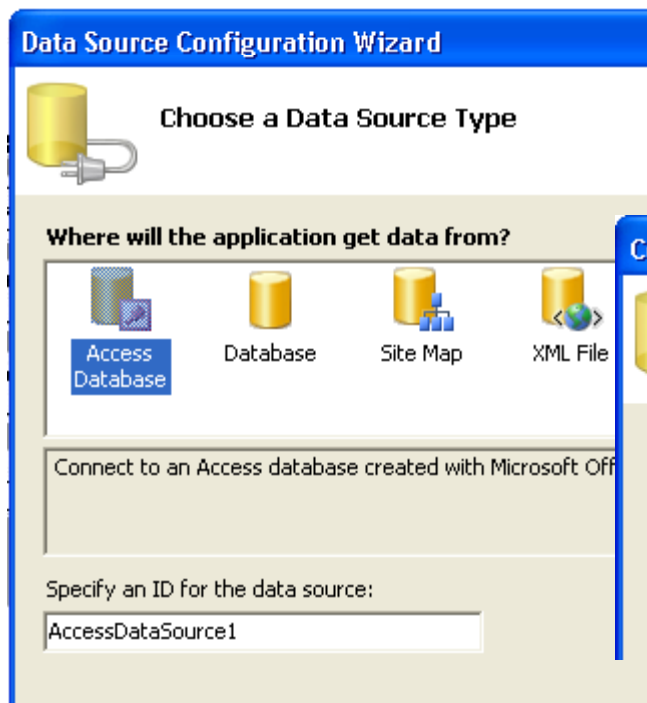
1:



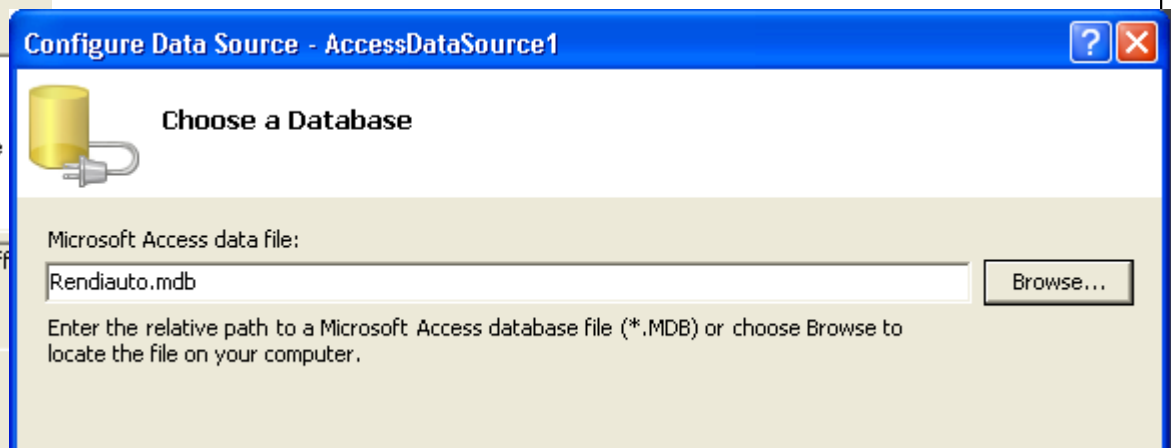
2:



3:



4:



Criar outro tipo de vista de dados (3) *drop down list* – criar nova fonte de dados

5:

