

DataGrid

1. Introdução

O DataGrid é um dos componentes mais usados em aplicações Web ASP.Net, na medida em que permite ver (e porventura alterar) dados na forma de linhas e colunas. Como muitos outros componentes no .Net, existem muitas outras alternativas (gratuitas ou não) de empresas/pessoas que não a Microsoft.

	<u>Company</u>	<u>Name</u>	<u>Contact</u>	<u>Title</u>	<u>Region</u>	<u>Postal</u>	<u>Code</u>	<u>XXXX</u>
<u>Select</u>	<u>Alfreds</u> <u>Futterkiste</u>	xxx Alfreds Futterkiste	Sales Representative		12209	<u>Delete</u>	<u>Edit</u>	
<u>Select</u>	<u>Ana Trujillo</u> <u>Emparedados</u> <u>y helados</u>	xxx Ana Trujillo Emparedados y helados	Owner		05021	<u>Delete</u>	<u>Edit</u>	
<u>Select</u>	<u>Antonio</u> <u>Moreno</u> <u>Taqueria</u>	xxx Antonio Moreno Taqueria	Owner		05023	<u>Delete</u>	<u>Edit</u>	
<u>Select</u>	<u>Around the</u> <u>Horn</u>	xxx Around the Horn	Sales Representative		WA1 1DP	<u>Delete</u>	<u>Edit</u>	
<u>Select</u>	<u>Berglunds</u> <u>snabbköp</u>	xxx Berglunds snabbköp	Order Administrator		S-958 22	<u>Delete</u>	<u>Edit</u>	
1 2 3 4 5 6 7 8 9 10 ...								

Figura 1: Exemplo de uma DataGrid

2. Fonte de dados

Antes de se usar as grelhas, temos que as ligar aos dados. Para fazer essa ligação faz-se o seguinte:

- Nas propriedades em DesignTime
 - Preencher a propriedade DataSource da DataGrid com o DataSet ou o DataTable que vai ter os dados.
- No código (a executar em run-time)
 - Preencher com dados o DataSet (ou DataTable)

2. Fazer `DataGrid.DataBind();` ou `DataBind();`. Este método faz a ligação entre a DataGrid (componente visual) e os dados especificados na propriedade DataSource. Se for executada a primeira opção, só é feita a ligação para a grelha; na 2ª opção é feita a ligação para todos os componentes da página

Exercício 1

Para o primeiro exemplo, vamos criar uma página que mostra numa grelha, a tabela Customers da base de dados NorthWind.

Siga os passos:

1. Crie um novo WebForm (nome = ExemploDataGrid.aspx)
2. Abra o Server Explorer (Ctrl+Alt+S).
3. No nodo Data Connections, faça: botão direito -> Add Connections ...
4. Preencha as propriedades de ligação à base de dados NorthWind
5. Procure e seleccione o nó relativo à tabela “Customers”.
6. Arraste-o para a página.
7. Neste ponto deverá ter 2 componentes: SqlConnection1 e SqlDataAdapter1 na página.

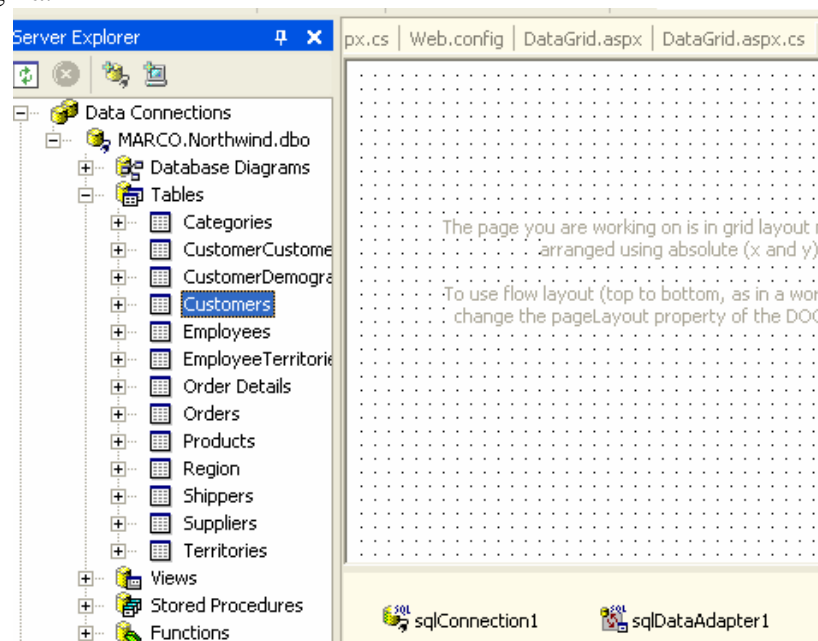
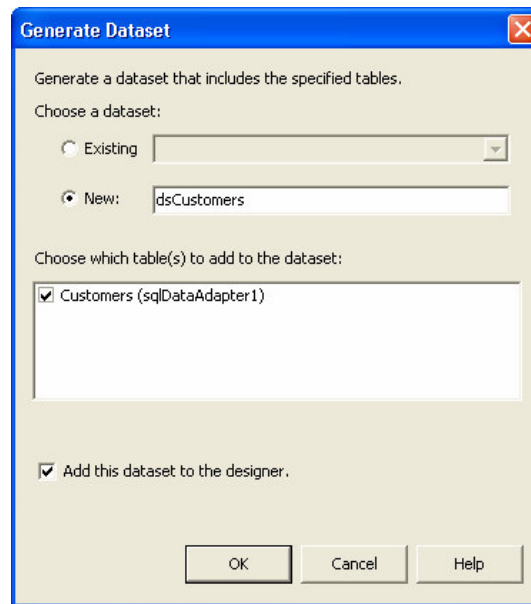
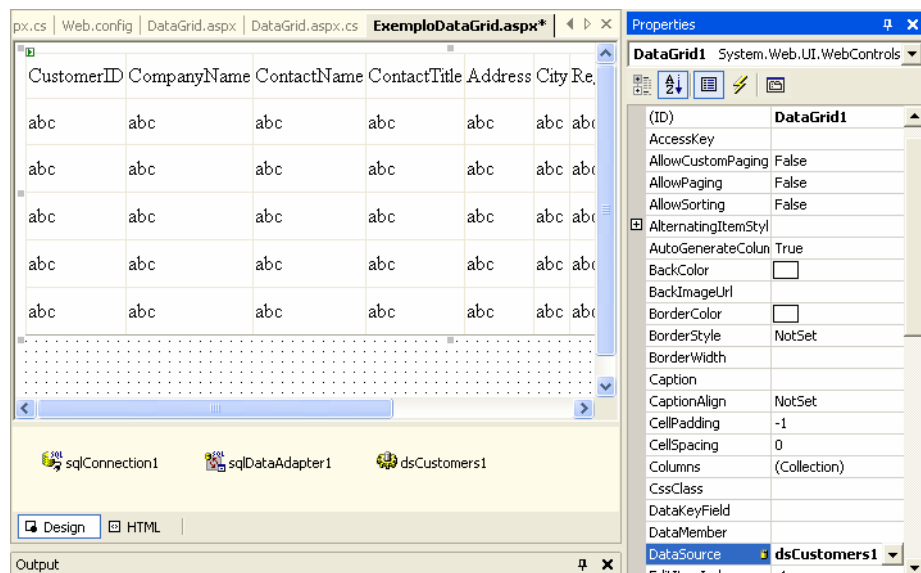


