

Perfiles y otros controles ASP.NET

Indice

Nº- 14 Perfiles.....	1
1. Personalización.....	1
1.1 Crear un perfil de usuario.....	4
1.2 Introducir código.....	7
1.3 Utilizar controles de validación.....	12
2. Utilizar el diseñador de formularios.....	13
Objetos en planos.....	15
2.1 Insertar imágenes.....	20
2.2 Crear una tabla de hipervínculos.....	21
3. Enlazar datos a controles.....	22
Mostrar valores únicos.....	25
4. Controles desplegables.....	33
4.1 Utilizar listas desplegables para filtrar registros.....	36
5. Control DataList.....	43
5.1 Dar formato al control.....	45
5.2 Utilizar los controles DataList para mostrar imágenes.....	48
5.3 Utilizar un DataList para mostrar hiperenlaces.....	58
6. El control FormView.....	62
6.1 Mostrar subtotales.....	66
7. Despedida.....	70
Ejercicios.....	72
Ejercicio 1.....	72
Ejercicio 2.....	72

Nº- 14 Perfiles

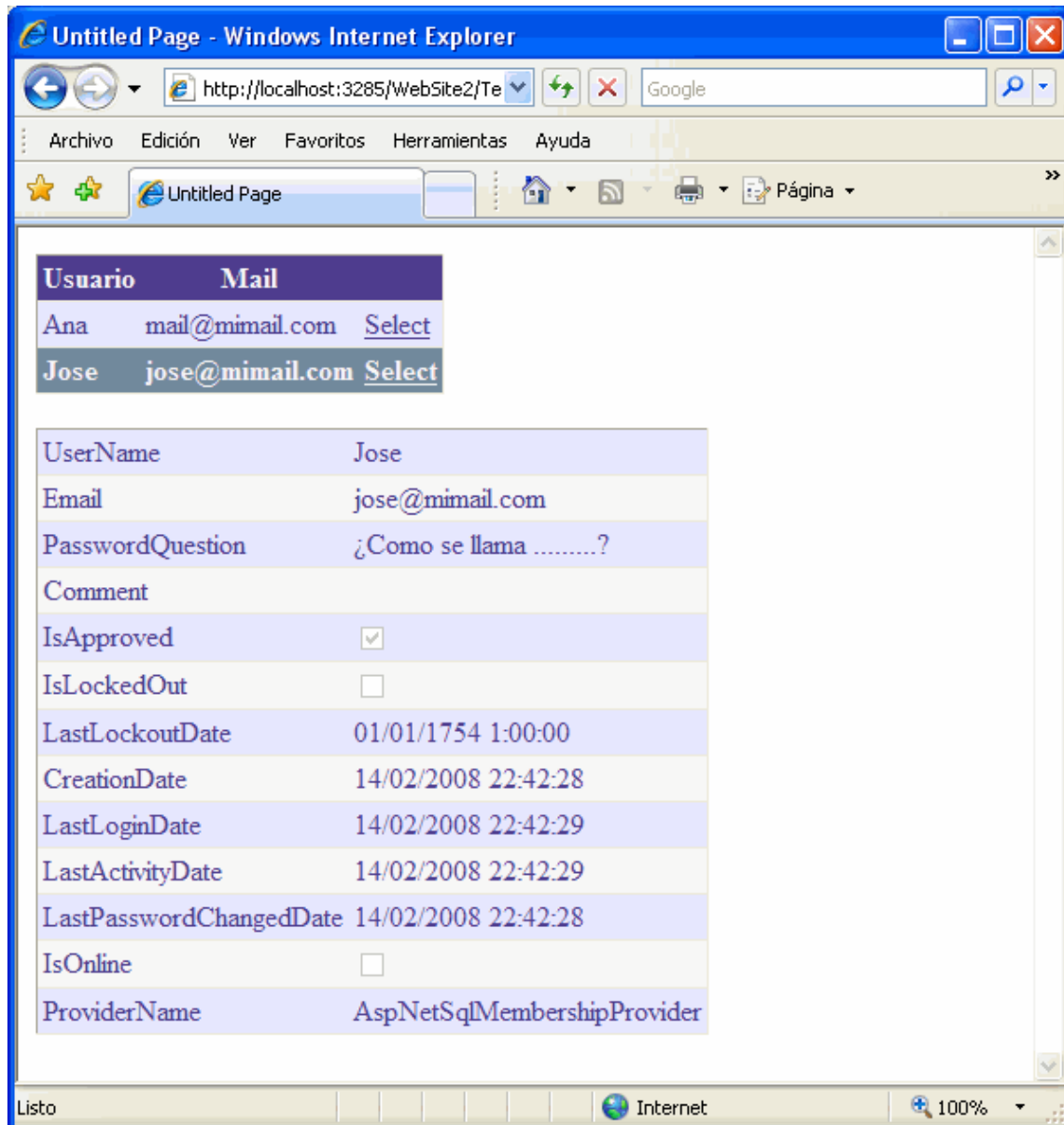
1. Personalización

Sabemos que cuando configuramos nuestro sitio Web para que almacene los usuarios se crean unas tablas con la información necesaria para almacenar estos campos: usuario, mail, dirección, ... si queremos modificar esta información, por ejemplo para añadir campos para el teléfono, ... debemos definir unas propiedades del perfil.

La idea es la siguiente: cada usuario autenticado tiene un perfil que contiene su información que sería por ejemplo: "Profile.propiedad" que almacenará cada uno de estos valores, así que lo primero que haremos será definir ese perfil con los campos que queremos para nuestra lista de usuarios.

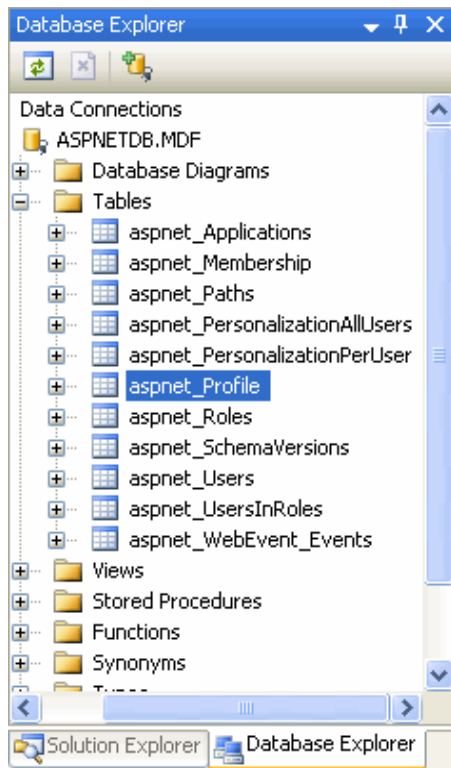
La ventaja de los perfiles es proporcionar una forma transparente de administrar información específica de usuarios sin tener que modificar datos ni código de ASP.NET. Me explico, queremos que nuestros usuarios tengan un campo mas para poner el número de cliente. Esto no lo tenemos en nuestra base de datos actual, que si recuerdas tenía los campos siguientes:

Perfiles y otros controles ASP.NET



Queremos añadir mas campos para personalizar nuestra base de datos pero claro no queremos escribir código especial adicional para que podamos editar esos nuevos campos y menos modificar la estructura de la base de datos de SQL Server! Si abres la estructura de la base de datos de nuestro SQL Server 2.005 Express verás que tiene una serie de tablas:

Perfiles y otros controles ASP.NET



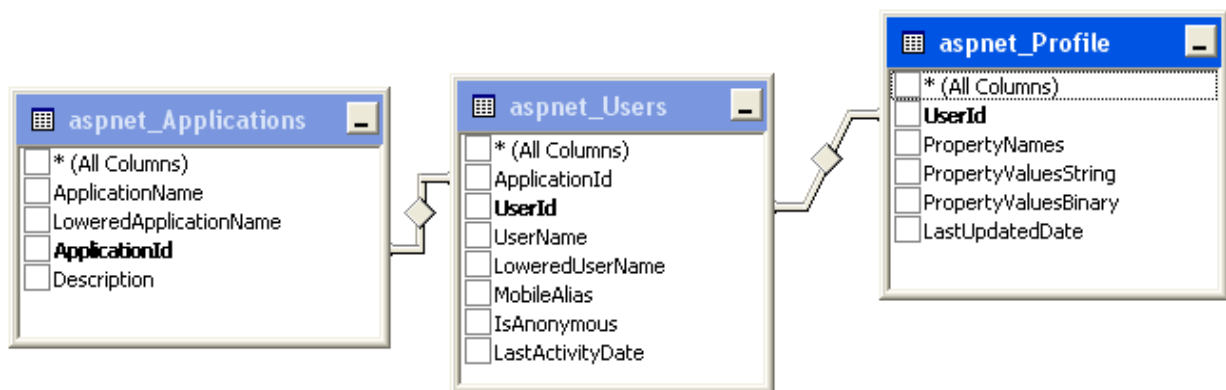
Veamos para que son estas tablas:

Atributo	Descripción															
aspnet_Applications	<p>Lista todas las aplicaciones web que tienen registros en estas bases de datos. Es posible que varias aplicaciones web utilicen la misma base de datos de aspnetdb. De esta forma podemos diferenciar los perfiles para asignárselos a cada aplicación. En nuestro ejemplo:</p> <table><tr><th></th><th>ApplicationName</th><th>LoweredApplica...</th><th>ApplicationId</th><th>Description</th></tr><tr><td>▶</td><td>/</td><td>/</td><td>24f9994b-b98a-...</td><td>NULL</td></tr><tr><td>*</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table> <p>Hay una sola aplicación web en la raíz del web.</p>		ApplicationName	LoweredApplica...	ApplicationId	Description	▶	/	/	24f9994b-b98a-...	NULL	*	NULL	NULL	NULL	NULL
	ApplicationName	LoweredApplica...	ApplicationId	Description												
▶	/	/	24f9994b-b98a-...	NULL												
*	NULL	NULL	NULL	NULL												
apsnet_Profile	Almacena la información de los perfiles de usuarios. Cada registro contiene la información completa del perfil del usuario. Properynames lista las propiedades y PropertyValuesString-PropertyValuesBINARY muestra sus datos si son de caracteres o numéricos.															
aspnet_SchemaVersions	Enumera los esquemas disponibles para los perfiles															
aspnet_Users	Lista los usuarios y los enlaza con las aplicaciones de la primera tabla. En nuestro ejemplo tenemos el campo:															

Perfiles y otros controles ASP.NET

	ApplicationId	UserId	UserName	LoweredUserN
▶	11c-3d3d1b207684	4dd4dcb7-bd70-...	Ana	ana
	24f9994b-b98a-...	d32b50f2-e49e-...	Jose	jose
	24f9994b-b98a-...	9cc8873c-96da-...	MiguelR	miguelr
*	NULL	NULL	NULL	NULL

La relación de las tablas es:



Vamos ya a definir perfiles...

Puesto que los perfiles son datos de los usuarios, necesitaremos lógicamente autenticar a los usuarios antes de leer o escribir información.

1.1 Crear un perfil de usuario

Un perfil de usuario consiste pues en un grupo de propiedades. A estas propiedades les pondremos un nombre descriptivo pero cortos y sin espacios en blanco: Nombre, Apellidos, Direccion, CodigoPostal, ... además definiremos un tipo de datos para cada uno de estos campos. Los campos que vamos a crear son:

- Nombre
- Apellidos
- Direccion1
- Direccion2
- Ciudad
- Provincia
- CodigoPostal
- Pais

Podremos poner algunas propiedades con valores predeterminados, por ejemplo para "Pais" podremos dejar fijo "España".

Configurar el perfil

Para modificar este perfil abriremos el fichero de configuración "Web.config", colócate justo antes de la etiqueta "</system.web>" y escribe la palabra "<profile>" verás como crea su pareja "</profile>":

```
<profile>

</profile>

</system.web>
```

Ahora escribiremos las propiedades para nuestro perfil:

```
<profile>
  <properties>

  </properties>
</profile>
```

Y dentro de las propiedades añadimos las que queremos que tenga nuestro perfil:

```
<properties>
  <add name="Nombre"/>
  <add name="Apellidos"/>
  <add name="Direccion1"/>
  <add name="Direccion2"/>
  <add name="Ciudad"/>
  <add name="Provincia"/>
  <add name="CodigoPostal"/>
  <add name="Pais" />
  <add name="FechaNacimiento"/>
  <add name="TemaPreferido"/>
</properties>
```

Ahora vamos a poner algún valor por defecto con la etiqueta: "defaultValue":

```
<properties>
  <add name="Nombre"/>
  <add name="Apellidos"/>
  <add name="Direccion1"/>
  <add name="Direccion2"/>
  <add name="Ciudad"/>
  <add name="Provincia"/>
  <add name="CodigoPostal"/>
  <add name="Pais" defaultValue="España" type="System.String"/>
  <add name="FechaNacimiento"/>
  <add name="TemaPreferido" defaultValue="Clasico"/>
</properties>
```

Bueno, ahora nos queda el tipo de datos, se podría omitir pero siempre es mejor ponerlo:

Perfiles y otros controles ASP.NET

```
<properties>
  <add name="Nombre" type="System.String"/>
  <add name="Apellidos" type="System.String"/>
  <add name="Direccion1" type="System.String"/>
  <add name="Direccion2" type="System.String"/>
  <add name="Ciudad" type="System.String"/>
  <add name="Provincia" type="System.String"/>
  <add name="CodigoPostal" type="System.String"/>
  <add name="Pais" defaultValue="España" type="System.String"/>
  <add name="FechaNacimiento" type="System.DateTime"/>
  <add name="TemaPreferido" defaultValue="Clasico"/>
</properties>
```

Si hubiéramos omitido ese valor se asumiría el tipo de datos "System.String" pero bueno, ya aprendimos que es muy buena técnica definir siempre los tipos de datos que se utilizan. En el ejemplo he puesto debajo que además utilice un tema para los estilos. No lo vamos a utilizar ahora en el ejemplo pero es una buena técnica para seguir teniendo un estilo coherente en nuestro web, así que de momento lo quitamos para seguir con los otros campos.

Los atributos o propiedades que podemos poner son los siguientes:

Atributo	Descripción
name	Nombre de la propiedad
type	Nombre completo de la clase del tipo de datos
serializeAs	Formato interno para la base de datos: String, Binary, XML o ProviderSpecific. Sin uso para nosotros
readOnly	Valor booleano que indica si este valor se puede modificar o no
defaultValue	Valor predeterminado para el campo
allowAnonymous	Valor booleano que indica si esta propiedad se puede utilizar para perfiles anónimos (lo veremos mas adelante)
provider	Proveedor que proporciona esta propiedad. Sin uso para nosotros.

Introducir información

El objetivo de la creación de un perfil de este tipo es que los usuarios escriban sus propios datos no que los mantengamos nosotros, así que vamos a proporcionar un formulario para que ellos mismos cambien sus datos.

Vamos a crear un formulario con los controles necesarios para que los usuarios puedan rellenar sus datos. Crea una página de tipo "Web Form" con el nombre "crearperfil.aspx"

	td																																																	
Nombre	<input type="text"/>																																																	
Apellidos	<input type="text"/>																																																	
Dirección 1	<input type="text"/>																																																	
Dirección 2	<input type="text"/>																																																	
Ciudad	<input type="text"/>																																																	
Provincia	<input type="text"/>																																																	
Código postal	<input type="text"/>																																																	
Fecha nacimiento	<div> <div><</div> <div>marzo de 2008</div> <div>></div> </div> <table> <tr> <td>lun</td> <td>mar</td> <td>mié</td> <td>jue</td> <td>vie</td> <td>sáb</td> <td>dom</td> </tr> <tr> <td>25</td> <td>26</td> <td>27</td> <td>28</td> <td>29</td> <td>1</td> <td>2</td> </tr> <tr> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> </tr> <tr> <td>10</td> <td>11</td> <td>12</td> <td>13</td> <td>14</td> <td>15</td> <td>16</td> </tr> <tr> <td>17</td> <td>18</td> <td>19</td> <td>20</td> <td>21</td> <td>22</td> <td>23</td> </tr> <tr> <td>24</td> <td>25</td> <td>26</td> <td>27</td> <td>28</td> <td>29</td> <td>30</td> </tr> <tr> <td>31</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> </tr> </table>	lun	mar	mié	jue	vie	sáb	dom	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6
lun	mar	mié	jue	vie	sáb	dom																																												
25	26	27	28	29	1	2																																												
3	4	5	6	7	8	9																																												
10	11	12	13	14	15	16																																												
17	18	19	20	21	22	23																																												
24	25	26	27	28	29	30																																												
31	1	2	3	4	5	6																																												
Pais	<input type="text"/>																																																	
Estilo de página:	<div>Clásico</div> <div>Moderno</div>																																																	
	<input type="button" value="Guardar datos"/>																																																	

Ahora debemos añadir un botón como ves en la figura anterior. De nombre ponle "btnenviar". Recuerda poner al principio del nombre el tipo de control que es: txt para cuadros de texto, btn para botones, luego me agradecerás esto porque nos va a facilitar mucho la tarea de leer el código.

1.2 Introducir código

Las propiedades de este perfil serán las que creamos antes en nuestro fichero de configuración: "Profile.Apellidos". Así que como te imaginas asignaremos a estos valores del perfil el contenido de los cuadros de texto:

```
Profile.nombre=txtnombre.text
```

```

Partial Class crearperfil
    Inherits System.Web.UI.Page

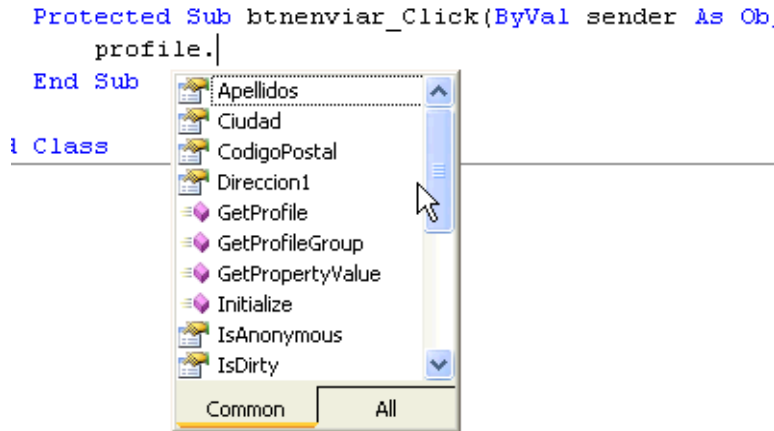
    Protected Sub btnenviar_Click(ByVal sender As Object, E

    End Sub
End Class

```

Perfiles y otros controles ASP.NET

Al seleccionar el control y pulsar en el icono que ves en pantalla podemos ver la lista de eventos que tiene cada control. Los botones no tienen el evento de doble clic así que cuando hacemos esta acción lo que nos muestra es la del clic simple. Si hacemos doble clic sobre el evento de esa lista nos creará el procedimiento "Sub" vacío listo para meter datos. Luego además nos mostrará en esta lista el nombre de este procedimiento así al ir haciendo clic en los distintos controles de la página sabremos enseguida que eventos tienen algo de código escrito.



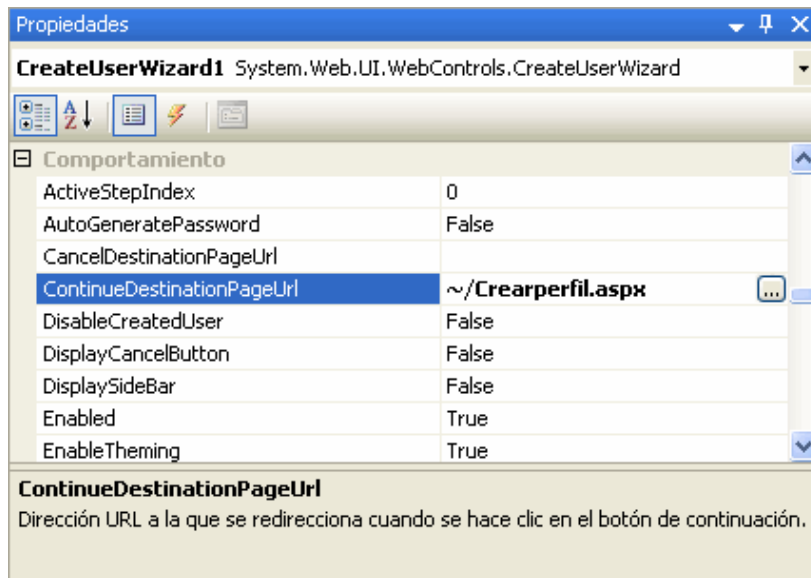
Escribimos pues todo el código para guardar el perfil:

```
Protected Sub btnenviar_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnenviar.Click
    Profile.Nombre = txtnombre.Text
    Profile.Apellidos = txtapellidos.Text
    Profile.Direccion1 = txtdireccion1.Text
    Profile.Direccion2 = txtdireccion2.Text
    Profile.Ciudad = txtciudad.Text
    Profile.Provincia = txtprovincia.Text
    Profile.CodigoPostal = txtcodigopostal.Text
    Profile.FechaNacimiento = fecha.SelectedDate
    Profile.Pais = txtpais.Text
    Profile.TemaPreferido = lista_estilos.SelectedValue
End Sub
```

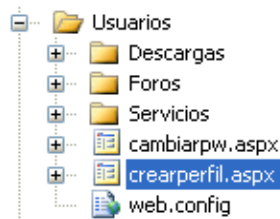
Como ves solo necesitamos asignar al perfil las propiedades que hemos creado.