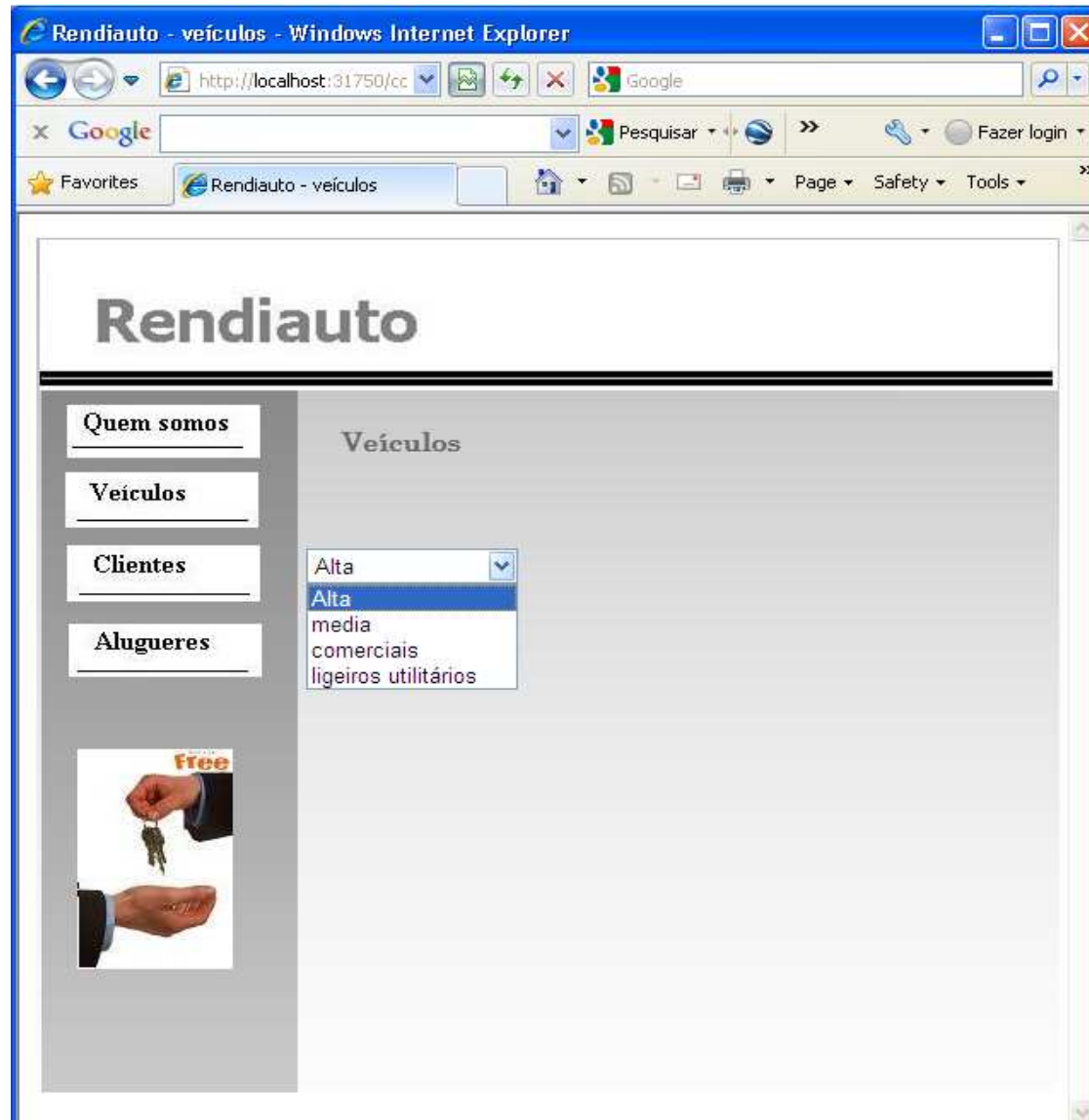
The screenshot shows the 'Configure Data Source - AccessDataSource1' dialog box. The title bar includes a help icon and a close button. The main heading is 'Configure the Select Statement'. Below this, there is a question: 'How would you like to retrieve data from your database?'. Two radio buttons are present: 'Specify a custom SQL statement or stored procedure' (unselected) and 'Specify columns from a table or view' (selected). Under the selected option, there is a 'Name:' dropdown menu with 'tblCategoria' selected. Below that is a 'Columns:' list with checkboxes: '*' (unchecked), 'categoriaID' (checked), 'nomeCategoria' (checked), and 'preçoDiario' (unchecked). To the right of the columns list is a checkbox for 'Return only unique rows' (unchecked) and three buttons: 'WHERE...', 'ORDER BY...', and 'Advanced...'. At the bottom, there is a 'SELECT statement:' text box containing the SQL query: 'SELECT [categoriaID], [nomeCategoria] FROM [tblCategoria]'. At the very bottom are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

6:

The screenshot shows the 'Data Source Configuration Wizard' dialog box. The title bar includes a help icon and a close button. The main heading is 'Choose a Data Source'. Below this, there is a question: 'Select a data source:'. A dropdown menu shows 'AccessDataSource1' selected. Below that is another question: 'Select a data field to display in the DropDownList:'. A dropdown menu shows 'nomeCategoria' selected. Below that is a third question: 'Select a data field for the value of the DropDownList:'. A dropdown menu shows 'categoriaID' selected.