Figura 2: Server Explorer

8. Sobre o componente SqlDataAdapter1, use o botão direito e escolha a opção Generate Dataset.

**Figura 3: Generate Dataset**

9. Na janela de Generate DataSet, escolha a opção “New:”, coloque o nome = dsCustomers
10. Certifique-se que activa a opção “Add this dataset to the designer”.
11. Nesta fase, para além dos 2 componentes anteriores, existe mais um componente com o nome: “dsCustomers1”
12. Coloque uma DataGrid na página
13. Altere a propriedade DataSource da DataGrid. DataSource = dsCustomers1

**Figura 4: DataGrid com DataSource preenchida**

14. Faça duplo-clique na página para criar o evento Page_Load.
15. Coloque o seguinte código:

```
private void Page_Load(object sender, System.EventArgs e)
{
    sqlDataAdapter1.Fill(dsCustomers1);
    DataGrid1.DataBind();
}
```

16. O método Fill preenche o DataSet com os dados da base de dados
17. O método DataBind() faz a ligação da grelha com os dados no DataSet.
18. Pode correr a aplicação.

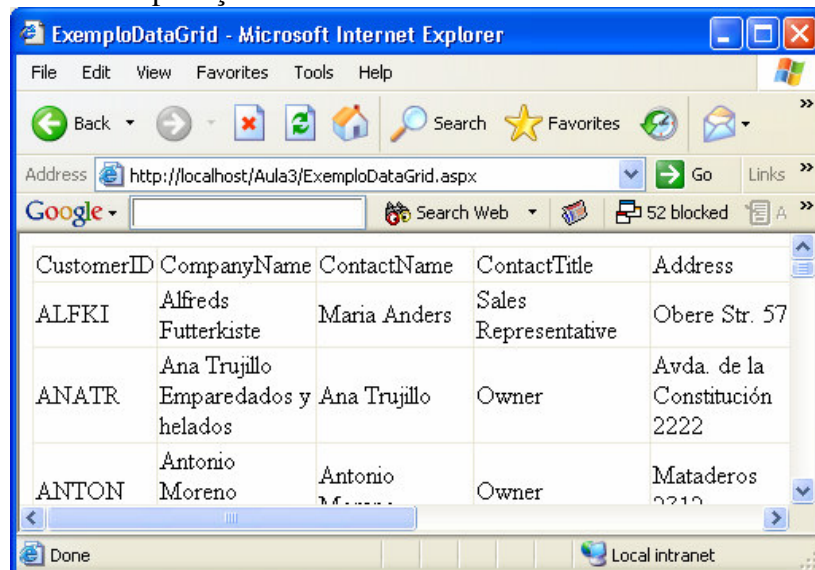


Figura 5: Exemplo de uma DataGrid

Apesar dos 19 pontos do exemplo anterior, se compararmos com o código necessário em ASP para gerar uma página similar verificamos que há uma grande diferença.

3. AutoFormat

A opção AutoFormat serve como wizard para alterar o aspecto da grelha em poucos segundos:

Exercício 2

Faça agora a seguinte experiência:

1. Sobre o componente DataGrid, pressione no botão direito do rato e escolha o menu: “Auto Format”

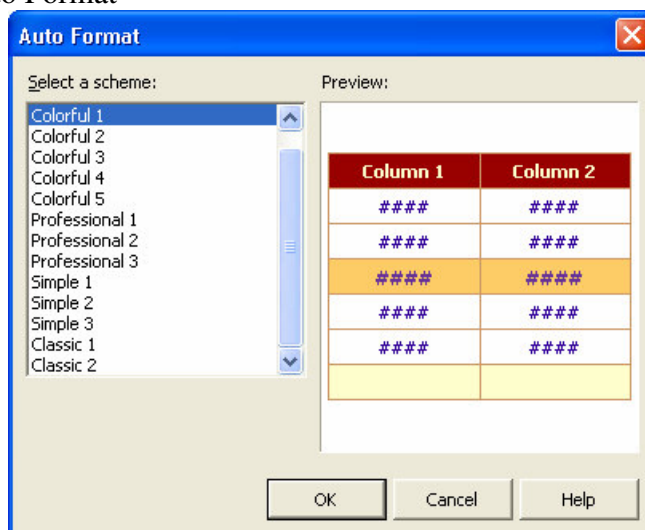


Figura 6: Escolha um dos formatos (exemplo: Coloful 1)

2. Correr o programa:

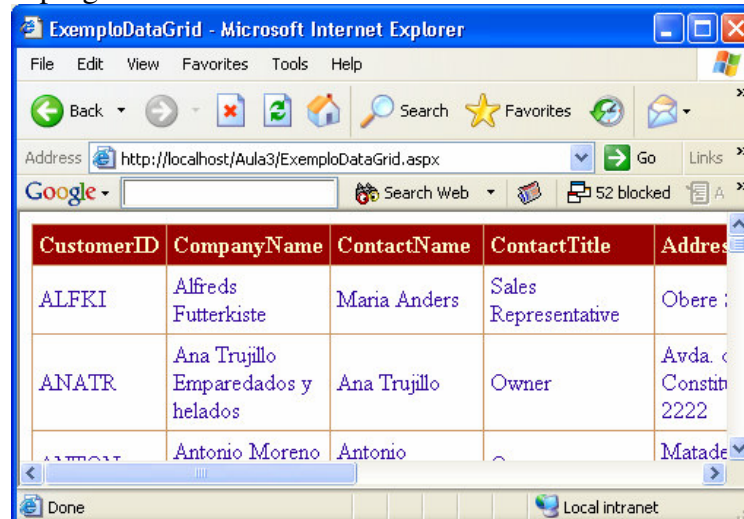


Figura 7: Exemplo com AutoFormat

4. Property Builder

Para manipular as propriedades da grelha pode usar a janela de propriedades ou o Property Builder.

1. Selecciona a DataGrid. Botão Direito -> menu: Property Builder

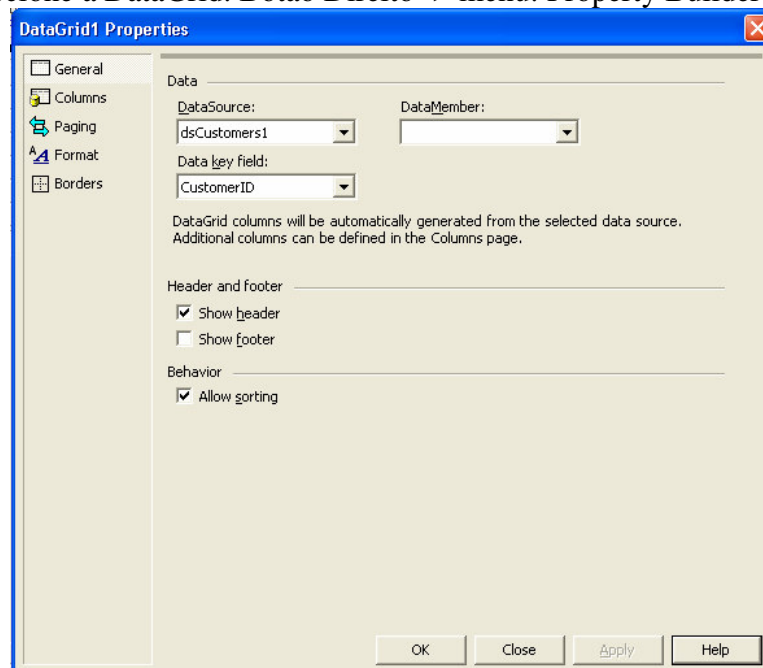


Figura 8: Property Builder / General

Na página “General”, pode preencher o DataSource e o DataMember para especificar a fonte de dados. Note que o DataMember permite especificar qual a tabela a usar, porque os datasets podem ter mais do que uma tabela (no exemplo, o DataSet só tem uma tabela, logo não é necessário especificar DataMember).

O campo DataKey permite indicar qual o campo que é chave primária. Este campo é especialmente útil para situações em que a chave primária das linhas não pertence à lista de colunas da grelha.

O Show Header indica se os cabeçalhos são ou não visíveis. O ShowFooter permite colocar uma linha no fim (como footer da grelha).

O campo Allow Sorting, activa o suporte à ordenação na grelha:

- Mostra os títulos das colunas como links
- Lança o evento sortcommand quando o utilizador clica nos links referidos no ponto anterior.

Exercício 3

Coloque as propriedades:

- **DataSource** = dsCustomers1
- **DataMember** =
- **Data Key Field** = CustomerID
- **Show Header** = True
- **Show Footer** = False
- **Allow Sorting** = True

Se correr a aplicação, a grelha vai ter os títulos como links (para ordenar), mas se carregarmos nos links, não faz nada.

5. Ordenação

A propriedade Allow Sorting apenas prepara a grelha para ordenação. Para realmente ordenarmos a grelha, temos que preencher o evento SortCommand.

SortCommand

O evento SortCommand ocorre sempre que o utilizador clicar no link do título quando pretende ordenar a grelha.

É passado para este evento o parâmetro “e”, com a propriedade SortExpression que indica qual o campo escolhido pelo utilizador para ordenar a grelha. Normalmente o SortExpression tem o nome do campo, mas pode ter outro valor qualquer definido pelo programador.