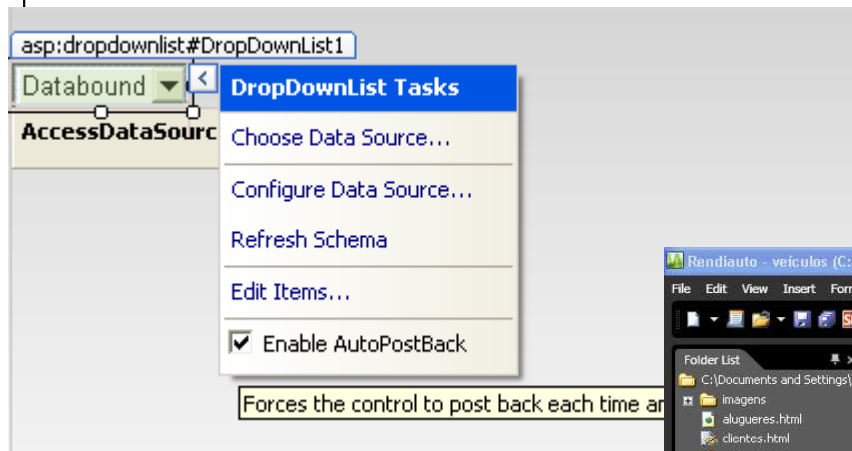
7: Testar no *browser* 😊

Criar outro tipo de vista de dados (4) *drop down list*



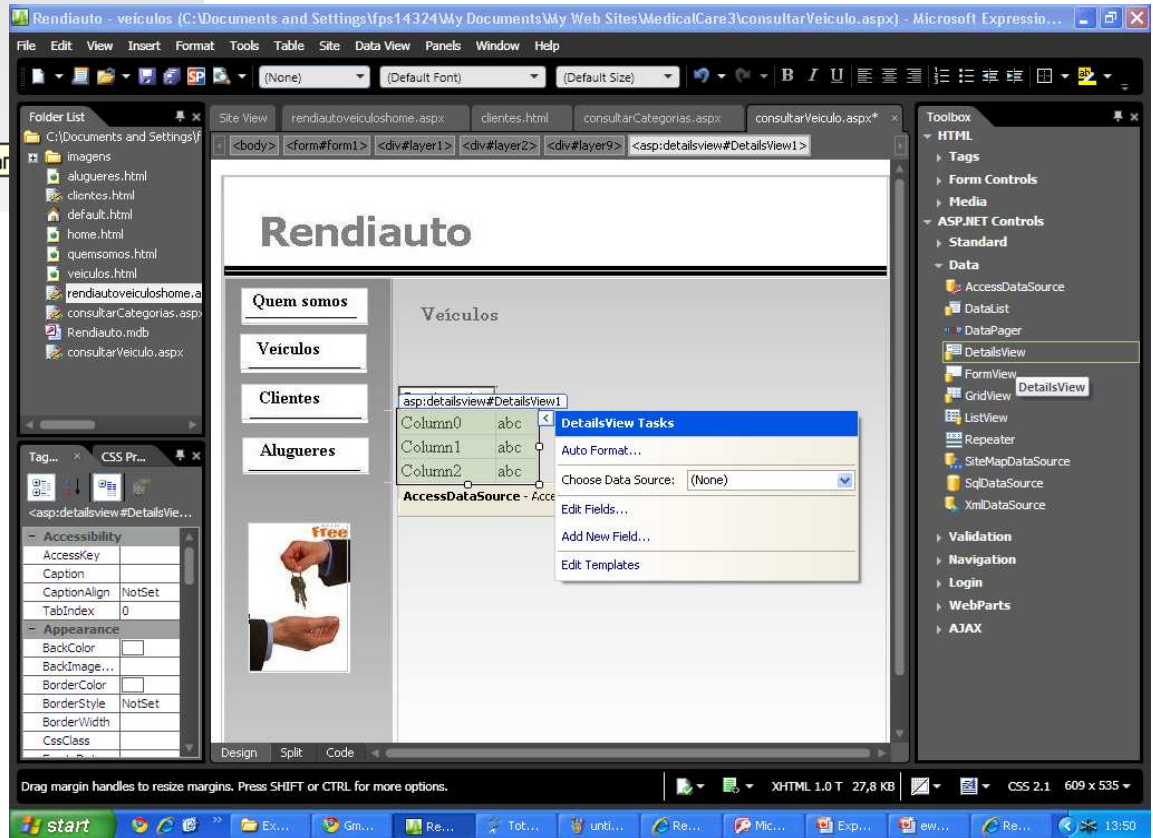
Agora devemos juntar a categoria seleccionada para mostrar todos os veículos respectivos