Exercício 4

Siga os passos seguintes para ordenar a grelha:

1. Certifique-se que **Allow Sorting** = True
2. Coloque o seguinte código no evento SortCommand da DataGrid:

```
{
    sqlDataAdapter1.SelectCommand.CommandText =
        "SELECT CustomerID, CompanyName, ContactName, "+
        " ContactTitle, Address, City, Region, PostalCode, "+
        " Country, Phone, Fax FROM Customers"+
        " ORDER BY " + e.SortExpression;
    dsCustomers1.Clear();
    sqlDataAdapter1.Fill(dsCustomers1);
    DataGrid1.DataBind();
}
```

3. Corra a aplicação. Verifique que agora quando se clica no título, a grelha é reordenada.

Vamos analisar rapidamente o código anterior. O comando select é alterado para ordenar pelo campo que é passado no parâmetro e (e.SortExpression). Os dados do Dataset são limpos (dsCustomers1.Clear()) e novamente preenchidos a partir da base de dados (sqlDataAdapter1.Fill(dsCustomers)) mas desta vez já com os registos correctamente ordenados.

6. Paginação

O funcionamento da paginação é similar à ordenação. Existem propriedades que preparam a grelha para a paginação e um evento que realmente faz a paginação (**PageIndexChanged**).

Vamos fazer um exemplo:

Exercício 5

Siga os passos seguintes:

1. Selecione a grelha e escolha o menu Property Builder.
2. Na página “General”, certifique-se que **Allow Sorting** = False. Nesta fase, vamos apenas paginar a grelha, esquecendo por momentos a ordenação.
3. Escolha a página “Paging”.

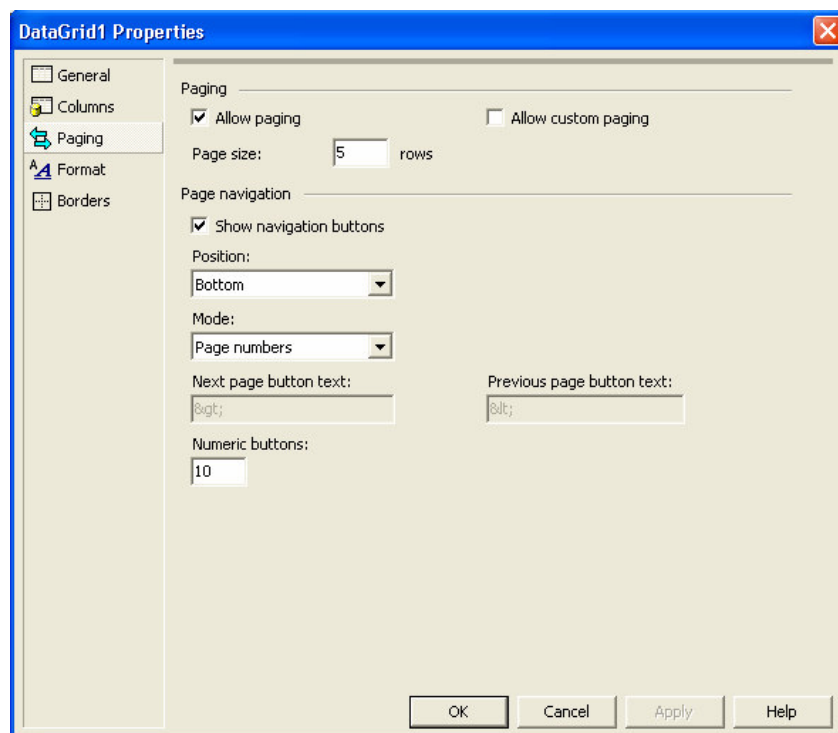


Figura 9: Propriedade de paginação

4. Nesta página, active a propriedade Allow paging
5. Page Size = 5 rows
6. Mode = Page numbers.
7. Corra a aplicação:

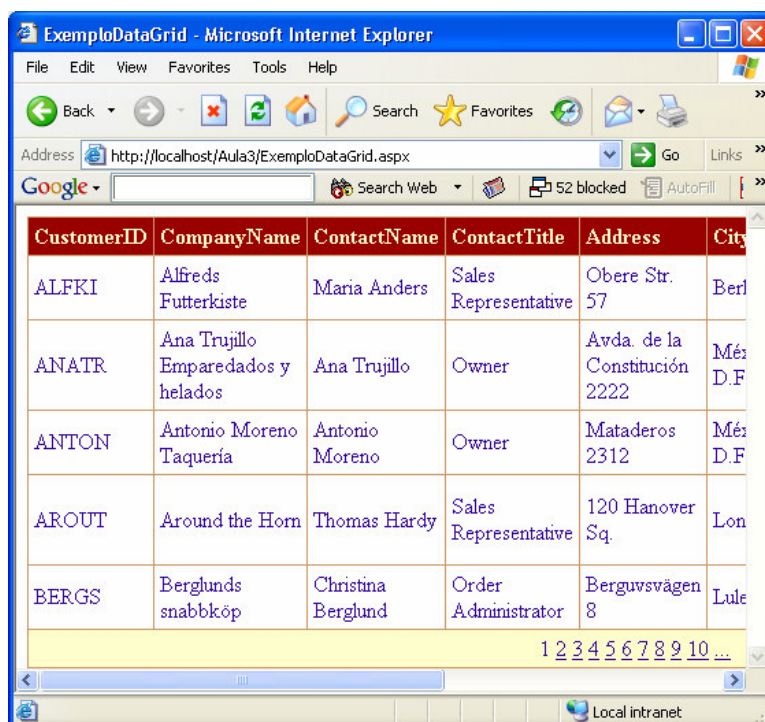


Figura 10: Exemplo com paginação

Se correremos a aplicação, a grelha já aparece paginada (com 5 linhas) e com os links para as diferentes páginas 1..10. No entanto se correremos e clicarmos na diferentes páginas não acontece nada.

Antes de resolvermos esse problema, vamos fazer uma análise rápida das propriedades da paginação:

- Allow paging – activa/desactiva paginação
- Page size – nº de linhas da paginação
- Show navigation buttons – Indica se os botões de navegação aparecem na grelha. Se desactivar esta opção, terá que ser o programador a criar esses botões.
- Position – Indica qual a posição dos botões de navegação (no topo, em baixo ou em ambos os locais).
- Mode - Indica se os botões aparecem na forma de números das páginas, ou se aparece apenas o botão anterior e próximo.
- Next (e previous) page button text – Texto usado para o próximo e anterior (em html).
- Numeric Buttons – Total de botões quando aparecem os números das páginas.

Para colocar a paginação a funcionar é necessário preencher o evento **PageIndexChanged**. Este evento é activado quando o utilizador altera a página ao carregar num dos botões de paginação. Para esse evento é passado o parâmetro “e”, com a propriedade **NewPageIndex** que indica qual página escolhida pelo utilizador.

Exercício 6

1. Escolha a DataGrid e crie o evento PageIndexChanged.

2. Coloque este código no evento:

```
{  
    DataGrid1.CurrentPageIndex = e.NewPageIndex;  
    DataGrid1.DataBind();  
}
```

3. Corra a aplicação.

O que o código faz é alterar a página actual da grelha (DataGrid.CurrentPageIndex) de acordo com a página escolhida pelo utilizador (e.NewPageIndex)

ATENÇÃO

No exemplo anterior, todos os dados são carregados no DataSet e só depois é que se altera a página a mostrar. Se a consulta devolvesse um milhão de registos, primeiro eram carregadas um milhão de linhas para o DataSet e só depois é que se seleccionava a página pretendida

Para que a paginação seja mais eficiente, em vez de se mudar a página actual da DataGrid (CurrentPageIndex) era necessário refazer a consulta SQL de forma a devolver apenas os dados pretendidos para a página (algo similar ao Sort Command feito anteriormente, mas com um comando Select bastante mais complexo).

7. Ordenação e Paginação

Então temos a paginação e a ordenação a funcionar, mas ainda não funcionam os dois em simultâneo.

Exercício 7

1. Volte a activar a propriedade Allow Sorting
2. Corra novamente a aplicação e analise-a.

Enquanto trabalhar apenas com a paginação a grelha funciona correctamente. No entanto, se ordenar a grelha por um dos campos, fica sempre na primeira página. Se voltar a escolher outra página, a ordenação anterior perde-se.

Ou seja, a paginação e ordenação não funcionam em simultâneo impossibilitando por exemplo mostrar a 3ª página quando a ordenação é feita pelo campo CompanyName. Para que essas características funcionem em simultâneo é necessário, guardar a ordenação escolhida entre chamadas (através das variáveis de sessão).

Exercício 8

1. Altere o evento Sort Command de forma a guardar o campo ordenado na sessão:

```
{
    sqlDataAdapter1.SelectCommand.CommandText =
        "SELECT CustomerID, CompanyName, ContactName, "+
        " ContactTitle, Address, City, Region, PostalCode, "+
        " Country, Phone, Fax FROM Customers"+
        " ORDER BY " + e.SortExpression;
    dsCustomers1.Clear();
    sqlDataAdapter1.Fill(dsCustomers1);
    DataGrid1.DataBind();
    Session["ExpressaoSort"] = e.SortExpression;
}
```

2. Altere o evento PageIndexChanged de forma a utilizar também a ordenação. Note que não é necessário guardar o valor da página actual seleccionada porque esse valor é guardado no ViewState (na propriedade CurrentPageIndex).

```
if (Session["ExpressaoSort"] != null)
{
    sqlDataAdapter1.SelectCommand.CommandText =
        "SELECT CustomerID, CompanyName, ContactName, "+
        " ContactTitle, Address, City, Region, PostalCode, "+
        " Country, Phone, Fax FROM Customers"+
        " ORDER BY " + Session["ExpressaoSort"].ToString();
    dsCustomers1.Clear();
    sqlDataAdapter1.Fill(dsCustomers1);
}
else
{
    sqlDataAdapter1.SelectCommand.CommandText =
        "SELECT CustomerID, CompanyName, ContactName, "+
        " ContactTitle, Address, City, Region, PostalCode, "+
        " Country, Phone, Fax FROM Customers";
    dsCustomers1.Clear();
    sqlDataAdapter1.Fill(dsCustomers1);
}
DataGrid1.CurrentPageIndex = e.NewPageIndex;
DataGrid1.DataBind();
```

3. Corra novamente a aplicação e analise-a.

Agora a aplicação já faz a paginação e ordenação em simultâneo. No entanto, apesar de funcionar correctamente, tem um problema de performance. Se repararmos no código, o Dataset é sempre preenchido 2 vezes:

- No evento load da página
- Num dos eventos da grelha (SortCommand ou ChangePageIndex)

Para resolvermos esse problema, basta garantir que no evento Load os dados só são carregados na primeira chamada da página.

Exercício 9

Altere o evento Page_Load para:

```
if (!IsPostBack)
{
    sqlDataAdapter1.Fill(dsCustomers1);
    DataGrid1.DataBind();
}
```

8. Format

No Property Builder da DataGrid, temos acesso à página Format, onde temos acesso a propriedades que permitem definir o aspecto visual da grelha. Parte dessas propriedades são preenchidas quando se usam o AutoFormat.