Ligar um parâmetro seleccionado a uma consulta (1)



Primeiro devem etiquetar a opção
Enable AutoPostBack

Depois escolham o controlo
DetailView



Ligar um parâmetro seleccionado a uma consulta (2)

Configure Data Source - AccessDataSource2

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:
tblVeiculo

Columns:

<input type="checkbox"/> *	<input type="checkbox"/> codCategoria
<input checked="" type="checkbox"/> matrícula	
<input checked="" type="checkbox"/> cilindrada	
<input checked="" type="checkbox"/> marca	
<input checked="" type="checkbox"/> modelo	
<input checked="" type="checkbox"/> ano	

Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:
SELECT [matrícula], [cilindrada], [marca], [modelo], [ano] FROM [tblVeiculo]

< Previous Next > Finish Cancel

Voltem a criar uma fonte de dados, especificando a base de dados, desta vez para a tabela veículo

Para ligar ao parâmetro seleccionado, carreguem na opção *Where*

Ligar um parâmetro seleccionado a uma consulta (3)

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:

Operator:

Source:

Parameter properties

Control ID:

Default value:

SQL Expression:

Value:

Add

WHERE clause:

SQL Expression	Value
----------------	-------

Remove

OK Cancel

Preencham os valores, como na figura e depois cliquem em *Add -> Ok*

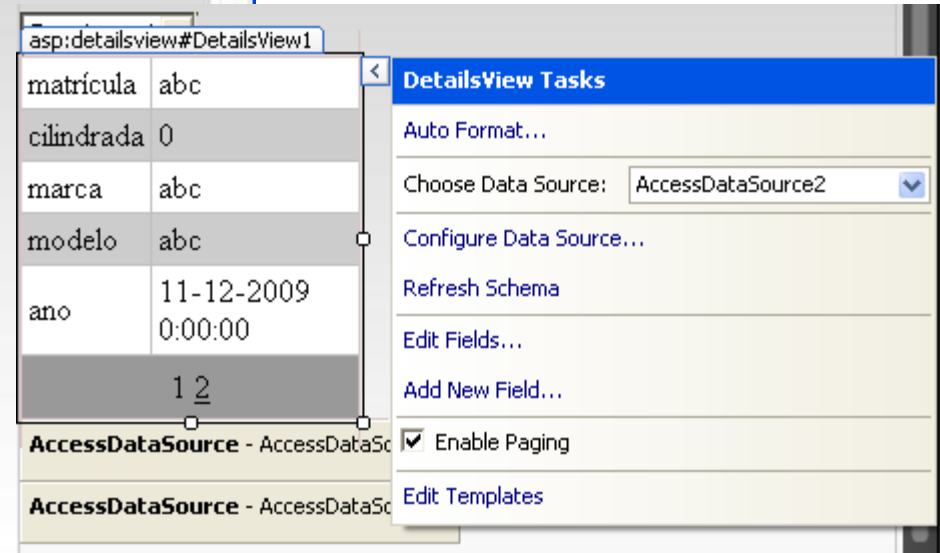
Quando terminarem, visualizem no *browser* o resultado!

Testar no *Browser* (1)



Reparem que para cada categoria apenas é apresentado um veículo

Para resolver este problema têm de etiquetar a opção *Enable Paging*



Testar no *Browser* (2) – *Voilà!*



The screenshot shows a Windows Internet Explorer browser window displaying a web application titled "Rendiauto - veículos". The address bar shows the URL "http://localhost:31750/cc". The page content includes a navigation menu on the left with links for "Quem somos", "Veículos", "Clientes", and "Alugueres". The main content area is titled "Veículos" and features a dropdown menu labeled "Escolha a categoria:" with "Alta" selected. Below this, a table displays the details of a selected vehicle:

Veículos correspondentes:	
matricula	28-FR-87
cilindrada	1500
marca	Mercedes
modelo	serie8
ano	01-08-2008
	0:00:00
	1 2

The browser's status bar at the bottom indicates "Done" and "Local intranet".

Ponto da Situação

- Com base no que aprenderam até aqui vão:
 - Alterar o que fizeram na aula passada para ligar às páginas dinâmicas
 - Criar páginas dinâmicas para os módulos que faltam
 - Criar componentes de consulta (simples e complexos) para os vários módulos de acordo com a base de dados
- Na próxima aula vamos ver:
 - Inserções
 - *Updates*
 - *Deletes*