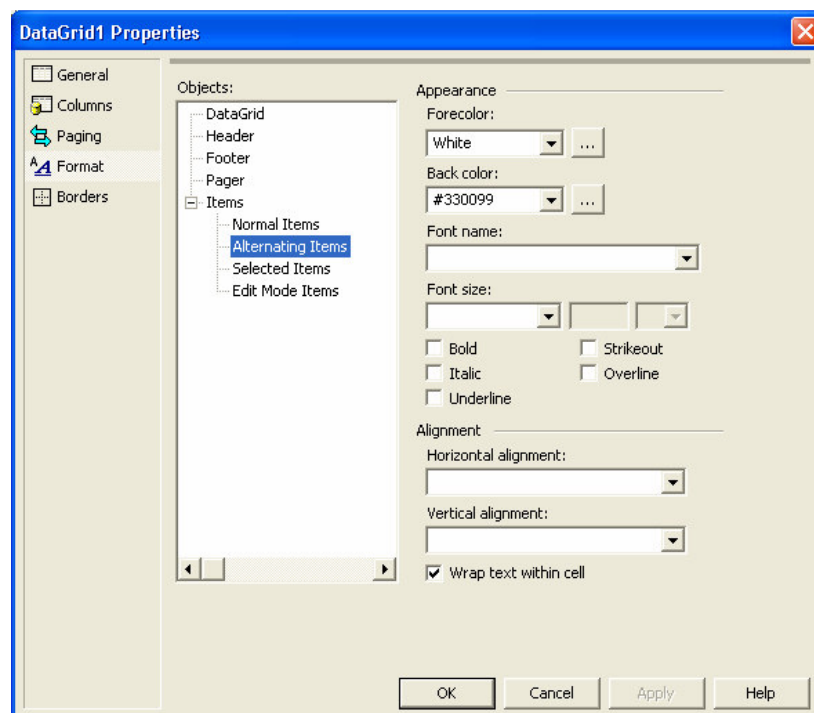


Figura 11: Format

Estas propriedades são relativas a:

- DataGrid – Propriedades gerais da grelha. Estes valores são sobrepostos quando são também preenchidos nos restantes elementos
- Header – Relativo aos cabeçalhos
- Footer – Relativo ao rodapé
- Pager – Relativo à zona de paginação
- Normal Items – Relativo às linhas “normais”
- Alternating Items – Se for preenchido, permite definir grelhas em que o aspecto das linhas é alterado. Linhas pares e impares têm formatações diferentes. O Alternating Item define as linhas pares e o Normal Items define as linhas impares.
- Selected Item – Relativo à linha seleccionada
- Edit Mode Items – Relativo às linhas que estão em modo de edição.

Exercício 10

1. Na página Format, selecione o elemento DataGrid
2. Altere a propriedade BackColor para AliceBlue
3. Selecione o elemento Normal Items
4. Coloque a propriedade BackColor a vazio. Quando a propriedade está vazia usa o valor definido na DataGrid.
5. Selecione o elemento Alternating Items
6. Coloque a propriedade ForeColor = White e BackColor = #330099
7. Corra a aplicação

CustomerID	CompanyName	ContactName	ContactTitle
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner
AROUT	Around the Horn	Thomas Hardy	Sales Representative
BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator
1 2 3			

Figura 12: Exemplo após a formatação

Cada uma das colunas pode também ter um formato específico. No entanto, essa funcionalidade só está disponível quando o programador criar colunas manualmente.

9. Borders

Para além da formatação anterior existe também a página referente às linhas (borders) da grelha:

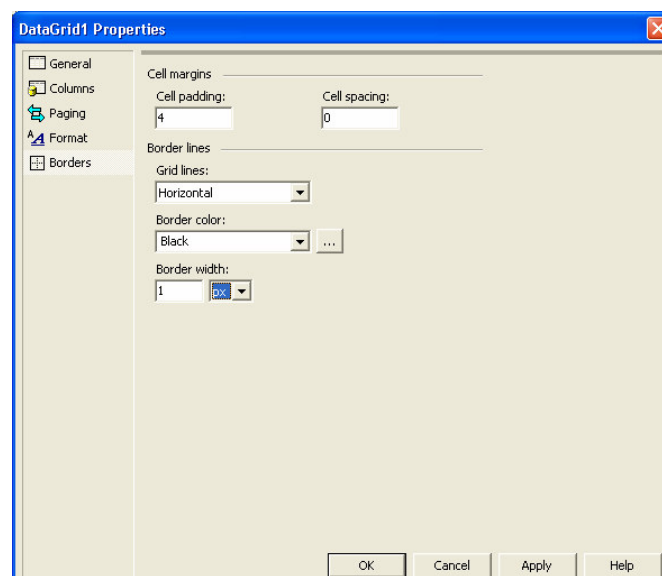


Figura 13: Borders

Exercício 10

1. Na página Borders, coloque a propriedade GridLines = Horizontal. As linhas verticais serão retiradas da grelha
2. Coloque a propriedade Border Color = black

10. Colunas

Em relação às páginas do Property Builder falta falar da página referente às colunas:

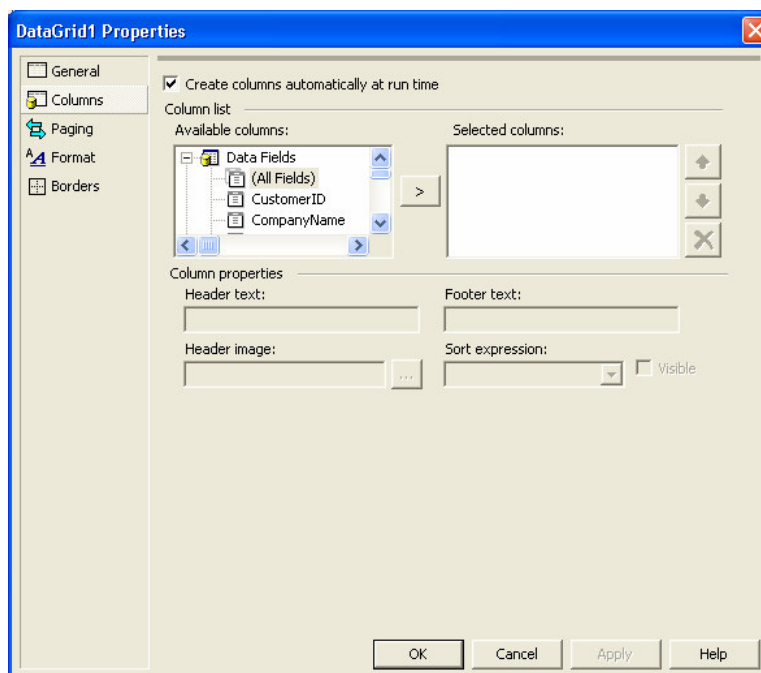


Figura 14: Colunas

A primeira opção nas colunas é “Create Columns automatically at run-time”. Se essa opção se mantiver activa, a DataGrid vai criar uma coluna para cada campo do comando Select. Com esta opção activa, boa parte das funcionalidades da grelha não estão acessíveis.

Na prática, a opção só se deve manter se a grelha servir apenas para consulta de dados. Mesmo nesse caso, deve haver um cuidado mínimo que implique a alteração dos títulos, isto porque os títulos a aplicar na grelha são os nomes das colunas nos comandos SQL. Esses nomes podem não ser o que o utilizador espera ver.

O que se deve fazer:

- Nos comandos SQL definir “alias” para os nomes das colunas.
- Exemplo: Select CompanyName as “Nome da Empresa” . . .

Exercício 11

Vamos alterar a página de forma a mostrar títulos amigáveis para o utilizador final:

1. Seleccionar o componente DataAdapter (nome = SqlDataAdapter1, se não foi alterado)
2. Na janela de propriedades, escolher a propriedade: SelectCommand e abri-la (carregar no mais)

3. Na sub-Propriedade CommandText, alterar o SQL de forma a usar alias:

```
SELECT CustomerID AS ID, CompanyName AS Nome,
       ContactName AS Contacto, ContactTitle AS Título,
       Address AS Endereço, City AS Cidade,
       Region AS Região, PostalCode AS [Cód. Postal],
       Country AS País, Phone AS Telefone, Fax AS Fax
FROM Customers
```

4. Sobre o componente DataAdapter, escolher o menu Generate Dataset.
5. Ao gerar o DataSet, certifique-se que vai gerar sobre o DataSet já existente (escolher opção Existing e verificar o nome do DataSet escolhido)
6. Nesta fase, a DataGrid pode aparecer como um bloco a conzento. Isto pode acontecer devido ao facto de haver propriedades inconsistentes que tornam impossível o desenho da grelha (em design time).
7. Seleccione a grelha e escolha a opção do menu “Property Builder”
8. Na página General, altere a propriedade Data Key Field = ID
9. Na página Columns, certifique-se que a opção “Create Columns automatically at run-time” está activa.
10. Corra a aplicação

ID	Nome	Contacto	Título	Endereço	Cidade	Região	Cód. Postal	País
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin		12209	Germany
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.		05021	Mexico

Figura 15: Título das Colunas alterado

Se analisarmos a aplicação a correr, verificamos que os títulos da grelha já estão em português. No entanto, se ordenarmos ou usarmos a paginação, a aplicação dá um erro. Isto deve-se ao facto de todo o código para ordenar e paginar usar o comando SQL original e não este novo comando SQL e assim, todos os nomes das colunas estão alterados.

Em vez de tentarmos resolver este problema (que resolve-se facilmente) vamos repor o SQL Original. O objectivo deste exercício era apenas mostrar que a utilização de “alias” no SQL permite criar facilmente grelhas apenas para consulta de dados.

Exercício 12

1. Reponha o comando SQL original no DataAdapter:

```
SELECT CustomerID, CompanyName, ContactName, ContactTitle,
       Address, City, Region, PostalCode, Country, Phone, Fax
FROM Customers
```

2. Faça Generate Dataset, novamente para o mesmo DataSet
3. Recoloque a propriedade Data Key Field = CustomerID
4. Corra a aplicação e certifique-se que tudo está correcto (ordenação/paginação)

A partir de agora, em vez de usarmos as colunas por defeito, vamos criar colunas manualmente.

Exercício 13

1. Escolha a página Columns e desactive a opção “Create Columns automatically at run-time”

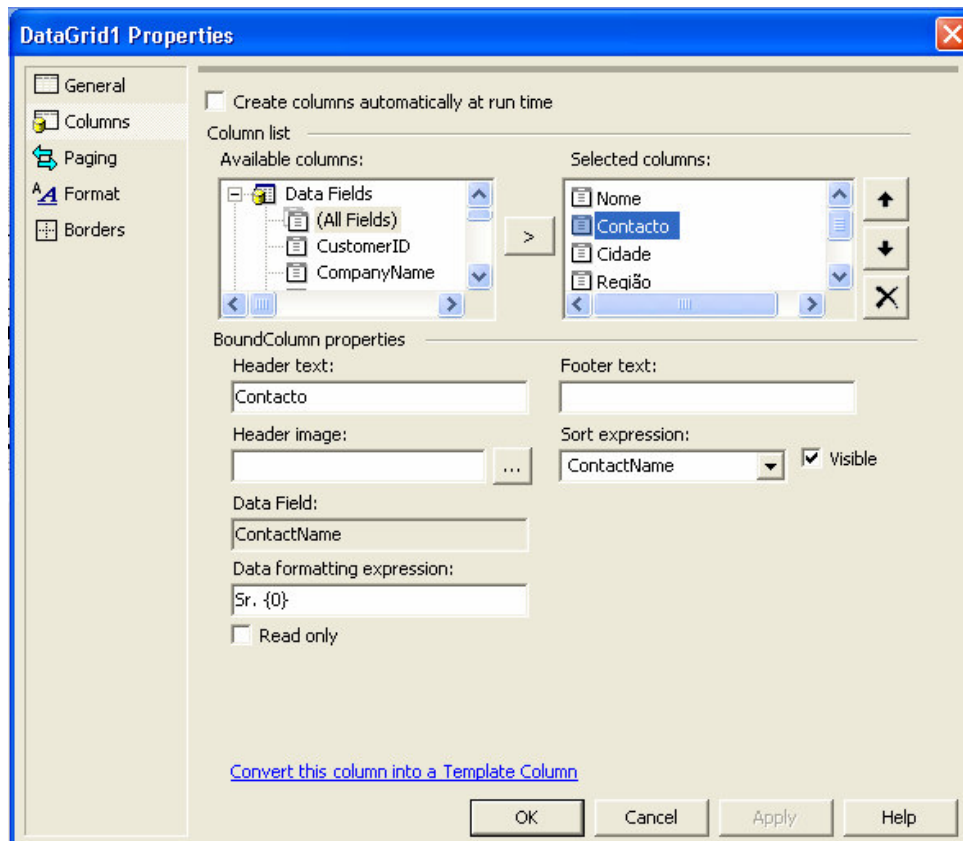


Figura 16: Definição manual de Colunas

2. Na lista de colunas disponíveis escolha a coluna CompanyName e pressione no botão (>) para a colocar na lista de colunas seleccionadas (Selected Columns)
3. No Header Text (título do cabeçalho) coloque o valor: Nome
4. Repita o passo 2 para as colunas ContactName, City, Region, Country
5. Repita o passo 3, com os textos: Contacto, Cidade, Região e País.
6. Selecione a coluna Contacto.
7. Na Propriedade Data Formatting Expression, coloque o valor: Sr.{0}
8. Corra a aplicação.

Nota:

Neste ponto do exemplo, existe um problema conceptual da página, porque apesar de apenas mostrarmos 5 campos, o comando SQL devolve 11 campos. Desses 11 campos, apenas 6 são necessários (5 colunas + chave primária). Numa situação real, deveria restringir o comando SQL para trazer apenas as colunas necessárias. Neste caso não o vamos fazer, para não termos que refazer o exercício completo.

ATENÇÃO

Os comandos SQL devem restringir as colunas devolvidas ao mínimo necessário para criar e manipular a página

Expressões

Várias propriedades das colunas admitem expressões (como no exemplo: Data Formatting Expression). Essas expressões são aplicadas aos dados da base de dados, antes de processar a página.

As expressões podem consistir em texto estático mais um especificador de formato que toma uma das seguintes formas: {0} ou {0:especificador formato}. Esse especificador refere-se ao valor que está no campo da base de dados (no exemplo, refere-se ao campo Contact).

Na prática, se as expressões forem preenchidas, o resultado será composto pelo que ficar definido na expressão, substituindo o local onde está {0} pelo valor que vem do campo.

Uma vez que os valores dos campos podem ser de diversos tipos (números, datas, etc...) o especificador pode também definir qual o formato a usar. Alguns exemplos:

- “{0:D4}” – Escreve o valor do campo (um inteiro) no mínimo com 4 dígitos (se necessário preenche com zeros). Por exemplo: 0023.
- “Preço: {0:C}” - Escreve a string “Preço:” seguida do valor do campo (numérico) usando o formato da moeda aplicado no servidor Web

Exercício 14

Depois de definir colunas “à mão”, também pode definir um formato diferente para cada uma delas.

1. Abra a página Format. Nesta fase, já aparecem itens relativos às colunas

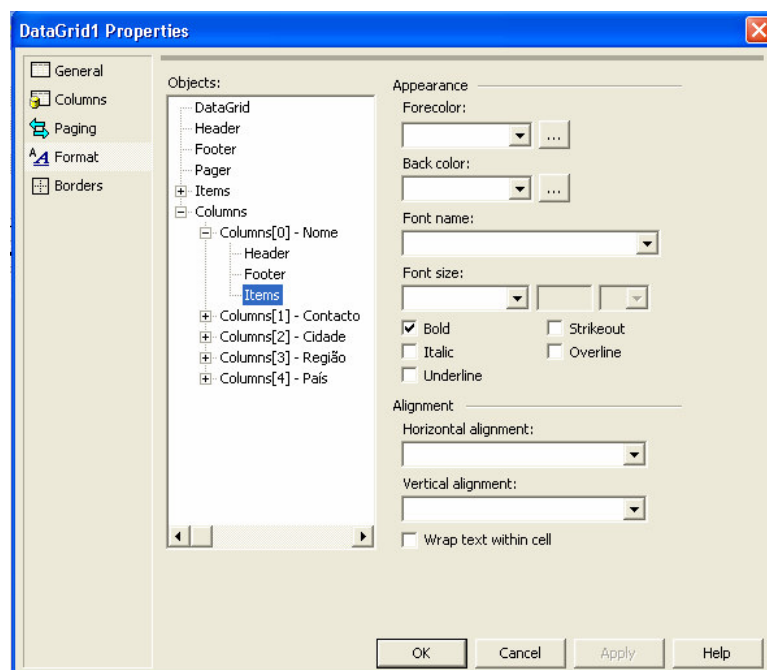


Figura 17: Formatação das colunas

2. Escolha a coluna 0 – Nome
3. Active a propriedade Bold
4. Desactive a propriedade Wrap text within cell
5. Repita o passo 4 para todas as colunas criadas.
6. Corra a aplicação

<u>Nome</u>	<u>Contacto</u>	<u>Cidade</u>	<u>Região</u>	<u>País</u>
Alfreds Futterkiste	Sr. Maria Anders	Berlin		Germany
Ana Trujillo Emparedados y helados	Sr. Ana Trujillo	México D.F.		Mexico
Antonio Moreno Taqueria	Sr. Antonio Moreno	México D.F.		Mexico
Around the Horn	Sr. Thomas Hardy	London		UK
Berglunds snabbköp	Sr. Christina Berglund	Luleå		Sweden
1 2 3 4 5 6 7 8 9 10 ...				

Figura 18: Grelha após a formatação das colunas