Ya tenemos lo necesario para que el usuario ejecute la página y cambie su perfil. Si hemos utilizado el control "CreateUserWizard" el proceso es muy sencillo porque tenemos una propiedad especial para que el usuario continúe escribiendo sus datos, fíjate en las propiedades de ese control:

Perfiles y otros controles ASP.NET



Una cosa, como a esta página sólo debemos acceder si es un usuario autenticado la pondríamos dentro de una carpeta de usuarios que no permite el acceso a la gente anónima:



Ahora podemos ver si funciona todo el proceso. Empezamos ejecutando la página de crear la cuenta:

Nombre de usuario:	<input type="text" value="Carmen"/>
Contraseña:	<input type="password" value="....."/>
Confirmar contraseña:	<input type="password" value="....."/>
Correo electrónico:	<input type="text" value="carmen@mail.com"/>
Pregunta de seguridad:	<input type="text" value="ninguna"/>
Respuesta:	<input type="text" value="ninguna"/>
<input type="button" value="Crear usuario"/>	

Metemos unos datos y pulsamos en "Create User":

Terminada

Cuenta creada correctamente

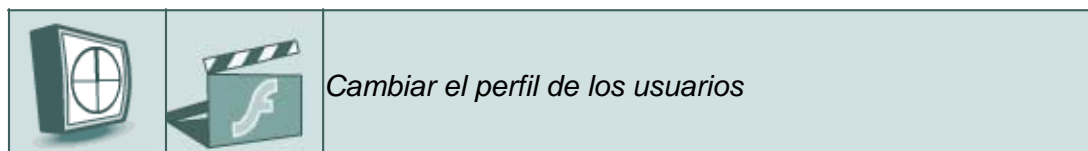
Continuar

La creación ha sido correcta así que pulsamos en "Continue"...

Personaliza tus datos

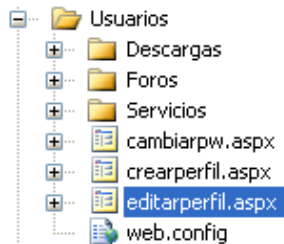
Nombre	<input type="text"/>																																																	
Apellidos	<input type="text"/>																																																	
Dirección 1	<input type="text"/>																																																	
Dirección 2	<input type="text"/>																																																	
Ciudad	<input type="text"/>																																																	
Provincia	<input type="text"/>																																																	
Código postal	<input type="text"/>																																																	
Fecha nacimiento	<div> <div>≤</div> <div>marzo de 2008</div> <div>≥</div> </div> <table> <tr> <th>lun</th> <th>mar</th> <th>mié</th> <th>jue</th> <th>vie</th> <th>sáb</th> <th>dom</th> </tr> <tr> <td><u>25</u></td> <td><u>26</u></td> <td><u>27</u></td> <td><u>28</u></td> <td><u>29</u></td> <td><u>1</u></td> <td><u>2</u></td> </tr> <tr> <td><u>3</u></td> <td><u>4</u></td> <td><u>5</u></td> <td><u>6</u></td> <td><u>7</u></td> <td><u>8</u></td> <td><u>9</u></td> </tr> <tr> <td><u>10</u></td> <td><u>11</u></td> <td><u>12</u></td> <td><u>13</u></td> <td><u>14</u></td> <td><u>15</u></td> <td><u>16</u></td> </tr> <tr> <td><u>17</u></td> <td><u>18</u></td> <td><u>19</u></td> <td><u>20</u></td> <td><u>21</u></td> <td><u>22</u></td> <td><u>23</u></td> </tr> <tr> <td><u>24</u></td> <td><u>25</u></td> <td><u>26</u></td> <td><u>27</u></td> <td><u>28</u></td> <td><u>29</u></td> <td><u>30</u></td> </tr> <tr> <td><u>31</u></td> <td><u>1</u></td> <td><u>2</u></td> <td><u>3</u></td> <td><u>4</u></td> <td><u>5</u></td> <td><u>6</u></td> </tr> </table>	lun	mar	mié	jue	vie	sáb	dom	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>
lun	mar	mié	jue	vie	sáb	dom																																												
<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>1</u>	<u>2</u>																																												
<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>																																												
<u>10</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>																																												
<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>																																												
<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>																																												
<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>																																												
País	<input type="text"/>																																																	
Estilo de página:	<div>Clásico</div>																																																	
<div>Guardar datos</div>																																																		

¡Bien! Ha funcionado, la propiedad que indicaba la siguiente página a utilizar después de ejecutar el asistente de creación de usuario ha funcionado y nos ha mostrado esta página donde el usuario puede personalizar sus datos.

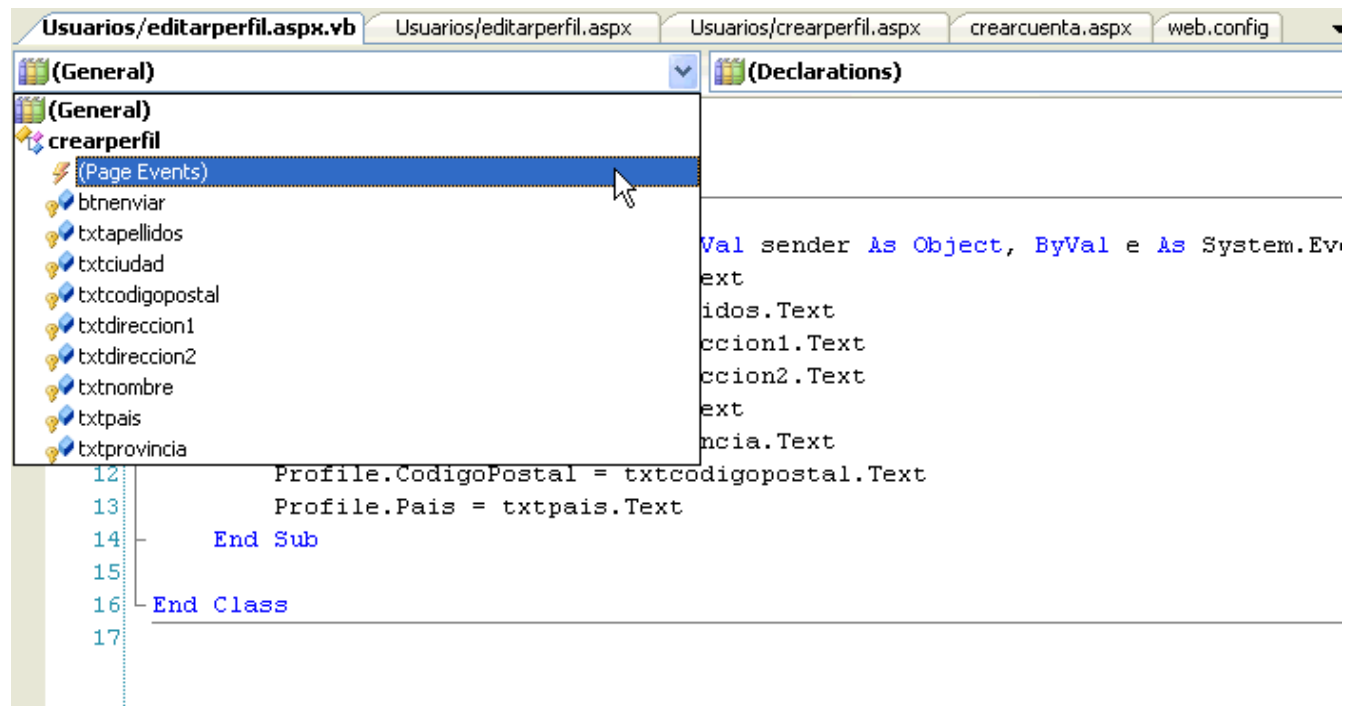


Permitir que el usuario modifique sus datos

Una vez que el usuario ha hecho el inicio de sesión y está navegando por el sitio Web debe poder acceder a la página de los datos para poder cambiar sus datos personales. Pero antes de mostrarle esta página debemos cargar sus datos ya que si ya tiene unos introducidos de antes debe modificarlos. Copia la página anterior de la creación del perfil y la pegas con el nombre de "editarperfil.aspx":



Debemos cargar los datos al inicio de la página así que recuerda que teníamos un procedimiento especial llamado "Sub Page_Load". Para que nos muestre los eventos disponibles en las páginas y así que nos lo ponga de forma automática en el editor como hicimos con el clic del botón haremos lo siguiente. En el editor hacemos clic en el desplegable la parte superior izquierda:



Para poder mostrar los eventos de la página y buscamos el Load para ponerle la carga de los datos:

Perfiles y otros controles ASP.NET

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles  
    If Not Page.IsPostBack Then  
        txtnombre.Text = Profile.Nombre  
        txtapellidos.Text = Profile.Apellidos  
        txtdireccion1.Text = Profile.Direccion1  
        txtdireccion2.Text = Profile.Direccion2  
        txtciudad.Text = Profile.Ciudad  
        txtprovincia.Text = Profile.Provincia  
        txtcodigopostal.Text = Profile.CodigoPostal  
        fecha.SelectedDate = Profile.FechaNacimiento  
        txtpais.Text = Profile.Pais  
        lista_estilos.SelectedValue = Profile.TemaPreferido  
    End If  
End Sub
```

1.3 Utilizar controles de validación

Pondremos controles de validación para los campos que consideremos obligatorios:

The screenshot shows a web form with the following fields and controls:

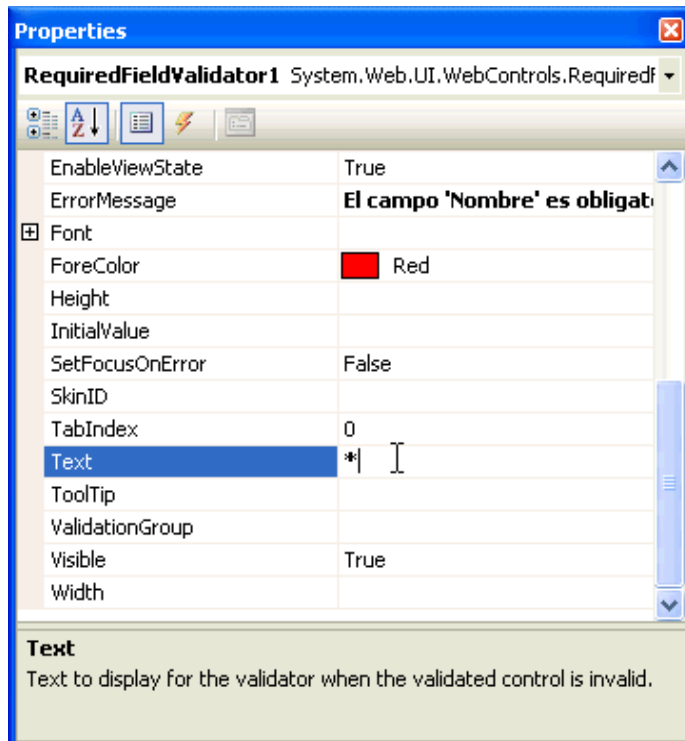
- Nombre**: Text box with a red asterisk validation icon.
- Apellidos**: Text box.
- Dirección 1**: Text box.
- Dirección 2**: Text box.
- Ciudad**: Text box.
- Provincia**: Text box.
- Código postal**: Text box.
- Fecha nacimiento**: Calendar control showing dates from January 1st to March 31st.
- Pais**: Text box.
- Estilo de página:** Dropdown menu with options 'Clásico' and 'Moderno', and a 'Guardar' button.

The **Propiedades** (Properties) window for the **RequiredFieldValidator1** control is open, showing the following settings:

- RequiredFieldValidator1** (System.Web.UI.WebControls.RequiredFieldValidator)
- TabIndex**: 0
- Apariencia** (Appearance):
 - BackColor**: (empty)
 - BorderColor**: (empty)
 - BorderStyle**: NotSet
 - BorderWidth**: (empty)
 - CssClass**: (empty)
 - Display**: Static
 - ErrorMessage**: *
- Comportamiento** (Behavior):
 - ControlToValidate**: txtnombre
 - EnableClientScript**: True
 - Enabled**: True
- ErrorMessage**: Mensaje que se muestra en ValidationSummary cuando el control validado no es válido.

Perfiles y otros controles ASP.NET

Como ves en la imagen, hemos puesto las propiedades imprescindibles, en "ControlToValidate" selecciona el control con el queremos asociarlo y en "ErrorMessage" escribiremos el texto que mostrará si hacemos un resumen de validación que veremos mas adelante. Lo que queremos que nos muestre al usuario se pone en el campo "Text":



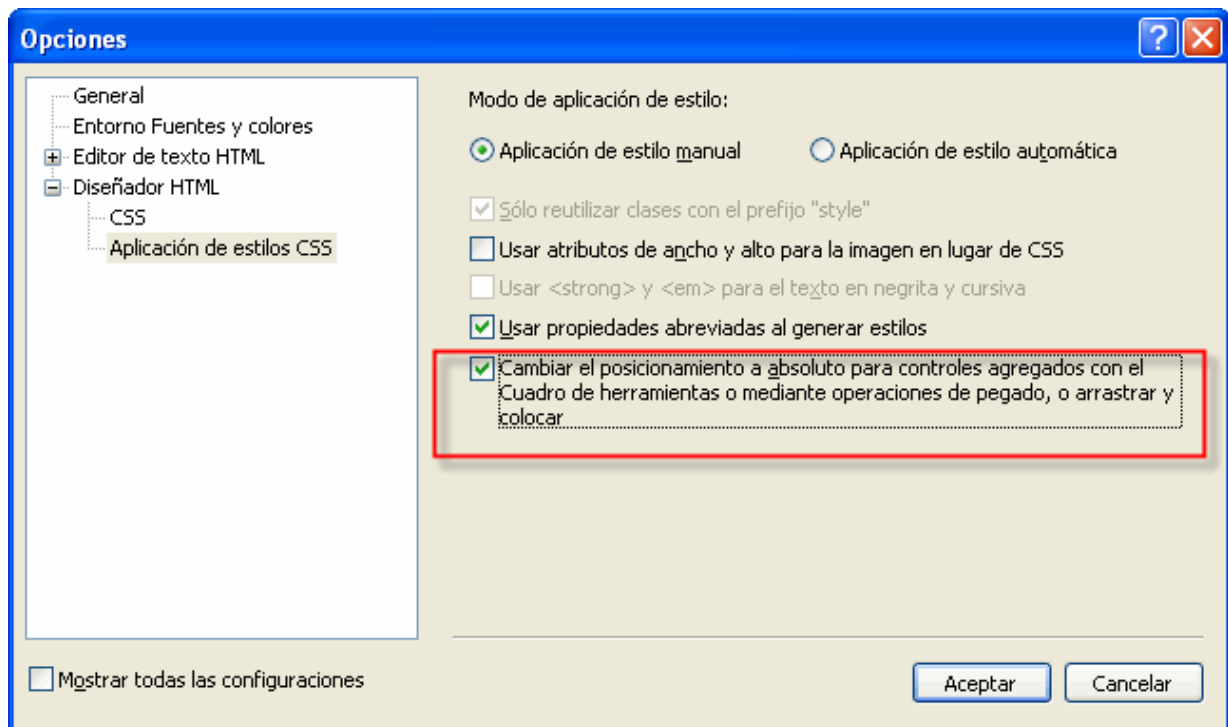
Que simplemente será un asterisco "*" de color rojo a la derecha del cuadro de texto, suficiente como para que el usuario lo identifique como que falta un valor.

2. Utilizar el diseñador de formularios.

A modo de recapitulación vamos a ver algunos detalles de la edición con el IDE que seguramente te habrá dado mas trabajo del que parecía al principio. En muchas ocasiones yo mismos he tenido que acudir a FrontPage para realizar una tabla o páginas con diseño para luego pegarla en mi página de ASP. Lo único malo de realizar esto es que los elementos que me coloca son de tipo HTML, así que si pongo en FrontPage cuadros de texto o botones, luego tendrás que acordarte de cambiarlos por tipos ASP: <input type= por <asp:button", ...

Sabemos utilizar la parte relacionada con ASP, poner controles, poner nombres a los controles, poner controles de validación y finalmente capturar los eventos de "submit" para realizar las operaciones necesarias. Todo esto en tablas ya formateadas para colocar correctamente los distintos controles.

Vamos a las opciones del IDE y seleccionamos la opción:



Que nos permitirá poner los controles de una forma mas libre por la pantalla y además con las opciones de abajo los podremos mover de 8 en 8 píxeles así será mas fácil de dejarlos todos con la misma distancia. Fíjate en el cambio de aspecto:

Personaliza tus datos

Nombre	Apellidos	
<input type="text"/>	<input type="text"/>	<input type="text"/>
Dirección 1	<input type="text"/>	
Dirección 2	<input type="text"/>	
Provincia	Dirección 2	Cod. Postal
<input type="text"/>	<input type="text"/>	<input type="text"/>
País	<input type="text"/>	<ul style="list-style-type: none"> • Error message 1. • Error message 2.

Nombre	<input type="text"/>
Apellidos	<input type="text"/>
Dirección 1	<input type="text"/>
Dirección 2	<input type="text"/>
Ciudad	<input type="text"/>
Provincia	<input type="text"/>
Código postal	<input type="text"/>
País	<input type="text"/>
<input type="button" value="Guardar datos"/>	

He puesto los controles libremente sin la tabla, al lado de dos de ellos (nombre y apellidos) un control de validación y abajo a la derecha un control resumen de validación. El aspecto ha mejorado pero habríamos obtenido parecido resultado con tablas y con bordes de tablas ocultos. Sin embargo es una forma muy práctica de colocar elementos arbitrariamente por la pantalla.

Objetos en planos

Al poner estos elementos de forma libre por la pantalla pueden quedar obviamente tapados unos por otros, así que entra en juego el control de las dimensiones o apilamiento que llamaremos "índice Z" y que define su posición en la pila de objetos.

Si tenemos un texto en pantalla con el valor de $z=0$ y ponemos un cuadro con el valor de $z=-1$ significa que este último cuando estará detrás de el texto. Por ejemplo tenemos un texto en nuestra página y he puesto un cuadro de texto de forma absoluta:

Este es un texto de prueba...

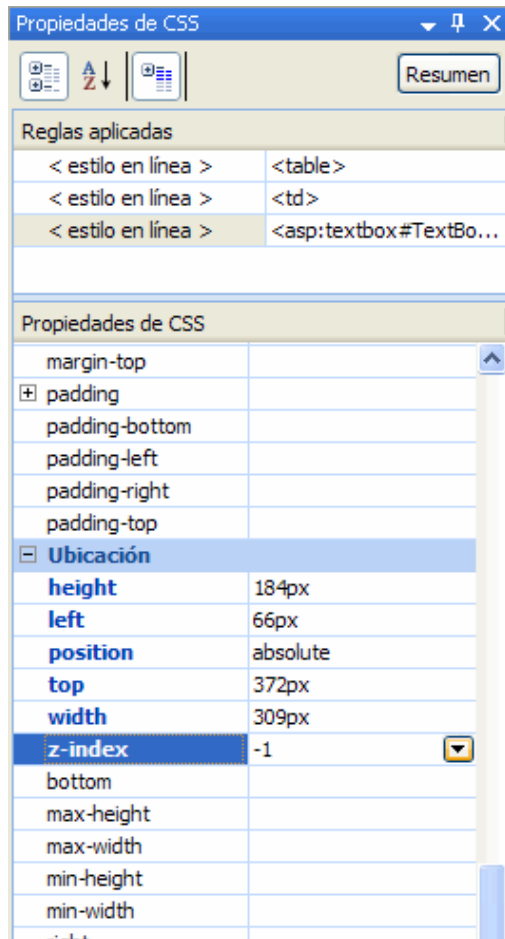
Al poner estos elementos de forma libre por la pantalla pueden quedar obviamente t
que entra en juego el control de las dimensiones o apilamiento que llamaremos "índic
en la pila de

Si tenemos un cuadro con el valor de $z=0$ y ponemos un cuadro con el val
último cuando estará detrás de el texto:

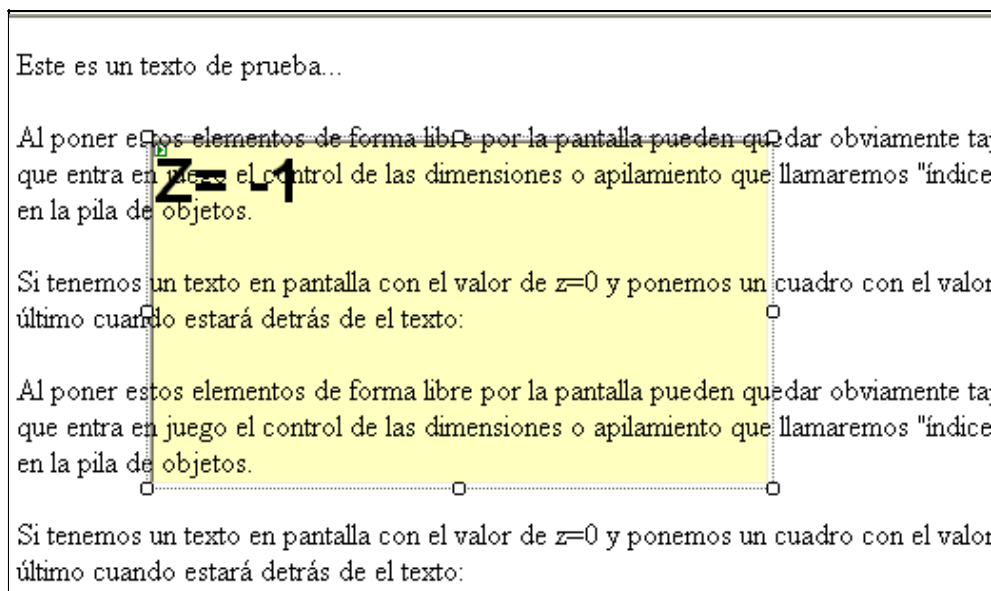
Al poner estos elementos de forma libre por la pantalla pueden quedar obviamente t
que entra en juego el control de las dimensiones o apilamiento que llamaremos "índic
en la pila de objetos.

Z= -1

Nos vamos a las propiedades de los estilos, que está en una pestaña debajo de los controles, a la izquierda de nuestro IDE:

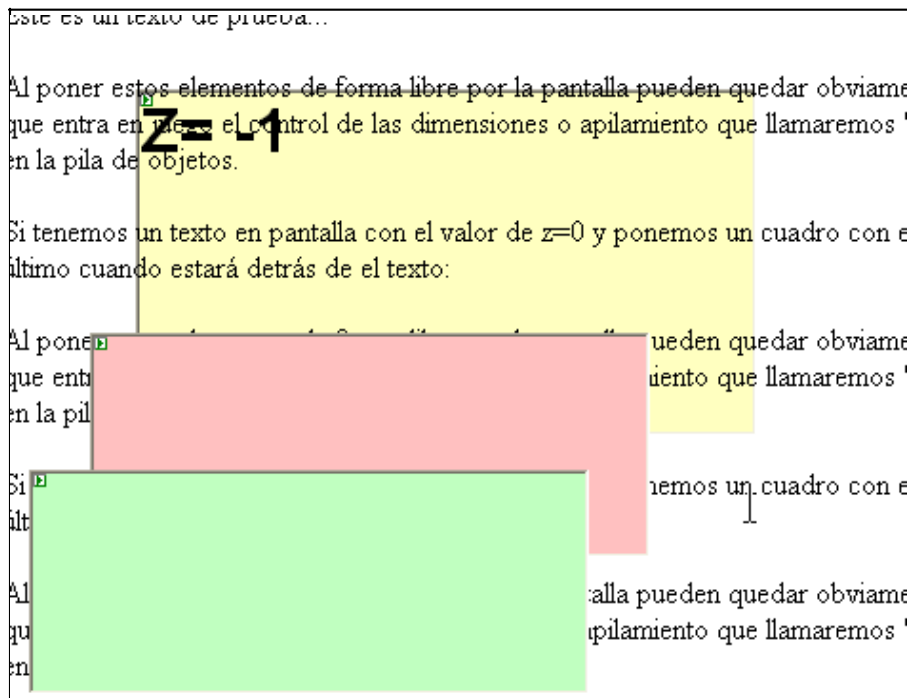


Como ves del índice Z que tenía le pongo un -1 para obtener entonces:

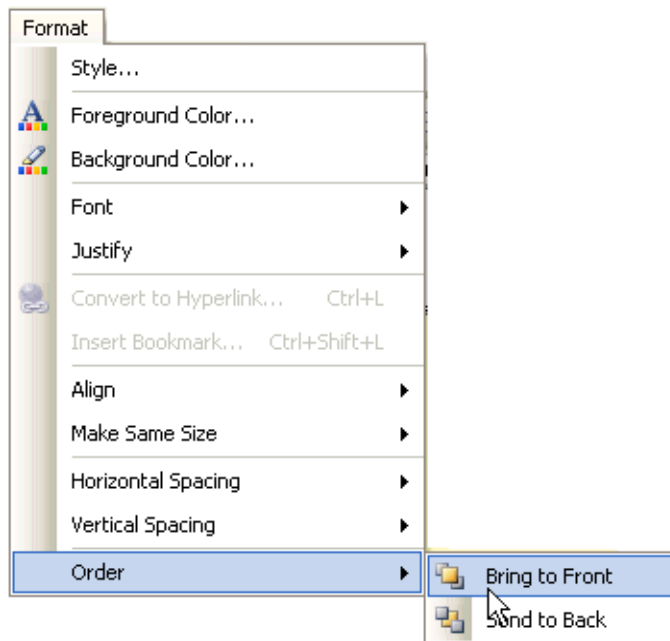


Si coloco dos cuadros mas de texto con Z de 1 y 2:

Perfiles y otros controles ASP.NET



Una forma rápida de colocar los elementos según ese valor Z es utilizando la opción:



Si ves el código fuente estos estilos se indican de esta forma:

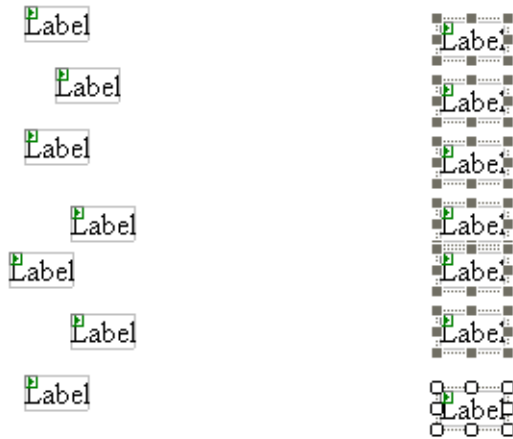
```
style="position:absolute; top:152px; left:56px; z-index:2"
```

Obviamente son las posiciones "top" y "left" respecto al origen de la página y el otro valor el del índice Z.

Alineación de controles

Para ayudarnos en este diseño libre de controles por la página podemos utilizar las opciones que hay en la sección de formato y así alinear los controles. Si has trabajado con editores de formularios como Visual Basic la filosofía es la misma.

Seleccionamos los controles que queremos alinear y seleccionamos la opción "align" del menú "format":



Y nos ajusta los controles a la derecha.

Ajustar tamaños

Caso parecido tenemos a la hora de ajustar el tamaño de los controles para hacerlos exactamente iguales. Los marcamos todos e iremos a la opción "Make Same Size" del menú "Format":



Una vez marcados y establecida esa acción si aumentamos con el ratón el tamaño de los controles se actualizará en todos a la vez.

Espacio entre controles

Para terminar disponemos de las mismas posibilidades a la hora de ponerles el mismo espacio a los controles. Utilizaremos para esto la opción "Horizontal Spacing" o "Vertical Spacing" del menú "Format":

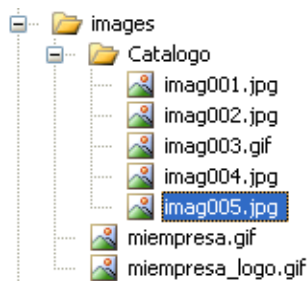


Así que por herramientas para dejar tus formularios en condiciones que no quede. Seguiremos ahora con la última parte antes de comenzar ya con las base

2.1 Insertar imágenes

En ocasiones necesitaremos incluir imágenes en nuestros registros, por ejemplo para un catálogo de artículos como este o para las fotografías de los usuarios. Estas imágenes no se guardan directamente en la base de datos porque la engordarían mucho y la harían mas ineficiente al tener que realizar un mantenimiento mas exhaustivo debido a su tamaño.

Lo habitual es que las imágenes asociadas a una tabla se almacenen en una carpeta y en la base de datos solo almacenemos la ruta y nombre de esa imagen. Dentro de la carpeta existente en nuestro proyecto de "images" voy a crear una que se llame "catalogo" y colocaré ahí varias imágenes que luego asociaré con la tabla de los artículos.




Ahora vamos a crear una tabla para tres campos: un identificador, una descripción y la ruta con nombre de archivo. Por supuesto con su clave primaria definida. El campo donde irá el nombre tendrá una información de este tipo:

```
~/images/catalogo/imag001.jpg
```

Nota El símbolo "~" que ya nos ha aparecido en alguna ocasión hace referencia a la raíz del web. Por lo tanto "~/images" es la carpeta "images" a partir de la raíz del Web.

Creamos entonces una nueva tabla con estos campos:

	Column Name	Data Type	Allow Nulls
	FotoID	int	<input type="checkbox"/>
	Descripcion	varchar(50)	<input type="checkbox"/>
	Ruta	varchar(64)	<input type="checkbox"/>

Que son los tres campos, el primero es un "identity" con un valor inicial de 1, nos da igual ese valor mientras sea único, lo que si será es clave única así que se lo marcas con el botón derecho. Los otros campos son pues para la descripción y la ruta de la imagen.

La diferencia entre los tipos de datos "varchar" y "nvarchar" es si recuerdas que los que no llevan la "n" son de longitud fija, así que rellena con espacios en blanco el texto no utilizado. Introduce a mano unos datos para la prueba:

	FotoID	Descripcion	Ruta
	2	Descripcion 1	~/images/catalogo/imag001.jpg
	3	Descripción 2	~/images/catalogo/imag002.jpg
	4	Descripción 3	~/images/catalogo/imag003.jpg
	5	Descripción 4	~/images/catalogo/imag004.jpg
	6	Descripción 5	~/images/catalogo/imag005.jpg

Una cosa, en el campo "FotoID" le puse como tipo de datos "Int", recuerda que tiene un topo de 32.000 valores así que si nuestra tabla de imágenes va a tener mas de esas filas debemos ponerle un entero largo como tipo de datos. Bien, de momento están ahí los enlaces, ahora veremos como establecer hipervínculos.

2.2 Crear una tabla de hipervínculos

Si nuestra sitio tiene muchos enlaces a sitios externos podemos crear una tabla para almacenarlos. Al tenerlos en un sitio organizado es mas fácil mantenerlos, de esta forma si los ponemos en varias páginas con solo administrar esta tabla quedan actualizadas todas las páginas y no tenemos que ir una a una a poner o actualizar enlaces.

Nuestra tabla sólo tendrá 3 campos, nuestro inevitable índice autonumérico que será clave primaria, el enlace y una descripción de él. Así que creamos la tabla:

	Column Name	Data Type	Allow Nulls
	EnlaceID	int	<input type="checkbox"/>
	NombreSitio	nvarchar(50)	<input type="checkbox"/>
	URL	nvarchar(64)	<input type="checkbox"/>

Selecciona con el botón derecho la opción de siempre de "Show Table Data" para introducir unos valores, recuerda poner el literal "http://" en las URL:

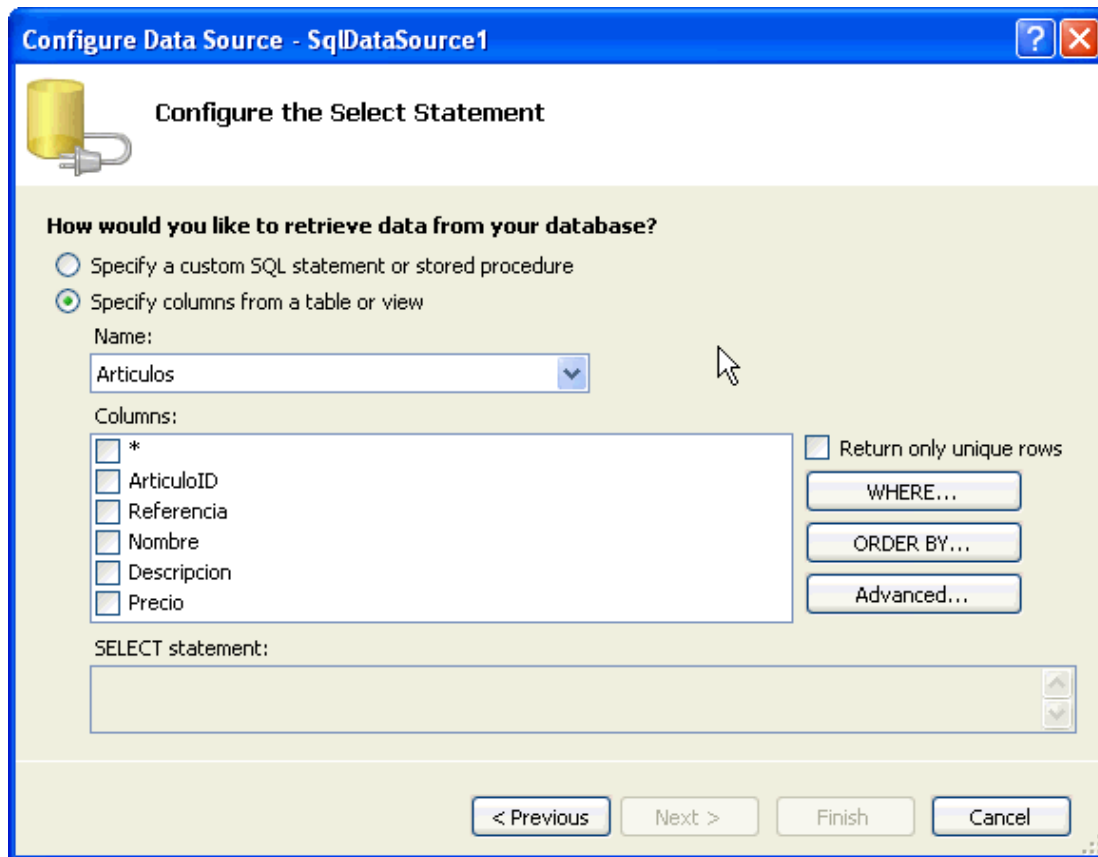
Perfiles y otros controles ASP.NET

	EnlaceID	NombreSitio	URL
	1	ASP.NET	http://www.asp.net
	2	Periodico El Mundo	http://www.elmundo.es
	3	Periodico EL Pais	http://www.elpais.es
	4	Microsoft	http://www.microsoft.com

Luego ya enlazaremos todo esto que estamos preparando... pero ¿se te ocurre que control es el mas adecuado para esto que acabamos de hacer? Pues si, si conseguimos que un cuadro de lista se autorellene con esta tabla obtendremos con dos líneas de código lo necesario para que al seleccionar un elemento de la lista el navegador se desplace a ese sitio Web.

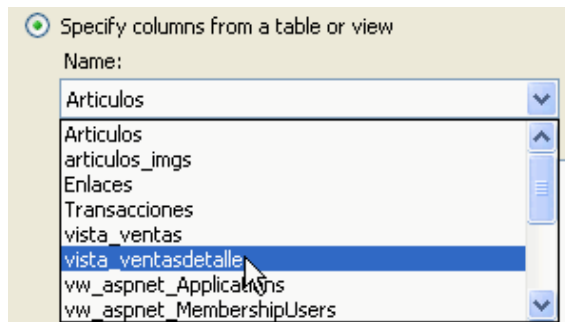
3. Enlazar datos a controles

También a modo de recapitulación vamos a ver unos detalles interesantes sobre las consultas a datos. Partimos desde la conocida pantalla donde ya hemos establecido una conexión con el servidor de base de datos:

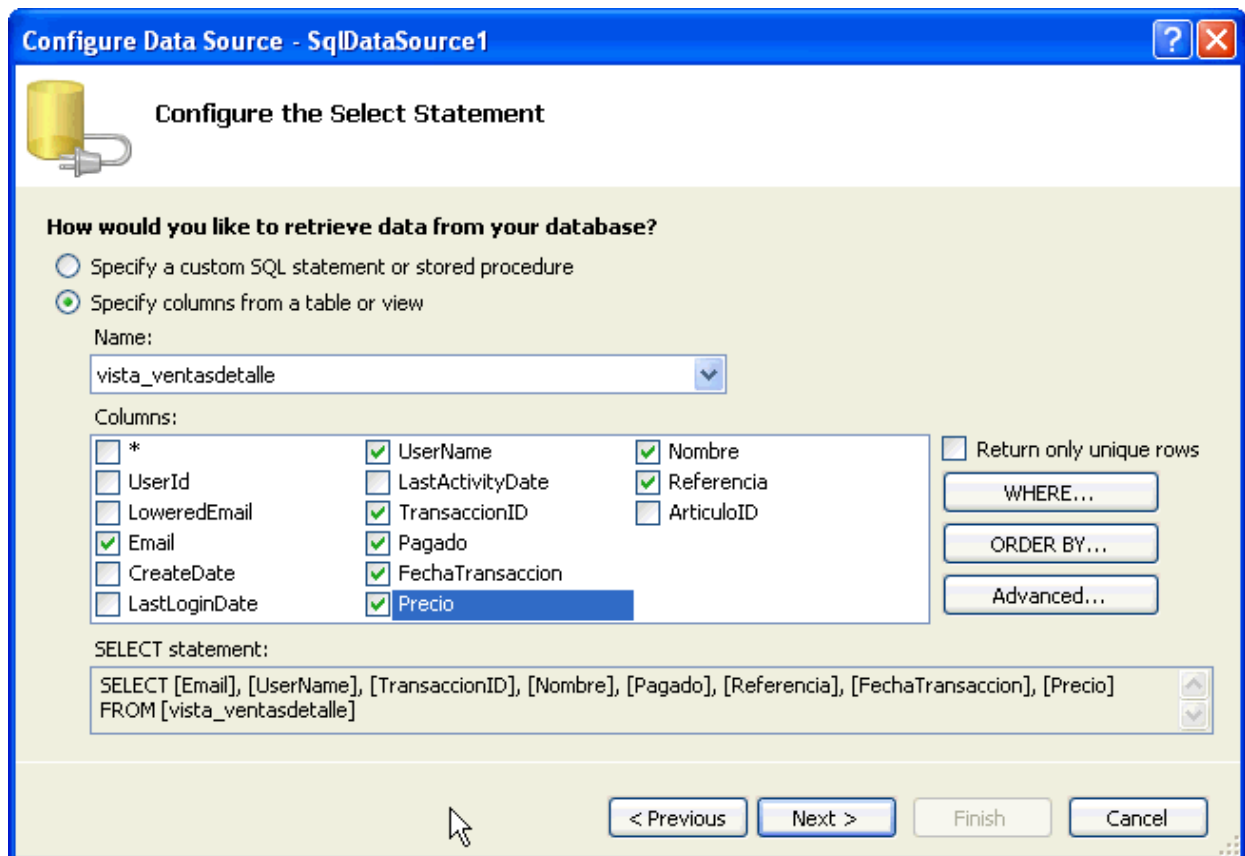


Y ya nos pregunta de ese origen de datos que hemos definido que datos queremos mostrar en el control. Podemos especificar una consulta SQL o la opción mas normal que es utilizar una tabla o vista ya existente. Seleccionamos esta última opción y debe aparecer la vista detallada que creamos en el capítulo anterior:

Perfiles y otros controles ASP.NET



Ahora debemos seleccionar las columnas que queremos mostrar:



Podemos seleccionar "*" que si recuerdas en SQL significa "todas las columnas". Como no es el caso seleccionamos las que ves en la pantalla y debajo va mostrando la consulta SQL resultante. Si queremos ordenar los valores pulsamos en "Order by":

Add ORDER BY Clause

Specify the columns you would like to order by.

Sort by
UserName ☒ Ascending ☐ Descending

Then by
 ☒ Ascending ☐ Descending

Then by
 ☒ Ascending ☐ Descending

SELECT statement:
SELECT [Email], [UserName], [TransaccionID], [Nombre], [Pagado],
[Referencia], [FechaTransaccion], [Precio] FROM [vista_ventasdetalle]
ORDER BY [UserName]

OK Cancel

Como ves en la pantalla le ha añadido la instrucción "Order by" para ordenar el resultado. Puede ser ascendente o descendente e incluir mas ordenaciones, por ejemplo por "ciudades" y luego por "apellidos" que no es el caso pero es un ejemplo de porque ordenar por mas de un criterio.

Pulsamos en "Next":

Configure Data Source - SqlDataSource1

Test Query

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

Email	UserName	TransaccionID	Nombre	Pagado	Referencia	FechaTransacc
jose@miempresa.com	jose	20000	Introducción a Visual Basic	<input checked="" type="checkbox"/>	ARC-001	01/04/2006
jose@miempresa.com	jose	20003	Los For-Terrier	<input checked="" type="checkbox"/>	ARC-002	01/01/2006
jose@miempresa.com	jose	20006	Fotografia	<input checked="" type="checkbox"/>	IMG-001	01/01/2006
antonio@miempresa.com	antonio	20007	Fotografia	<input checked="" type="checkbox"/>	IMG-001	01/01/2006
antonio@miempresa.com	antonio	20008	Fotografia II	<input checked="" type="checkbox"/>	IMG-002	01/01/2006

Test Query

SELECT statement:

```
SELECT [Email], [UserName], [TransaccionID], [Nombre], [Pagado], [Referencia], [FechaTransaccion], [Precio] FROM [vista_ventasdetalle]
```

< Previous Next > Finish Cancel

Para, pulsando "Test Query", ver una vista previa del resultado de la consulta, todo bastante útil para conseguir nuestro resultado. Pulsamos en "Finish"

Mostrar valores únicos

En ocasiones podemos querer recuperar sólo los distintos valores de un campo. Por ejemplo queremos saber los distintos usuarios que hay en esta vista, nos da igual los artículos que hayan comprado, solo queremos saber los valores distintos que haya del campo "username". Esto será muy útil para cuando creamos cuadros desplegables en nuestras páginas para realizar selecciones. Esto lo haremos desde la opción de selección de columnas pero marcando la casilla de verificación de "Return only unique rows":

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: vista_ventasdetalle

Columns:

<input type="checkbox"/> *	<input type="checkbox"/> CreateDate	<input type="checkbox"/> TransaccionII
<input type="checkbox"/> UserId	<input type="checkbox"/> LastLoginDate	<input type="checkbox"/> Pagado
<input type="checkbox"/> LoweredEmail	<input checked="" type="checkbox"/> UserName	<input type="checkbox"/> FechaTransa
<input checked="" type="checkbox"/> Email	<input type="checkbox"/> LastActivityDate	<input type="checkbox"/> Precio

☒ Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:

SELECT DISTINCT [UserName], [Email] FROM [vista_ventasdetalle]

< Previous Next > Finish Cancel

Y el resultado será:

UserName	Email
antonio	antonio@miempresa.com
jose	jose@miempresa.com

Si no marcamos esa opción nos muestra solo esos dos campos de todas las filas:

UserName	Email
jose	jose@miempresa.com
jose	jose@miempresa.com
jose	jose@miempresa.com
antonio	antonio@miempresa.com
antonio	antonio@miempresa.com

Que como eran 5 filas nos muestra las 5 pero solo con las columnas indicadas. Está claro que sólo hay dos valores distintos para los usuarios (Jose y Antonio) así que eso es lo que nos ha mostrado en la consulta anterior.

Recuperar sólo determinadas filas

En ocasiones vamos a querer filtrar el resultado para restringir por ejemplo la consulta a todos los de la ciudad "Logroño", o los usuarios "Jose". Esto lo realiza la consulta "Where" que tiene mas cosas así que vamos a

verla mas detalladamente.

La cláusula SQL que realiza este filtro del resultado es "Where" que la seleccionaremos desde la pantalla de la selección de columnas:

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: vista_ventasdetalle

Columns:

<input type="checkbox"/> *	<input checked="" type="checkbox"/> UserName	<input checked="" type="checkbox"/> Nombre
<input type="checkbox"/> UserId	<input type="checkbox"/> LastActivityDate	<input checked="" type="checkbox"/> Referencia
<input type="checkbox"/> LoweredEmail	<input checked="" type="checkbox"/> TransaccionID	<input type="checkbox"/> ArtículoID
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> Pagado	
<input type="checkbox"/> CreateDate	<input checked="" type="checkbox"/> FechaTransaccion	
<input type="checkbox"/> LastLoginDate	<input checked="" type="checkbox"/> Precio	

☐ Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:

```
SELECT [Email], [UserName], [TransaccionID], [Nombre], [Pagado], [Referencia], [FechaTransaccion], [Precio]
FROM [vista_ventasdetalle]
```

< Previous Next > Finish Cancel

El primer paso es elegir los campos que vamos a querer utilizar de filtro:

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:

Operator:

Source:

SQL Expression:

Value:

Parameter properties

WHERE clause:

SQL Expression	Value

Buttons: Add, Remove, OK, Cancel

Los operadores disponibles son:

Operador	Acción
=	Igual
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
<>	Distinto de

Después de elegir la columna para el filtro y el operador,

Perfiles y otros controles ASP.NET

Column:
UserName

Operator:
=

Source:
None
Control
Cookie
Form
Profile
QueryString
Session

Hay varias opciones pero las mas utilizadas son:

- Control. Elegiremos esta si lo que queremos comparar es el valor de otro control de esta página (muy útil)
- Profile. Elegiremos esta para seleccionar registros para el usuario que está viendo la página, es por tanto, de su perfil ("Profile")

Las otras opciones son interesantes, por ejemplo la comparación con una variable global de tipo "session" o de un parámetro del formulario de una página anterior "querystring", pero las mayoría de las veces serán la anteriores las que utilizaremos.

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:
UserName

Operator:
=

Source:
Profile

SQL Expression:
[UserName] = @UserName

Value:
Profile("UserName")

Parameter properties
Property name:
UserName
Default value:

WHERE clause:

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

Remove

OK Cancel

Debemos añadir el parámetro que ves a la derecha con el mismo nombre que el campo y luego pulsaremos en "Add":

Perfiles y otros controles ASP.NET

WHERE clause:

SQL Expression	Value
[UserName] = @UserName	Profile("UserName")

Que será la lista de filtros o restricciones que le vamos aplicando a esta consulta. Ojo, recuerda que esta columna debe estar seleccionada en la lista de columnas que mostraremos en la consulta. Como en este caso la consulta tiene una variable llamada "UserName", así que al comprobar si funciona la consulta nos preguntará por un valor:

Parameter Values Editor

For each parameter defined in the select statement, specify a type and value.

Parameter	Type	Value
UserName	String	Jose

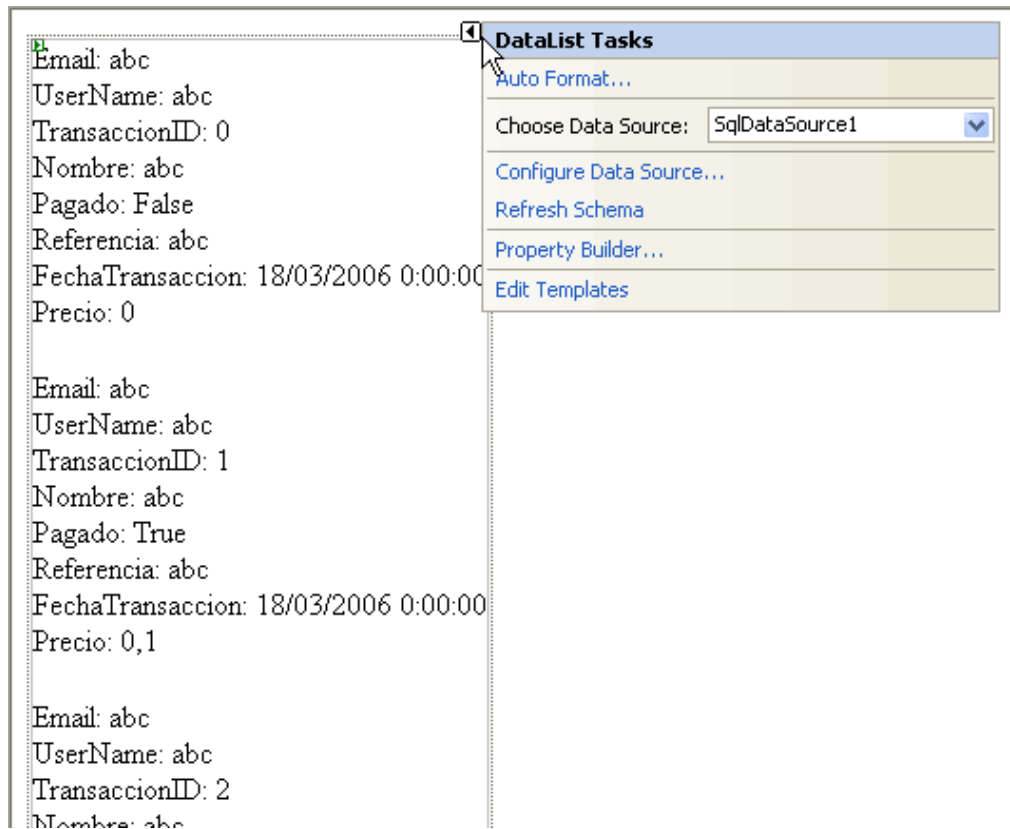
OK Cancel

Es decir, nos va a filtrar por las filas en las que el nombre de usuario "Username" es igual al de un usuario cuyo perfil existe y tiene ese valor en su propiedad. Es un caso un poco especial porque normalmente será una variable libre pero que veas que podemos poner en una consulta que nos restrinja un valor que coincida con el nombre de un perfil de usuario existente.

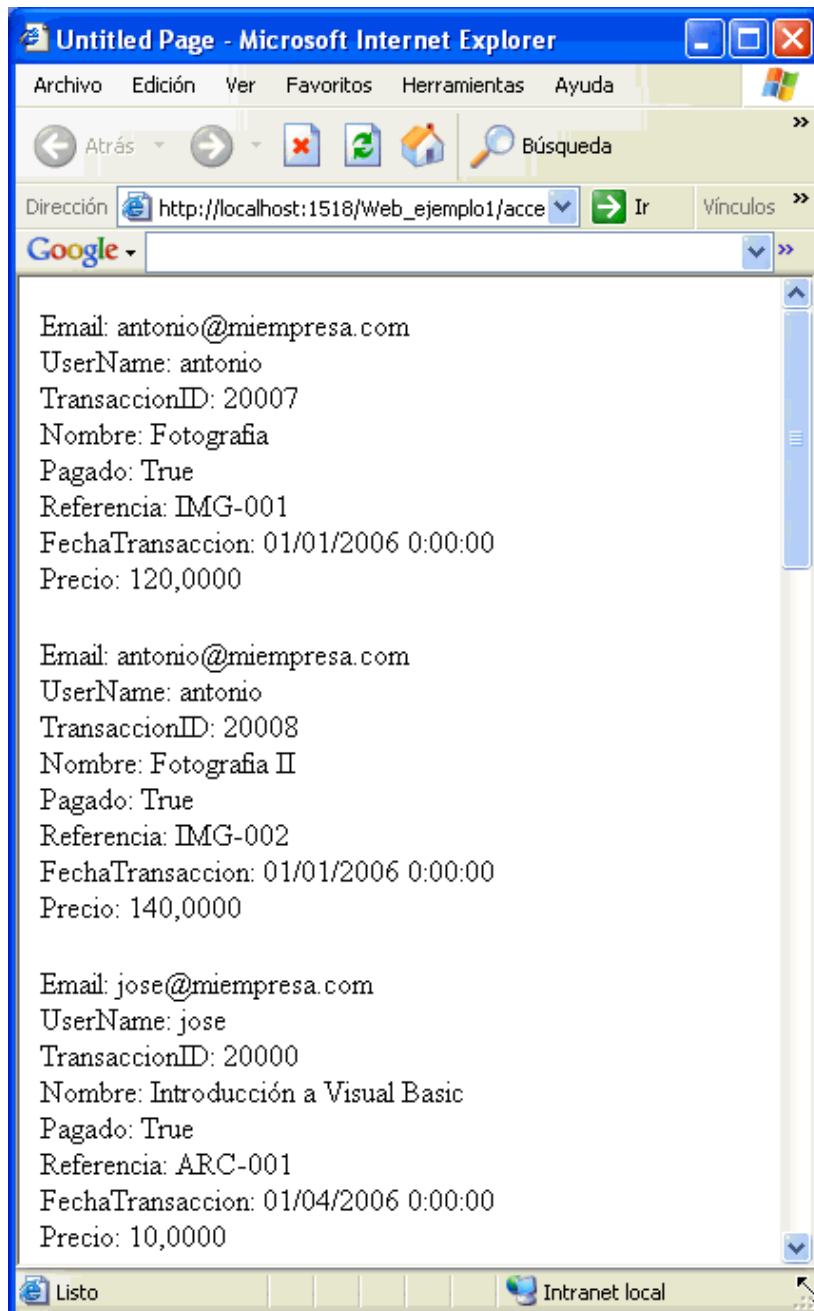
UserName	Referencia	Nombre
jose	ARC-001	Introducción a Visual Basic
jose	ARC-002	Los For-Terrier
jose	IMG-001	Fotografía

Control de datos en la vista de diseño

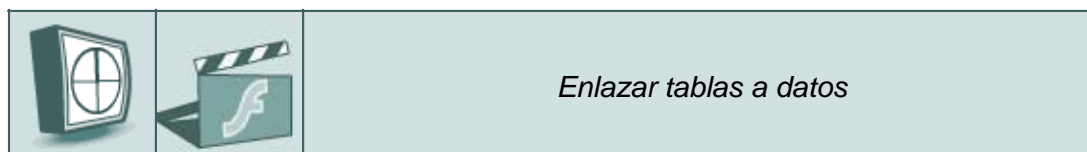
Veamos una consulta pero esta vez lo haremos con un control nuevo: el Datalist, que es muy interesante para mostrar por ejemplo un catálogo. Pon uno y le asocias un origen de datos:



Bueno, el resultado es bastante feo además nos muestra una vista previa de los datos pero sin contenido, como ves pone "abc" en todos los grupos de datos. Si guardamos la página y la vemos en el navegador:



Ahora si que ha funcionado bien, aunque sigue siendo bastante feo..., recuerda que le quitamos las restricciones así que nos muestra todos los registros. Pero ya ves, con un poco de práctica con este asistente de consultas hemos conseguido crear una página dinámica. Tendremos decenas de formas de mostrar la información, ten en cuenta que lo pretendido ahora era conocer el asistente, los controles los veremos enseguida...

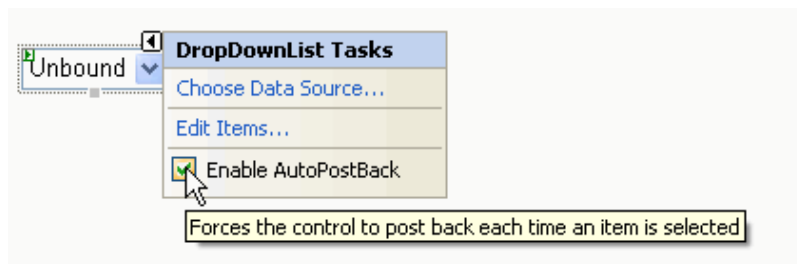


4. Controles desplegables

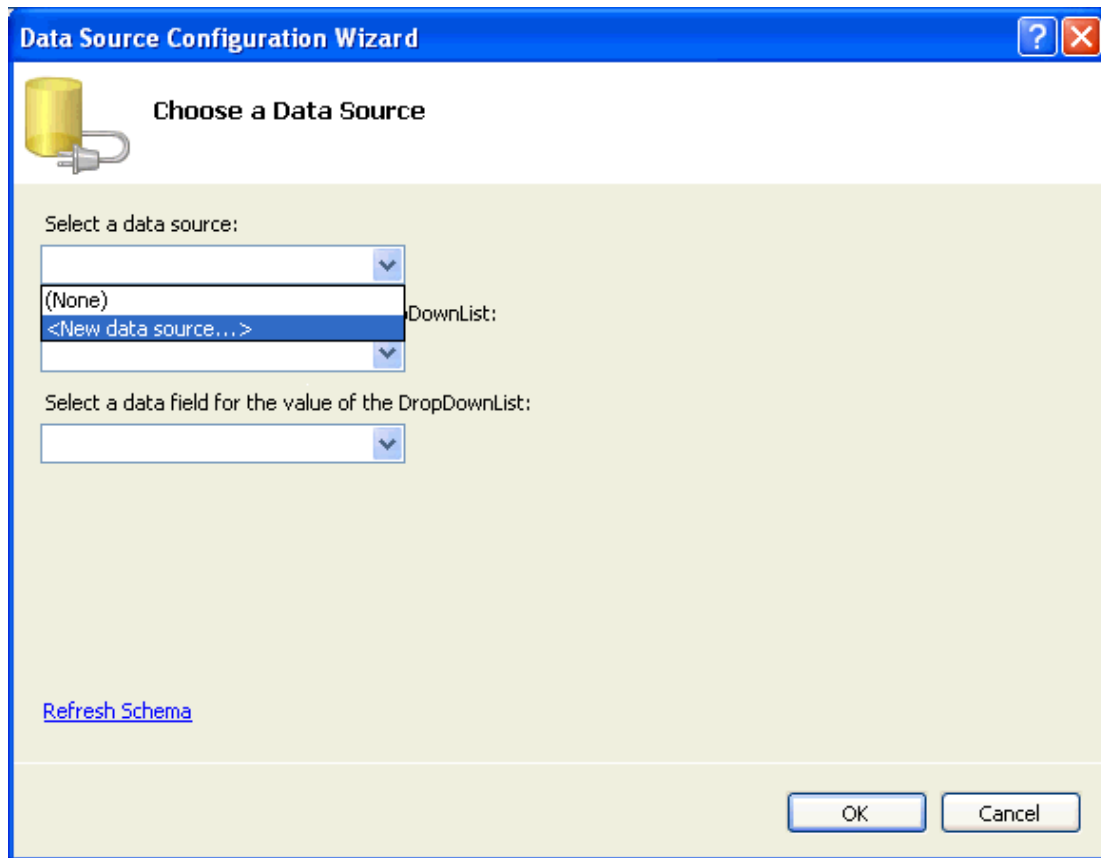
Otro ejemplo de repaso con los controles desplegables...

Los cuadros desplegables también tienen su oportunidad en el enlace a datos, se hecho se utilizan mucho mas de lo que piensas ya que no se rellenan de forma estática sino dinámica con datos provenientes de una consulta a una base de datos. Ejemplo: queremos proporcionar una pantalla de consulta a nuestra base de datos de transacciones y queremos que la consulta sea por las personas que han hecho algún pedido. Podemos entonces rellenar un cuadro de lista con todos los nombre distintos que aparezcan en esas transacciones que si recuerdas ya lo vimos al seleccionar la casilla de verificación de devolver solo las filas únicas. Además cuando seleccionemos una persona: Jose, Antonio, ... en lugar de devolver ese nombre podemos devolver su identificador único así será mas sencillo de consultar en la base de datos se transacciones. Este es un ejemplo muy típico... vamos a hacer un sencillo ejemplo...

Crea una página "desplegable.aspx" y arrastra un control de este tipo "Dropdownlist" (de la lista de controles "Standard"):



Activamos el "AutoPostBack" para que al seleccionar un elemento automáticamente navegue para procesarlo. Ahora selecciona un "Data Source":



Si nuestra página no tiene ningún origen de datos porque sea el primer control debemos crear uno seleccionando "New data source..." y procedemos como ya conocemos, luego en la selección de datos:

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:

Columns:

<input type="checkbox"/> *	<input type="checkbox"/> IsAnonymous	<input checked="" type="checkbox"/> Return only unique rows
<input type="checkbox"/> ApplicationId	<input type="checkbox"/> LastActivityDate	<input type="button" value="WHERE..."/>
<input type="checkbox"/> UserId		<input type="button" value="ORDER BY..."/>
<input checked="" type="checkbox"/> UserName		<input type="button" value="Advanced..."/>
<input type="checkbox"/> LoweredUserName		
<input type="checkbox"/> MobileAlias		

SELECT statement:

< Previous Next > Finish Cancel

He indicado una vez sacar los nombres de usuario distintos de la tabla de usuarios "vw_aspnet_users", indicando "Return only unique rows". Además en el botón "Order by" le hemos puesto que lo ordene por el nombre de usuario, fíjate en la SQL generada en la parte inferior.

El resultado es:

Untitled Page - Mi...

Archivo Edición Ver Fav >>

Atrás > Ir Vínculos >>

Google

antonio
antonio
jose
Josemari

Intranet local

Que como te imaginas es muy útil para realizar filtros en otras consultas.

4.1 Utilizar listas desplegables para filtrar registros.

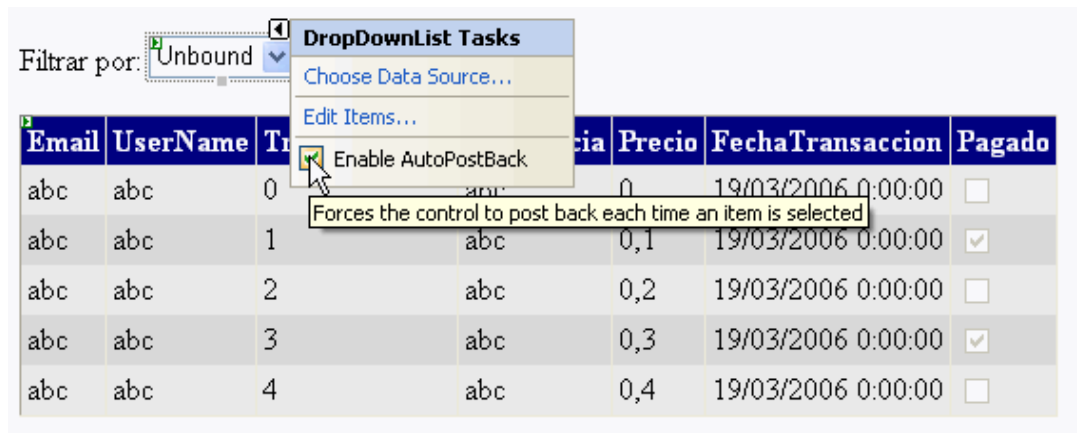
Como acabamos de comentar vamos a ver como utilizamos este control en una página que tenga una cuadrícula para filtrar datos. Vamos a crear una página por ejemplo "consulta.aspx" y ponemos un "GridView" con un origen de datos por ejemplo de la vista detallada de la venta de artículos (vista_ventadetalhada) que creamos ejemplos atrás. Si no la tienes no importa, utiliza una cualquiera. Por ejemplo entonces:

Con lo que tenemos como resultado de momento:

Email	UserName	TransaccionID	Referencia	Precio	FechaTransaccion	Pagado
jose@miempresa.com	jose	20000	ARC-001	10,0000	01/04/2006 0:00:00	<input checked="" type="checkbox"/>
jose@miempresa.com	jose	20003	ARC-002	34,0000	01/01/2006 0:00:00	<input checked="" type="checkbox"/>
jose@miempresa.com	jose	20006	IMG-001	120,0000	01/01/2006 0:00:00	<input checked="" type="checkbox"/>
antonio@miempresa.com	antonio	20007	IMG-001	120,0000	01/01/2006 0:00:00	<input checked="" type="checkbox"/>
antonio@miempresa.com	antonio	20008	IMG-002	140,0000	01/01/2006 0:00:00	<input checked="" type="checkbox"/>

Perfiles y otros controles ASP.NET

Ahora añadimos un control de lista desplegable con el contenido con el que queremos filtrar. Por ejemplo como en el caso anterior creamos un control con los nombres de usuarios:



Creamos un nuevo origen de datos y lo rellenamos con:

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:

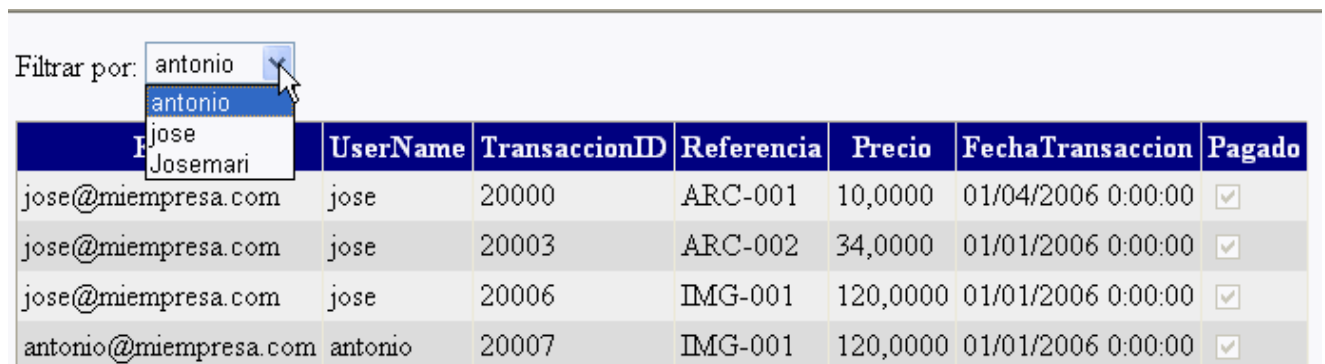
Columns:

<input type="checkbox"/> *	<input type="checkbox"/> IsAnonymous	<input checked="" type="checkbox"/> Return only unique rows <input type="button" value="WHERE..."/> <input type="button" value="ORDER BY..."/> <input type="button" value="Advanced..."/>
<input type="checkbox"/> ApplicationId	<input type="checkbox"/> LastActivityDate	
<input type="checkbox"/> UserId		
<input checked="" type="checkbox"/> UserName		
<input type="checkbox"/> LoweredUserName		
<input type="checkbox"/> MobileAlias		

SELECT statement:

```
SELECT DISTINCT [UserName] FROM [vw_aspnet_Users] ORDER BY [UserName]
```

Probamos los dos controles aunque todavía no están conectados:



Por separado funcionan, ya ves la idea, es que al seleccionar un nombre del desplegable se nos actualice la cuadrícula con los que cumplan ese criterio, que lógicamente será que el "UserName" sea el seleccionado.

Perfiles y otros controles ASP.NET

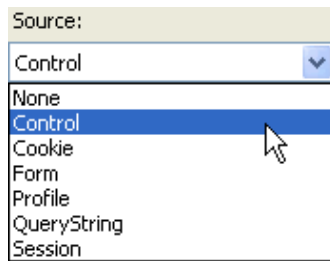
El quiz de la cuestión está en la selección de las filas de la cuadrícula... ¿recuerdas que en las cláusulas "Where" podíamos hacer referencia a valores de la página? Bueno pues si, vamos a ver esta consulta de la cuadrícula en la parte del "Where":

Ahora atento porque es muy fácil. Primero seleccionamos la columna con la que queremos hacer la comparación:

Comparador lógicamente el "=":

El origen de donde vamos a comparar los datos es un control así que lo indicamos:

Perfiles y otros controles ASP.NET



Source:

Control

None

Control

Cookie

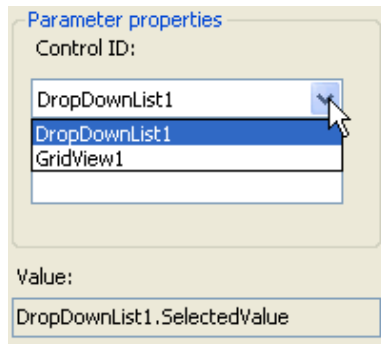
Form

Profile

QueryString

Session

Al seleccionar esta opción se activa la parte derecha arriba donde nos muestra los controles disponibles en la página:



Parameter properties

Control ID:

DropDownList1

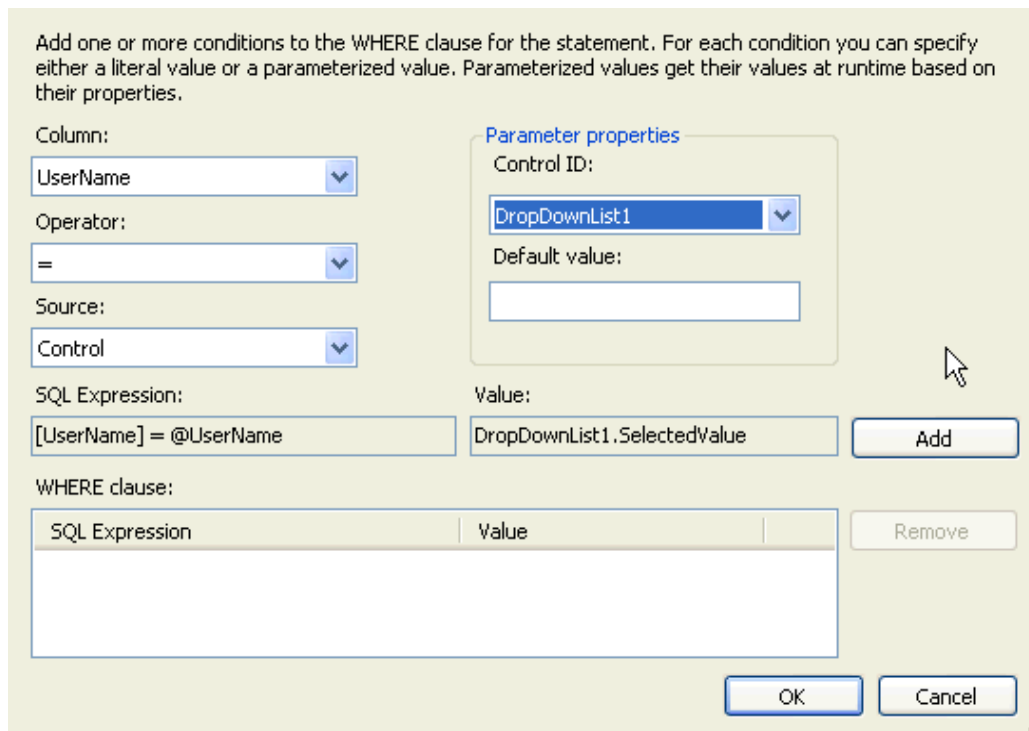
DropDownList1

GridView1

Value:

DropDownList1.SelectedValue

Es decir queremos comparar el campo "UserName" con lo que contenga el control desplegable "DropDownList1". Como ves debajo pone en "Value" que lo comparará con el valor seleccionado en ese desplegable:



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:

UserName

Operator:

=

Source:

Control

SQL Expression:

[UserName] = @UserName

WHERE clause:

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

Remove

Parameter properties

Control ID:

DropDownList1

Default value:

Value:

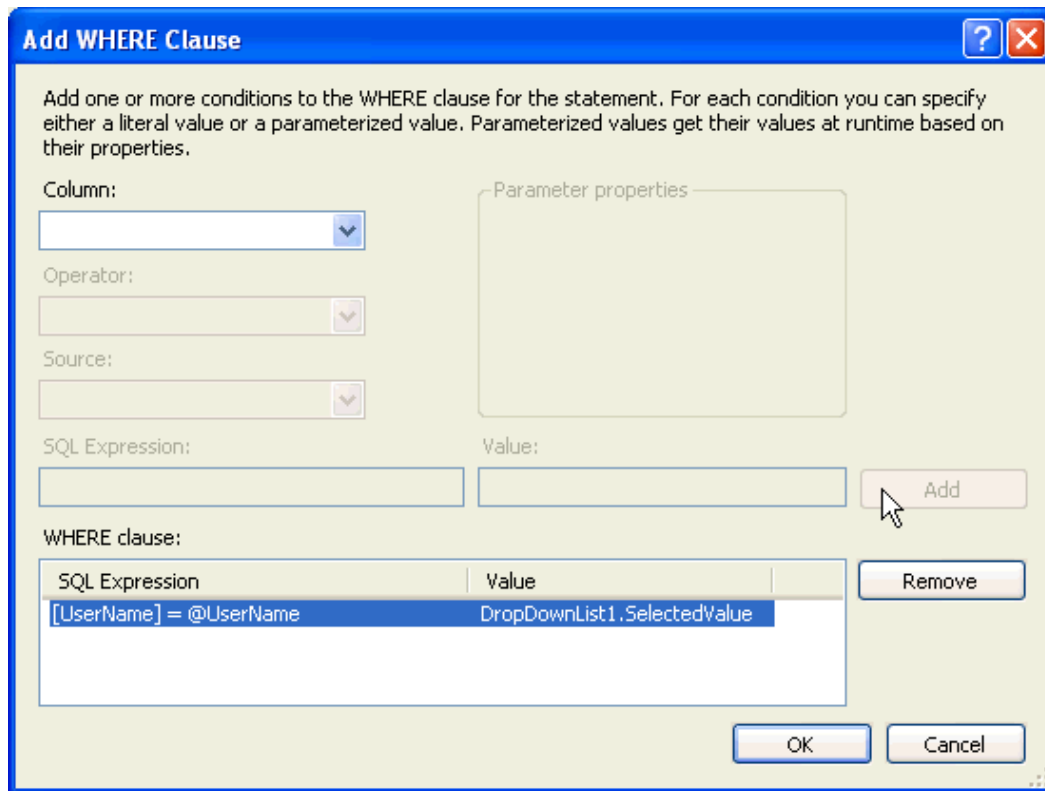
DropDownList1.SelectedValue

Add

OK

Cancel

Finalmente pulsamos en "Add" para añadir este criterio de comparación:



Guardamos estos cambios en la página y la ejecutamos:

Filtrar por:

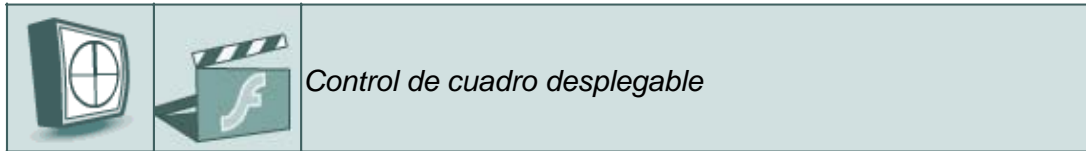
Email	UserName	Transaccion
antonio@miempresa.com	antonio	20007
antonio@miempresa.com	antonio	20008

Y si seleccionamos "Jose":

Filtrar por:

Email	UserName	Transaccion
jose@miempresa.com	jose	20000
jose@miempresa.com	jose	20003
jose@miempresa.com	jose	20006

Así que perfecto, sin escribir ni una sola línea de código hemos creado un par de controles enlazados donde el resultado de la consulta SQL de uno (gridview) depende de la selección de otro control (dropdown)



Ver y editar las propiedades de usuario

Podemos hacer otro ejemplo para ver y editar las propiedades de los perfiles de los usuarios. Recuerda que las propiedades de los perfiles se almacenan en "profile.propiedad". Como novedad podemos leer propiedades de otros usuarios que no sea el activo con:

```
Profile.GetProfile(Usuario).propiedad
```

¿Para que queremos leer otros perfiles? Bueno, si tenemos un desplegable con la lista de usuarios actualizaremos sus propiedades en unos cuadros de texto. Así que para extraer los valores de los perfiles de los usuarios será casi lo mismo que esa instrucción anterior ya que debemos indicar el valor contenido en el cuadro desplegable:

```
Profile.GetProfile(DropDownList1.SelectedValue).propiedad
```

Crea una página que se llame verperfiles.aspx con un desplegable y luego pon unos cuadros de texto con los campos conocidos:

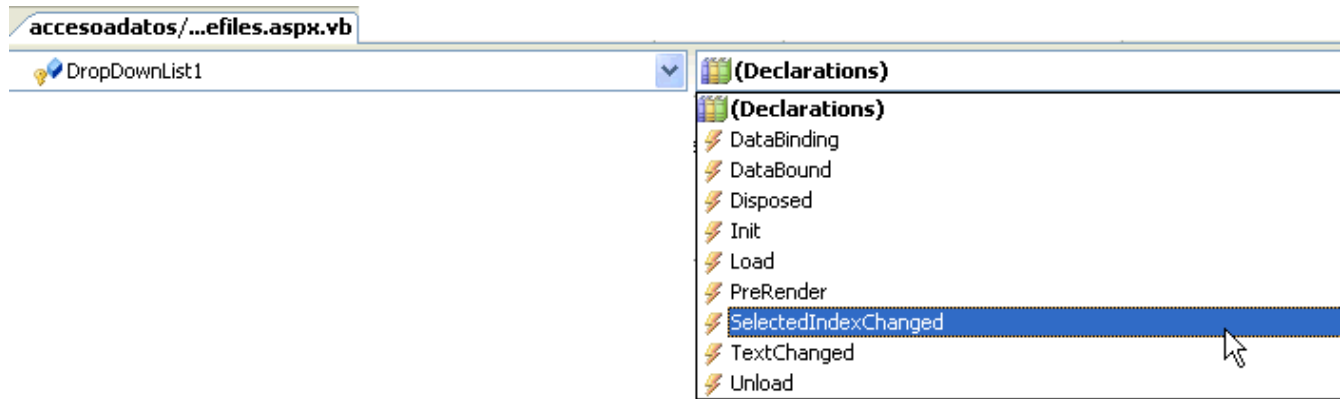
Selecciona un usuario:

Nombre	<input type="text"/>
Apellidos	<input type="text"/>
Dirección 1	<input type="text"/>
Dirección 2	<input type="text"/>
Ciudad	<input type="text"/>
Provincia	<input type="text"/>
Código postal	<input type="text"/>
País	<input type="text"/>
Tema	<input type="text"/>
	<input type="button" value="Enviar"/>

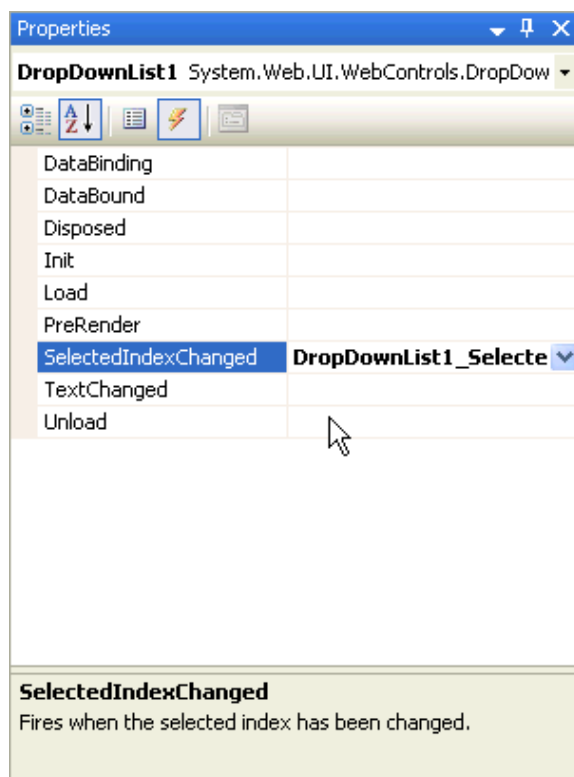
Cada vez que seleccionemos un elemento debemos mostrar el contenido de los campos en el cuadro de texto. Así que vamos a un evento que se dispara cuando seleccionamos un elemento de la lista:

```
DropDownList1.SelectedIndexChanged
```

Te recuerdo las dos formas de llegar a este evento, por el editor (pulsando con el botón derecho en la página y en View code) y seleccionando a la izquierda el control y a la derecha el evento:



O desde la lista de eventos seleccionando el control:



Escribiremos el código necesario para escribir los campos del perfil en las propiedades "Text" de los cuadros de texto:

```
Protected Sub DropDownList1_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)
    txtnombre.Text = Profile.GetProfile(DropDownList1.SelectedValue).UserName
    txtapellidos.Text = Profile.GetProfile(DropDownList1.SelectedValue).Apellidos
    txtdireccion1.Text = Profile.GetProfile(DropDownList1.SelectedValue).Direccion1
    txtdireccion2.Text = Profile.GetProfile(DropDownList1.SelectedValue).Direccion2
    txtciudad.Text = Profile.GetProfile(DropDownList1.SelectedValue).Ciudad
    txtprovincia.Text = Profile.GetProfile(DropDownList1.SelectedValue).Provincia
    txtcodigopostal.Text = Profile.GetProfile(DropDownList1.SelectedValue).CodigoPostal
    txtpais.Text = Profile.GetProfile(DropDownList1.SelectedValue).Pais
    txttema.Text = Profile.GetProfile(DropDownList1.SelectedValue).PreferredTheme
End Sub
```

Ejecuta la página:

Selecciona un usuario: antonio

Nombre: antonio

Apellidos: Lopez Lopez

Dirección 1: Gran via

Dirección 2:

Ciudad: Logroño

Provincia: La Rioja

Código postal: 26006

País: España

Tema: Clasico

Enviar

Y verás como se actualiza el contenido al seleccionar un usuario. La modificación del perfil al pulsar el botón de enviar te la dejo como ejercicio para este tema.

5. Control DataList

Este control es perfecto para mostrar datos como si fuera una lista de elementos en lugar de una tabla con celdas. Veamos un rápido ejemplo para comprobarlo. Crea una nueva página llamada "listaarticulos.aspx" y arrastra un control de este tipo. Luego en el origen de datos ponemos la lista de nuestros artículos:

Perfiles y otros controles ASP.NET

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:

Columns:

<input type="checkbox"/> *	<input type="checkbox"/> Return only unique rows
<input type="checkbox"/> ArtículoID	<input type="button" value="WHERE..."/>
<input checked="" type="checkbox"/> Referencia	<input type="button" value="ORDER BY..."/>
<input checked="" type="checkbox"/> Nombre	<input type="button" value="Advanced..."/>
<input checked="" type="checkbox"/> Descripción	
<input checked="" type="checkbox"/> Precio	

SELECT statement:

Ponle un autoformato y lo vemos en el navegador:

Referencia: ARC-001
Nombre: Introducción a Visual Basic
Descripción: Libro que describe...
Precio: 10,0000

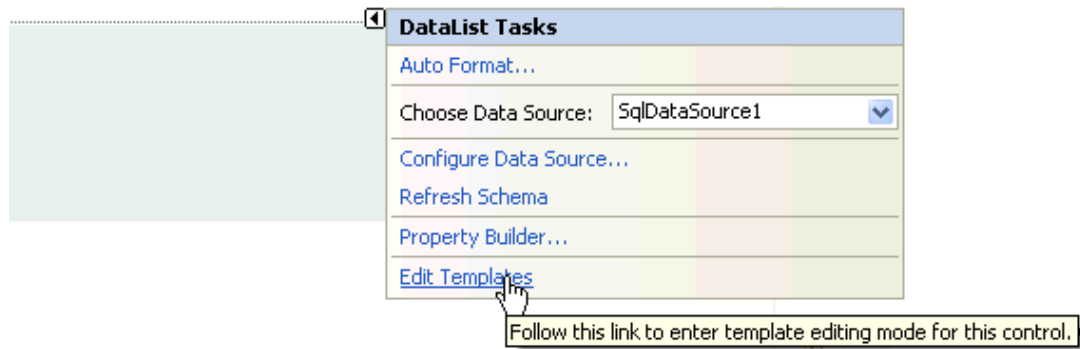
Referencia: ARC-002
Nombre: Los For-Terrier
Descripción: Descripción de la raza...
Precio: 34,0000

Referencia: IMG-001
Nombre: Fotografía
Descripción: Manual de fotografía
Precio: 120,0000

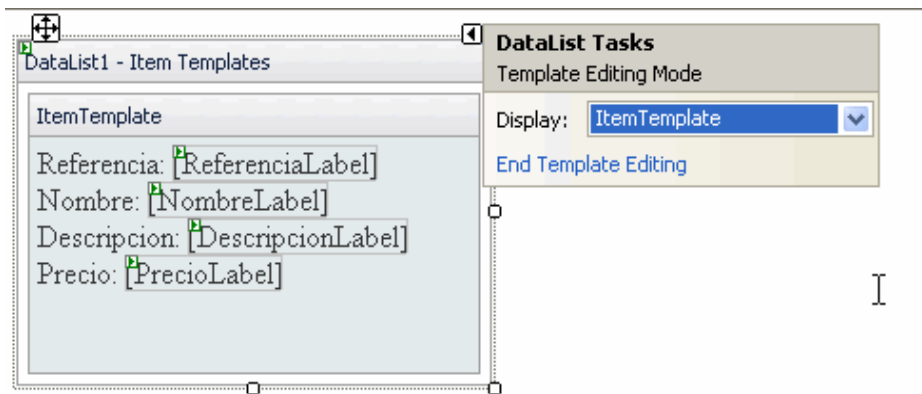
Referencia: IMG-002
Nombre: Fotografía II
Descripción: Manual de fotografía

Lo mas interesante de este control es la capacidad que tiene para dar formato al contenido, ya que tiene mas opciones que los demás. Esto se hace editando las plantillas "templates" que hemos visto que tenían todos estos controles pero hasta ahora no nos podíamos a modificar. De las opciones del control, selecciona "Edit Templates":

Perfiles y otros controles ASP.NET



Que nos muestra:

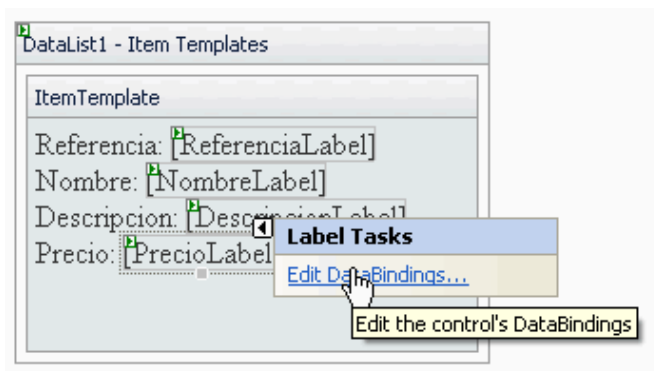


Como ves es la lista de campos que luego nos ha mostrado en la página Web. Podemos modificar y dar formato el texto que describe los campos pero el que está entre los corchetes [] son los identificadores de los campos de la base de datos. Para modificar el aspecto de estas variables podemos utilizar la venta de propiedades y con el botón derecho la tradicional opción de estilos.

Puedes colocar controles en la misma fila teniendo cuidado de que pueden crecer en anchura y solaparse, así que por ejemplo se pueden poner los de fechas, valores numéricos, booleanos, ...

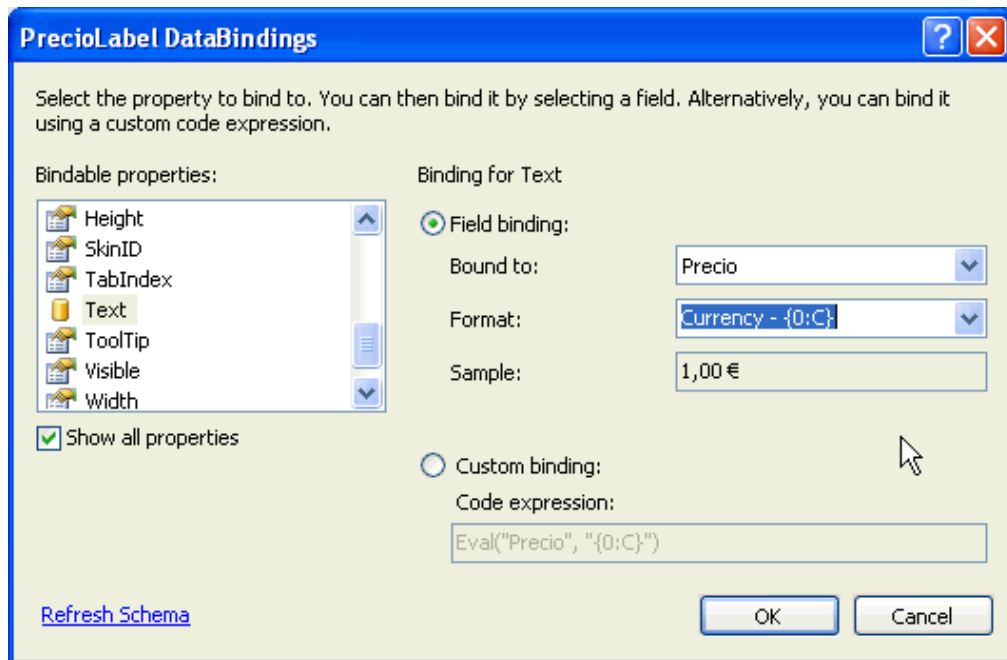
5.1 Dar formato al control

Además de los formatos visuales y de posición en pantalla tenemos la posibilidad de darles un formato especial según el contenido, esto es:



Perfiles y otros controles ASP.NET

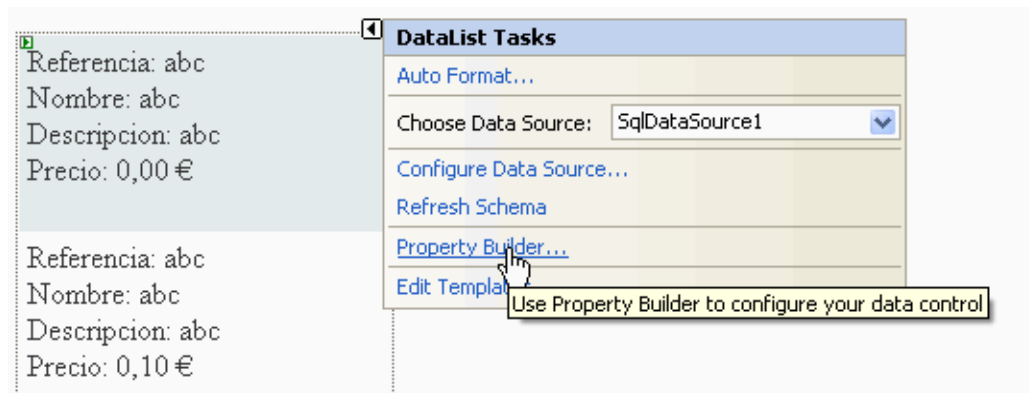
En cada uno de ellos tenemos esa opción que nos muestra:



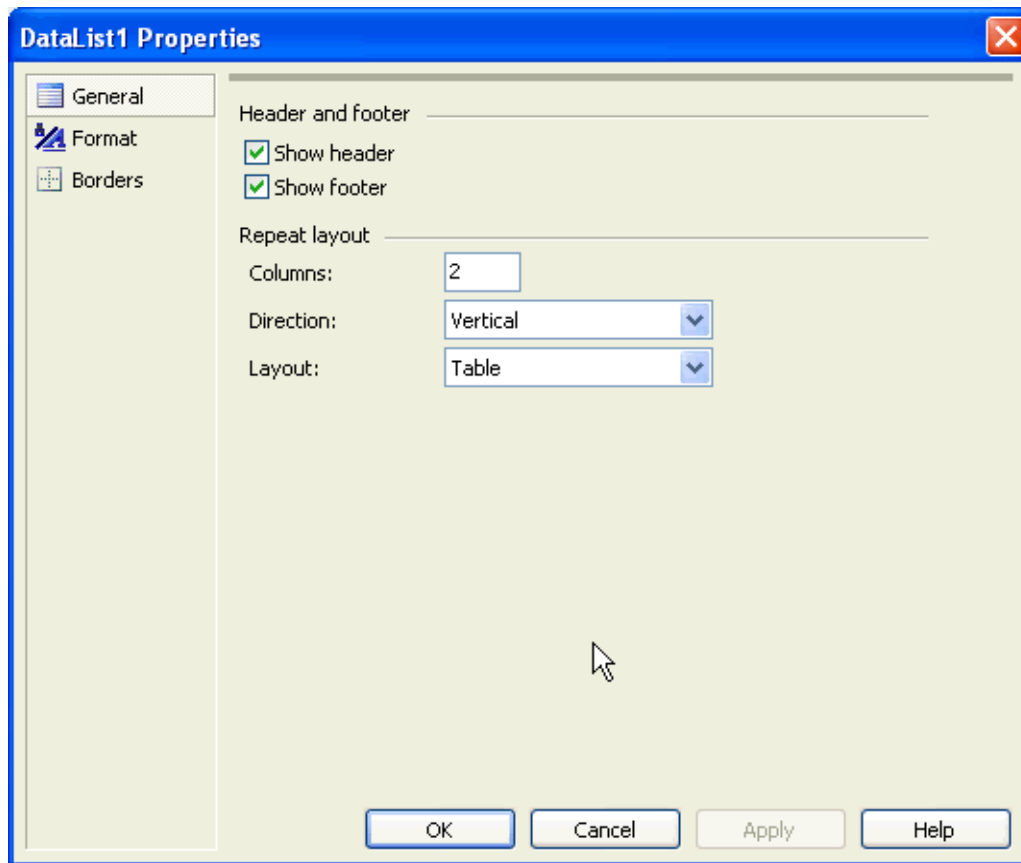
Todas las posibilidades para dar formato a esos valores. Por ejemplo en el caso del precio he puesto el formato "{0:C}" que es el especial para mostrar lo que pone debajo de "Sample" con el símbolo de €. Y como ves con ilimitadas posibilidades...

Mostrar los datos como columnas

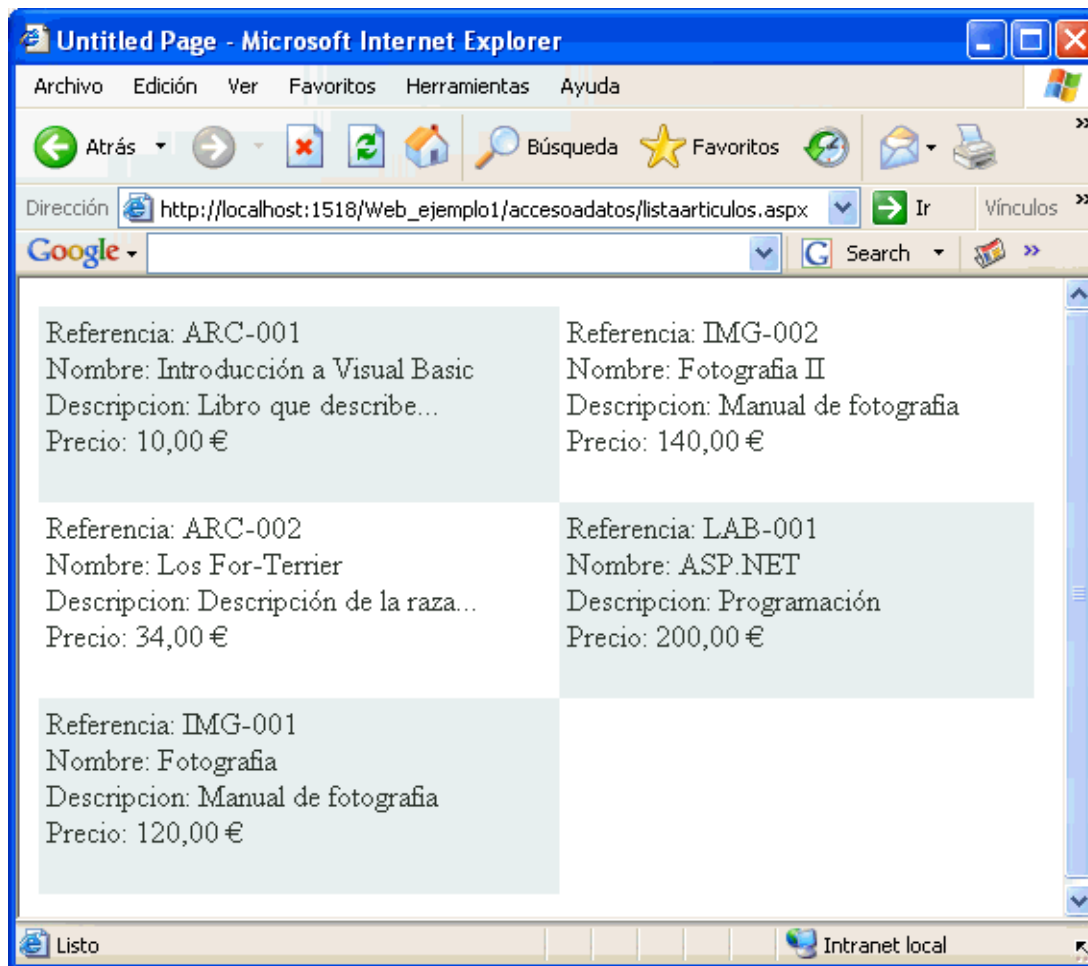
Para mostrar los datos como columnas editamos las plantillas seleccionamos "End edition" para cerrar la edición anterior. En el control seleccionamos ahora "Property Builder":



Que nos muestra:

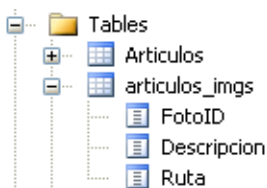


Nuevamente nos encontramos con multitud de opciones para configurar el aspecto de la salida. Por ejemplo el número de columnas, la dirección, ... si elegimos horizontal el primer registro aparece a la izquierda, el segundo a la derecha en la primera fila, ... En "layout" debemos poner en tipo "Table" sino no funcionará el encolumnado, el otro formato "Flow" se utiliza precisamente para los demás casos que no se utilizan tablas.



5.2 Utilizar los controles DataList para mostrar imágenes.

Supongamos una tabla donde tenemos un campo con el nombre de una imagen asociada al registro, por ejemplo, de un catálogo de artículos. Vamos a ver como podemos mostrarlos con este control. Los campos son:



Un identificador del registro, una descripción y por último la ruta de almacenamiento de la base de datos. Vamos a crear una página llamada "catalogo_fotos.aspx" con un control de este tipo, editamos el origen de datos y buscamos nuestra tabla de "articulos_imgs":

Que de momento nos muestra:

```

Description: abc
Ruta: abc

Description: abc
Ruta: abc

Description: abc
Ruta: abc

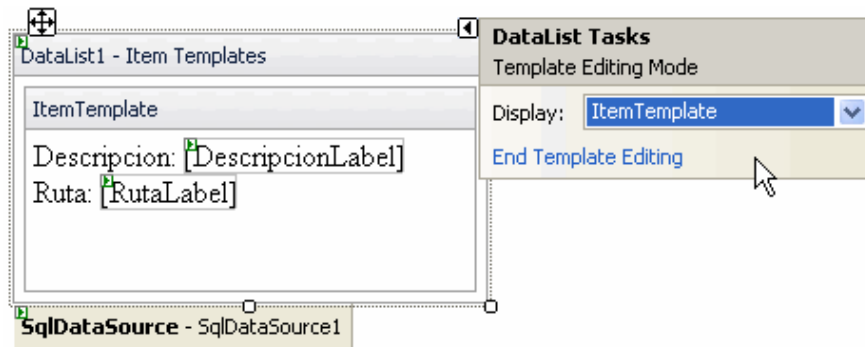
Description: abc
Ruta: abc

Description: abc
Ruta: abc
    
```

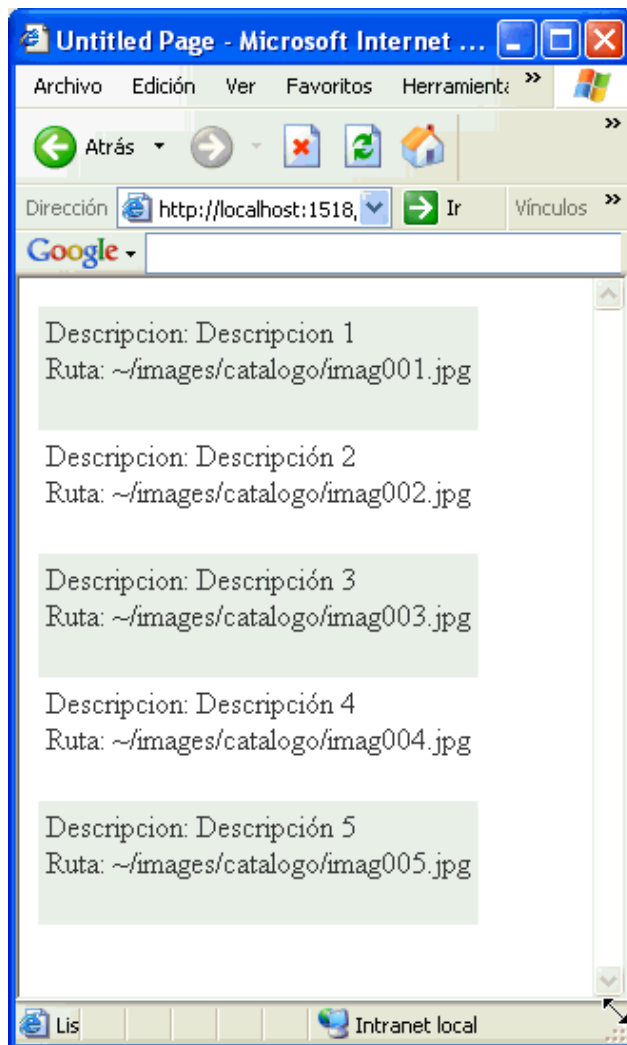
SqlDataSource - SqlDataSource1

Le ponemos un formato decente con el autoformato y editar la plantilla del elemento "Item Template":

Perfiles y otros controles ASP.NET

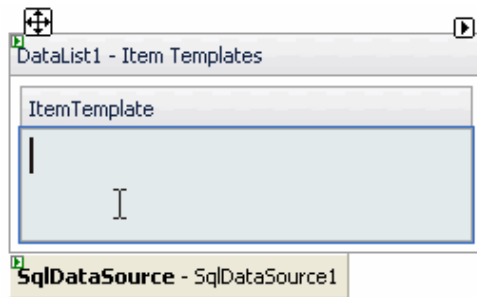


Ya sabes, estas plantillas son como pequeñas páginas web que se van a repetir en cada registro, de hecho de momento si la ejecutamos nos muestra:

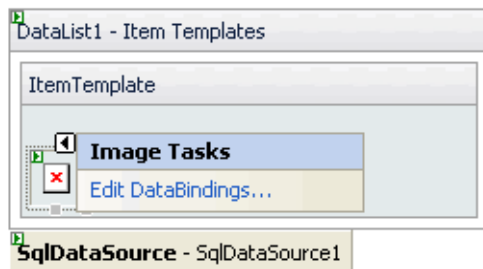


Ahora vamos a prepararla para mostrar las imágenes, parece que en principio debemos poner un control de tipo "imagen" y la ruta que sea la variable del campo "ruta". Vamos a borrar todo el contenido para dejar el control vacío:

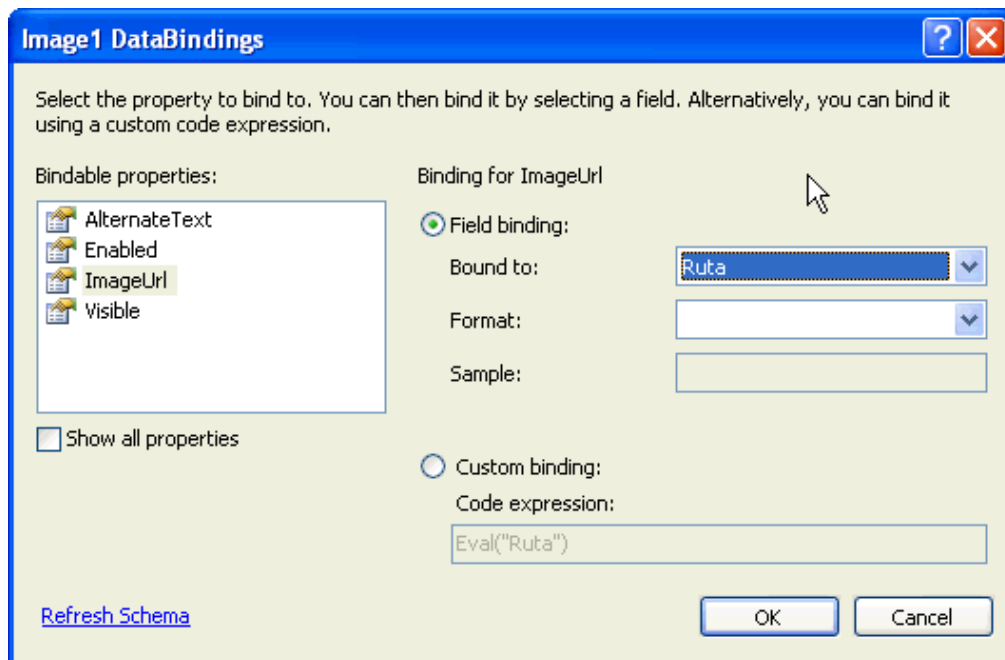
Perfiles y otros controles ASP.NET



Ahora desde los controles estándar de la izquierda arrastramos un control de tipo "Image" dentro de esta plantilla:



Ahora editamos el control con "Edit DataBindings" para editar el enlace a datos:



Lo que queremos mostrar es una imagen de una URL así que eso es lo que indicamos en la parte izquierda. Ya ves que esto es lo que podemos enlazar en este control de imagen. Ahora en la parte derecha le indicamos que queremos un "Field binding" es decir, enlazarlo con un campo de una tabla. Está claro que está perfectamente preparado este control para enlazarse con una base de datos, ya que por defecto es lo que me está preguntando... Ahora simplemente le decimos que queremos enlazarlo con el campo de la URL como ves en la pantalla anterior.

Perfiles y otros controles ASP.NET

Bueno, y esto es todo, la verdad es que no se puede negar que es sencillo. Este editor es verdaderamente potente, ya se que lo he dicho muchas veces pero los que yenimos del mundo de ASP de los primeros tiempos nos cuesta adaptarnos a semejante sencillez. Probemos ahora la página:

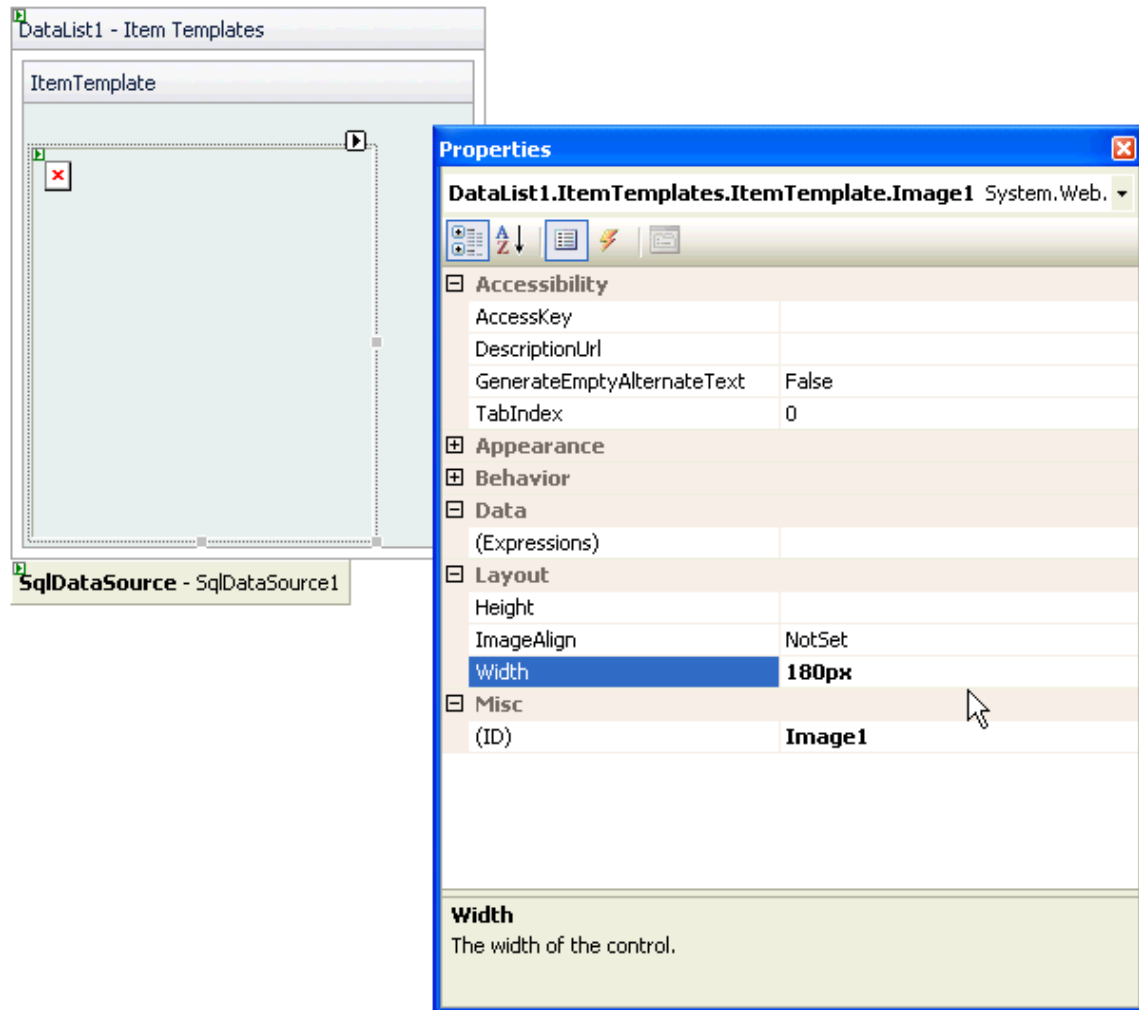


Ya tenemos nuestro sencillo catálogo. Está claro que podíamos haber puesto el título, unos cuadros y algún adorno mas pero de momento lo yemos en plan sencillo.

Tamaño de las imágenes

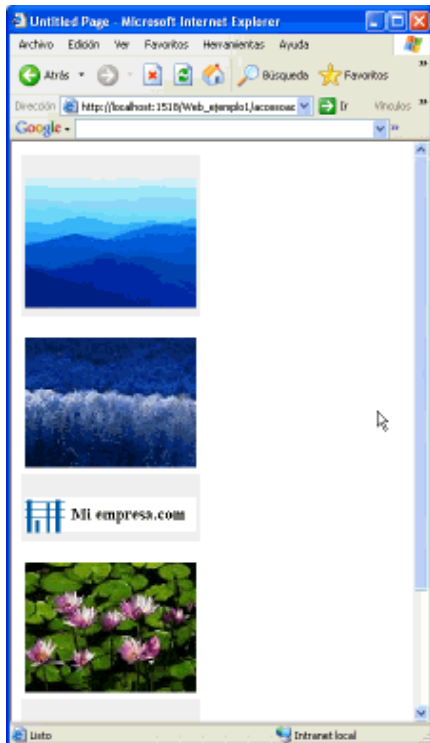
Si queremos poner un mismo tamaño a todas las imágenes podemos hacerlo poniendo alguna de las propiedades del control, por ejemplo una anchura de 180 pixeles a todas las imágenes:

Perfiles y otros controles ASP.NET



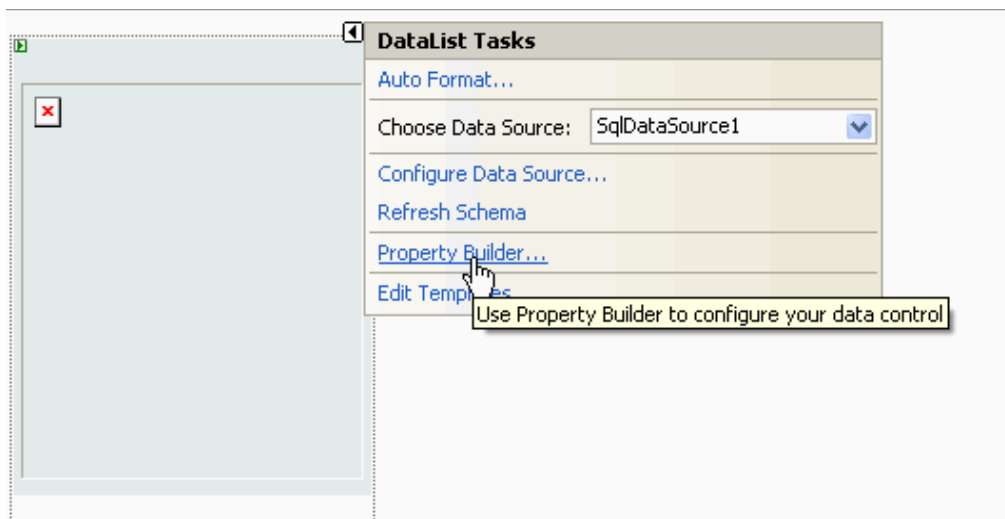
Produce el siguiente efecto:

Perfiles y otros controles ASP.NET

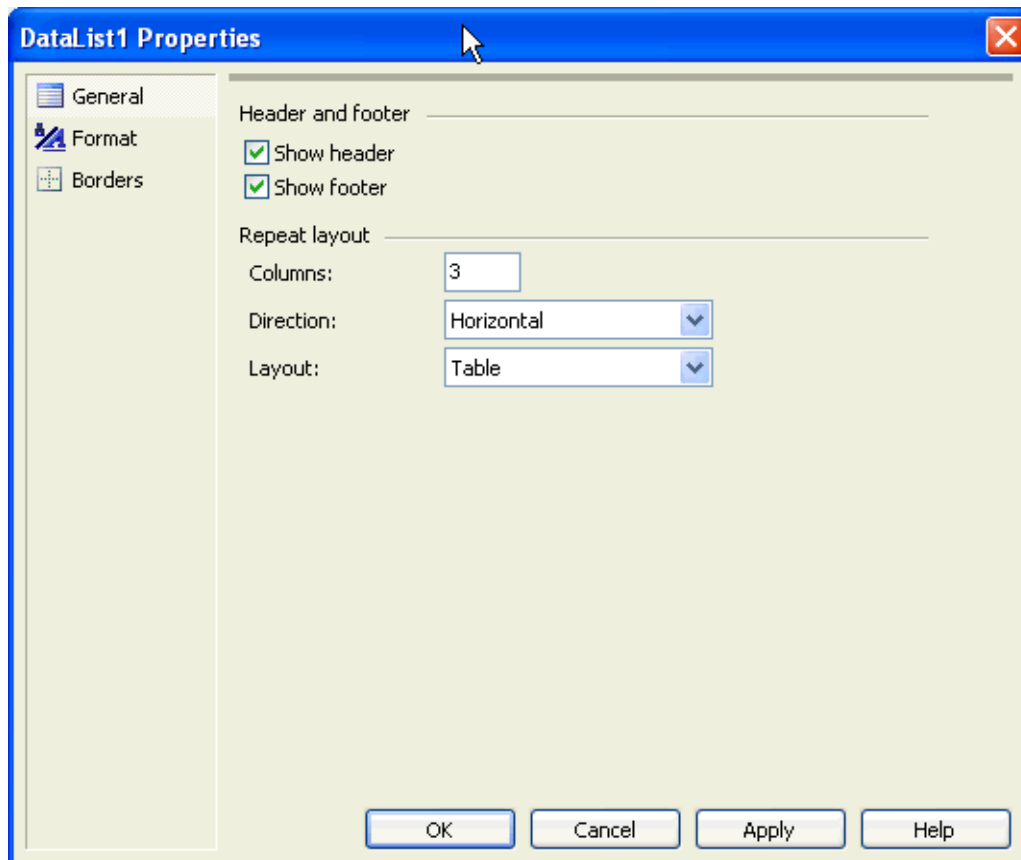


Mostrar imágenes en columnas

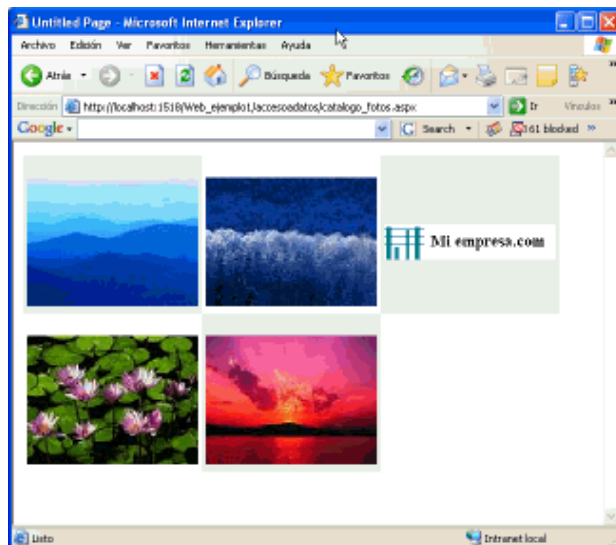
Sabemos que este control puede mostrar controles en columnas así que salimos de la edición de la plantilla y nos vamos a "Property Builder":



Y le indicamos por ejemplo 3 columnas y el orden horizontalmente:



Que produce...

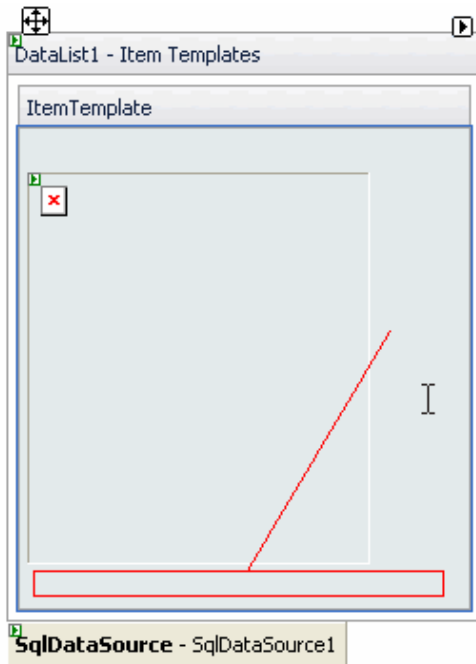


Por supuesto están yigentes todas las propiedades del mundo para las imágenes: ponerles bordes, centrar en las celdas...

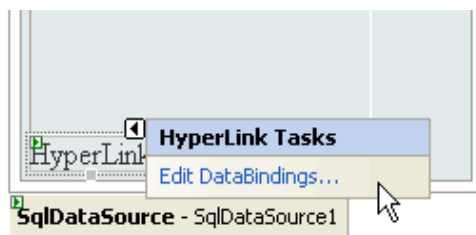
Añadir texto a las imágenes

Vamos a mostrar el título a cada imagen pero con algún detalle interesante, por ejemplo que esos títulos sean hipervínculos y que lleven a una página donde mostremos la imagen a tamaño completo. Como cualquier buen catálogo de imágenes en Internet.

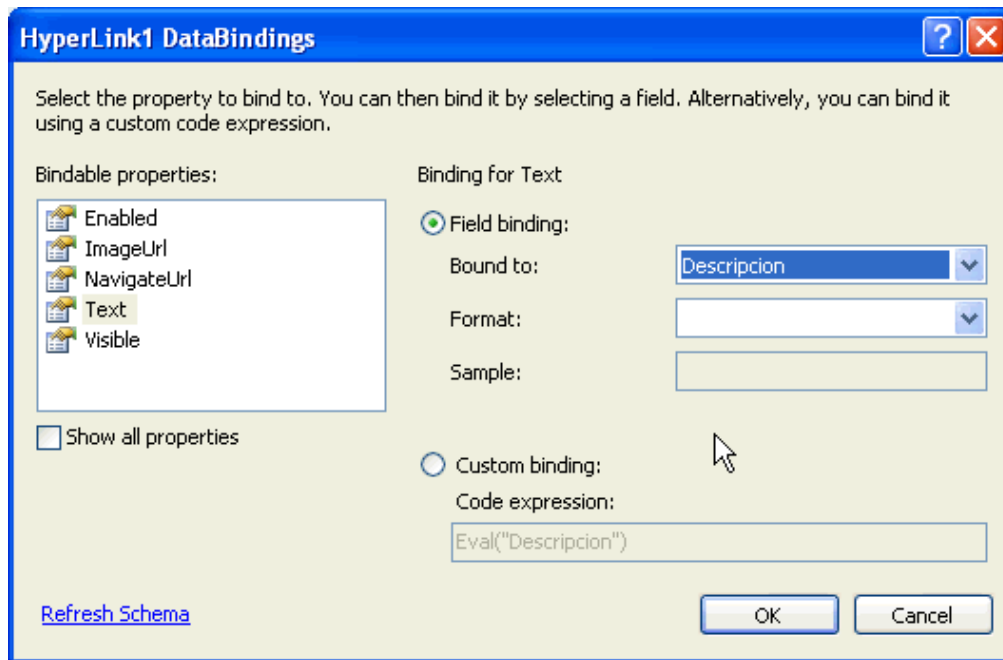
Esto es sencillo, vamos a añadir un control de tipo hipervínculo "Hyperlink" a nuestro control DataList. Así es lo fácil que es añadir más controles y enlazarlos a datos. El texto de ese enlace será por supuesto el título de la fotografía de nuestra base de datos y el enlace en sí será el campo de la "URL". Editamos la plantilla otra vez y pulsamos un "intro" debajo de la imagen para hacer sitio al control:



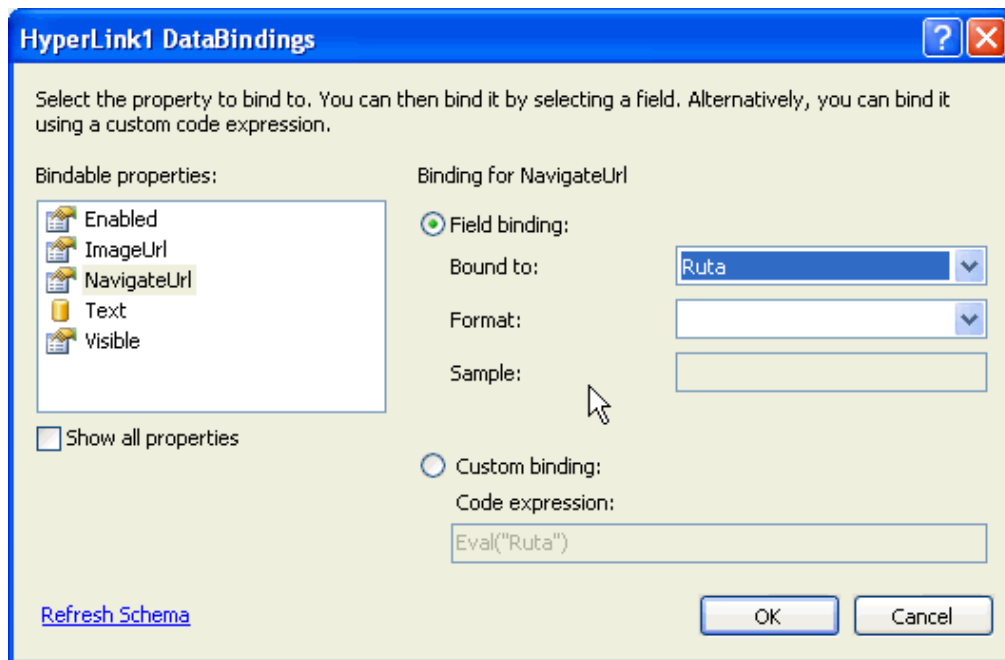
Del grupo de controles estándar, arrastramos el de tipo "Hyperlink":



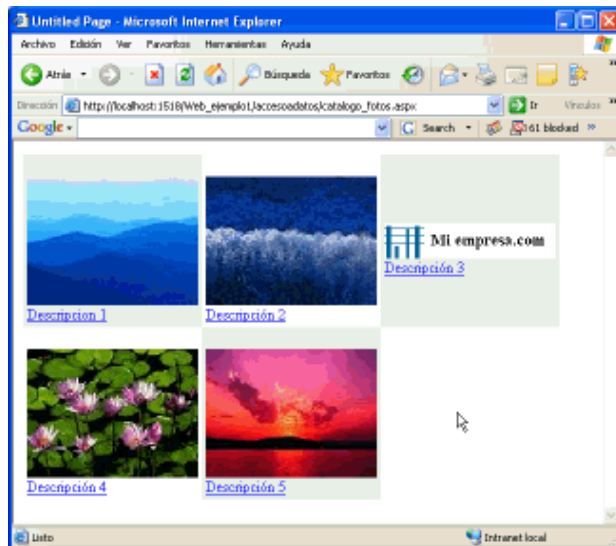
Ahora editamos sus propiedades con "Edit DataBindings", esta claro que como está contenido ya dentro de un control "DataList" que está enlazado a datos sabe con que conexión y tabla estamos trabajando, igual que nos pasó antes con el de la imagen:



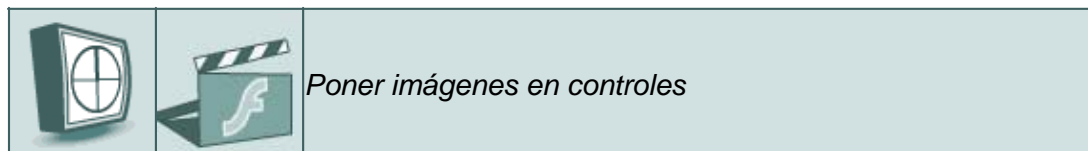
El texto lo tenemos claro, es el que yes en el ejemplo, ahora yiene lo bueno, hacemos clic a la izquierda en "NayigateUrl":



Y le indicamos que debe saltar a la imagen. Grabamos y el resultado es:



Y si hacemos clic en cada descripción abriremos la página con la imagen a tamaño completo porque no está contenida en un control que le fuerza a tener una anchura de 180 pixeles como pusimos al diseñar este control de imagen.

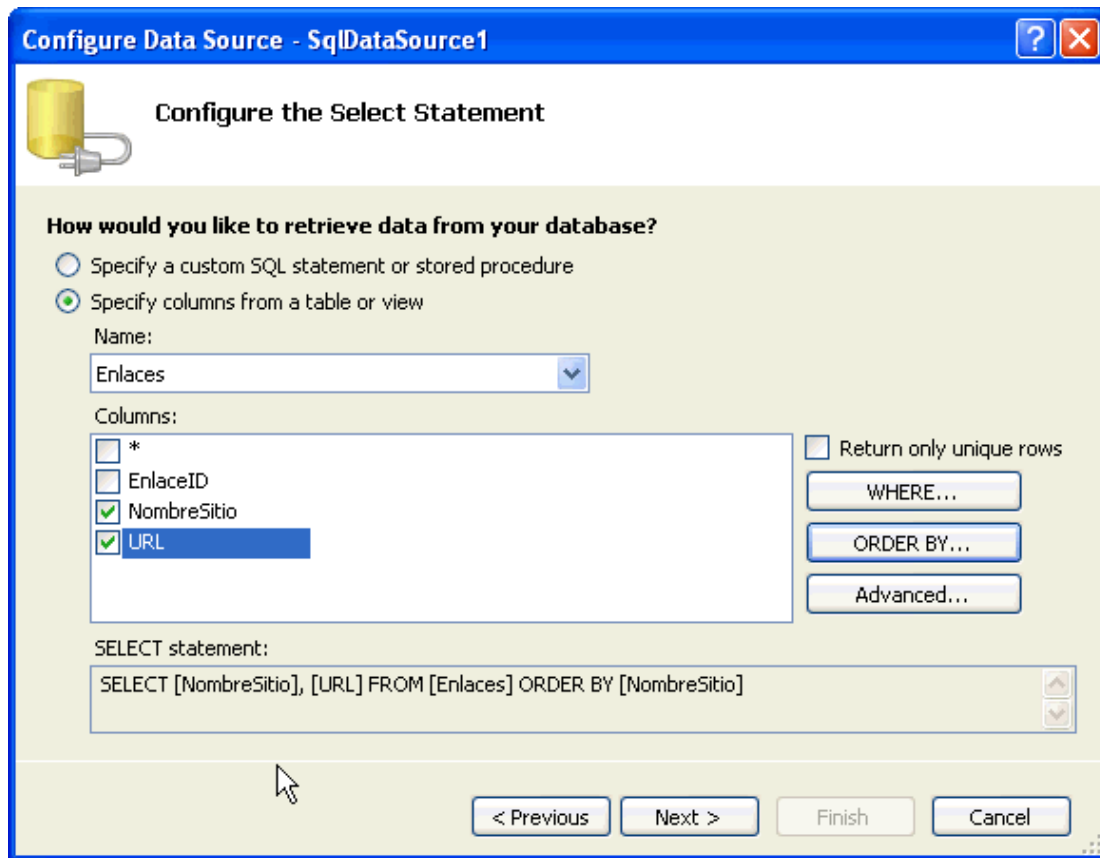


5.3 Utilizar un DataList para mostrar hiperenlaces

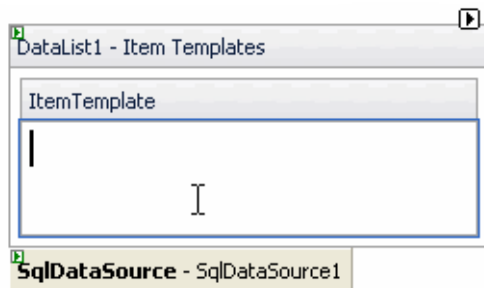
En el ejemplo anterior hemos utilizado por primera vez los hiperenlaces en la plantilla de un control de tipo DataList. Podemos hacer lo mismo para mostrar la lista de hiperynculos de una tabla de una base de datos. Si recuerdas, temas atrás creamos una tabla con sitios interesantes para poner en nuestra web:

	EnlaceID	NombreSitio	URL
▶	1	ASP.NET	http://www.asp.net
	2	Periodico El Mundo	http://www.elmundo.es
	3	Periodico EL Pais	http://www.elpais.es
	4	Microsoft	http://www.microsoft.com
*	NULL	NULL	NULL

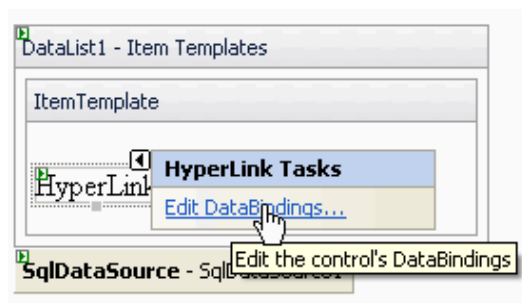
Vamos a crear una página que se llame "enlaces.aspx" con un control Datalist. Edita la conexión de datos para seleccionar nuestra tabla de enlaces:



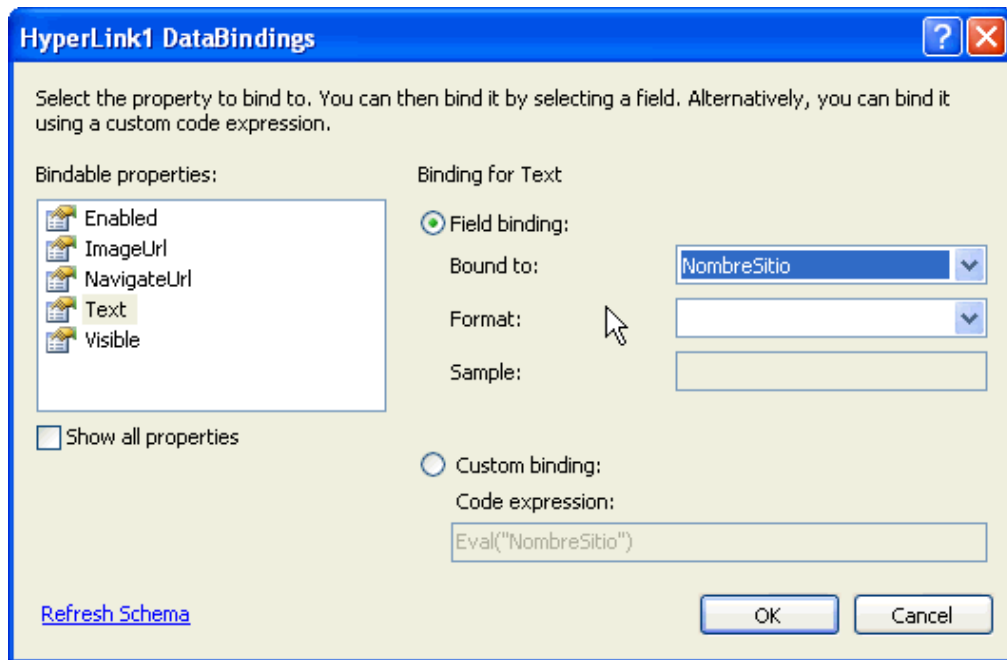
Vamos a mostrar los textos que aparecen en "Nombresitio" e internamente los enlaces serán lógicamente a "URL". Editamos la plantilla y borramos el contenido porque no queremos mostrar esta información:



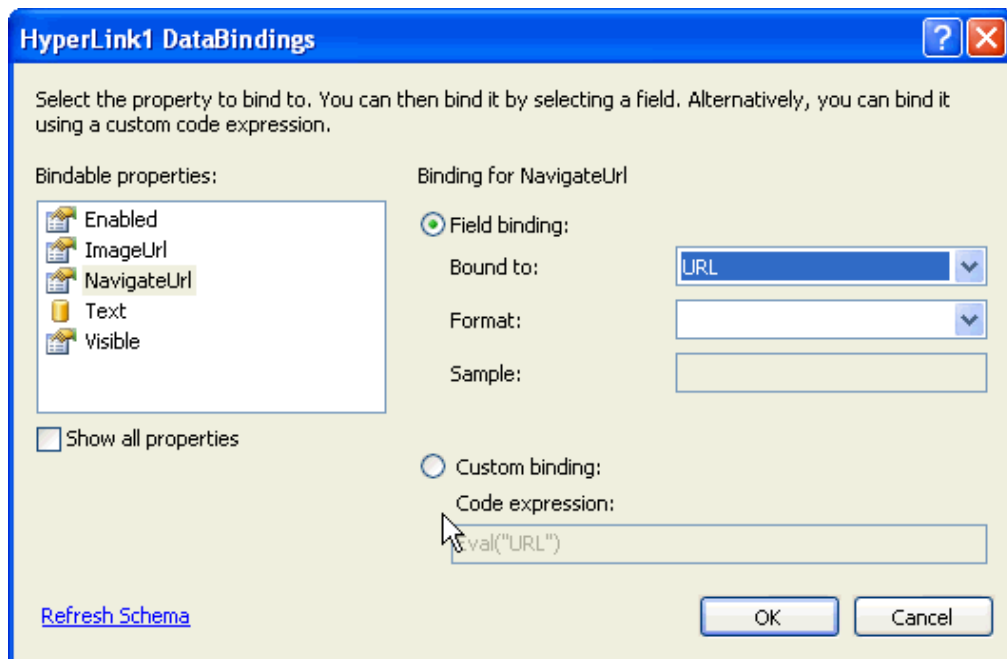
Añadimos otra vez un control de tipo "Hyperlink" y editamos su enlace a datos:



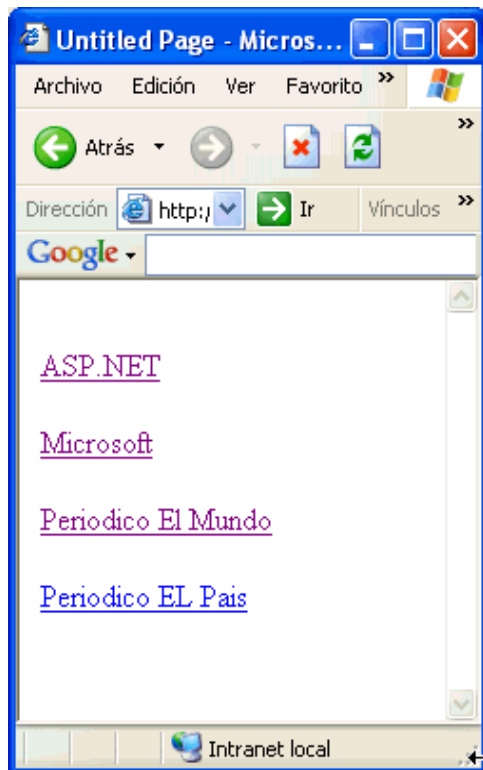
En la parte del texto:



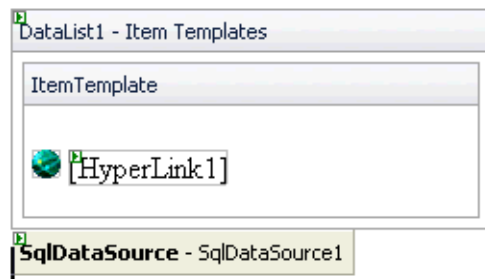
Y lo mismo para el enlace:



Grabamos el ejemplo y lo yemos en el explorador.



Como nuestra plantilla es una pequeña página web podemos poner un gráfico para que se repita y así quedar estéticamente más interesante:



Para producir:



6. El control FormView

Este control es similar al "DetailsView" ya que muestra un solo registro a la vez, y también se parece al "DataList" ya que permite poner los controles en forma de tabla o formulario. Si queremos mostrar un registro para poder editarlo debemos utilizar el ya conocido de "DetailsView" ya que es mas sencillo de utilizar, y veremos enseguida la finalidad de este.

El control FormView funciona de forma similar a los otros controles enlazados a datos, los arrastramos a nuestra página de diseño y los enlazamos con una tabla o vista. Como está limitado a mostrar un solo registro a la vez su uso se reduce a mostrar por ejemplo sumas o totales.

Anteriormente creamos una vista llamada "vista_yentasdetalle" que contenía información detallada de las transacciones. Queremos ahora realizar una suma de los precios de los artículos para ver el importe total de las transacciones. Comenzamos creando una página "calculos.aspx", arrastramos un control de este tipo y editamos el origen de datos:

En este caso no vamos a mostrar una tabla o consulta ya que debemos realizar una operación, así que seleccionamos:

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: Articulos

Columns:

- ☐ *
- ☐ ArticuloID
- ☐ Referencia
- ☐ Nombre
- ☐ Descripcion
- ☐ Precio

☐ Return only unique rows

WHERE...

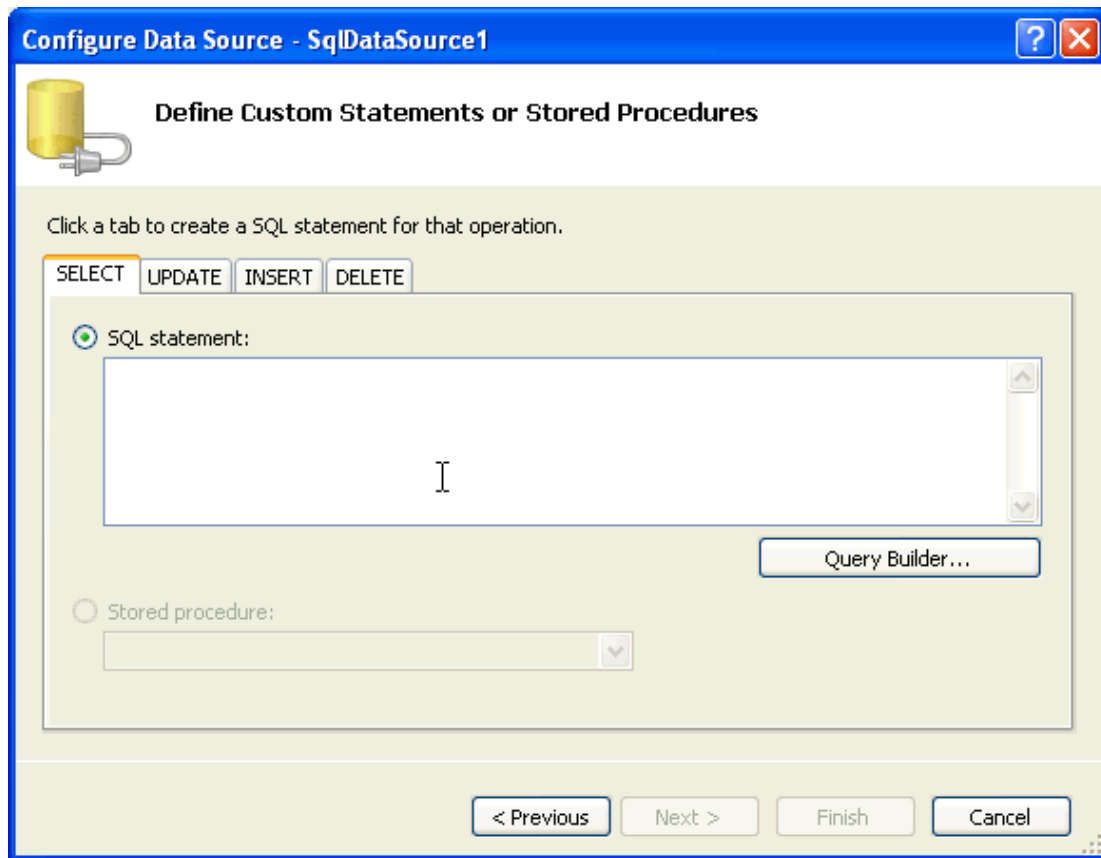
ORDER BY...

Advanced...

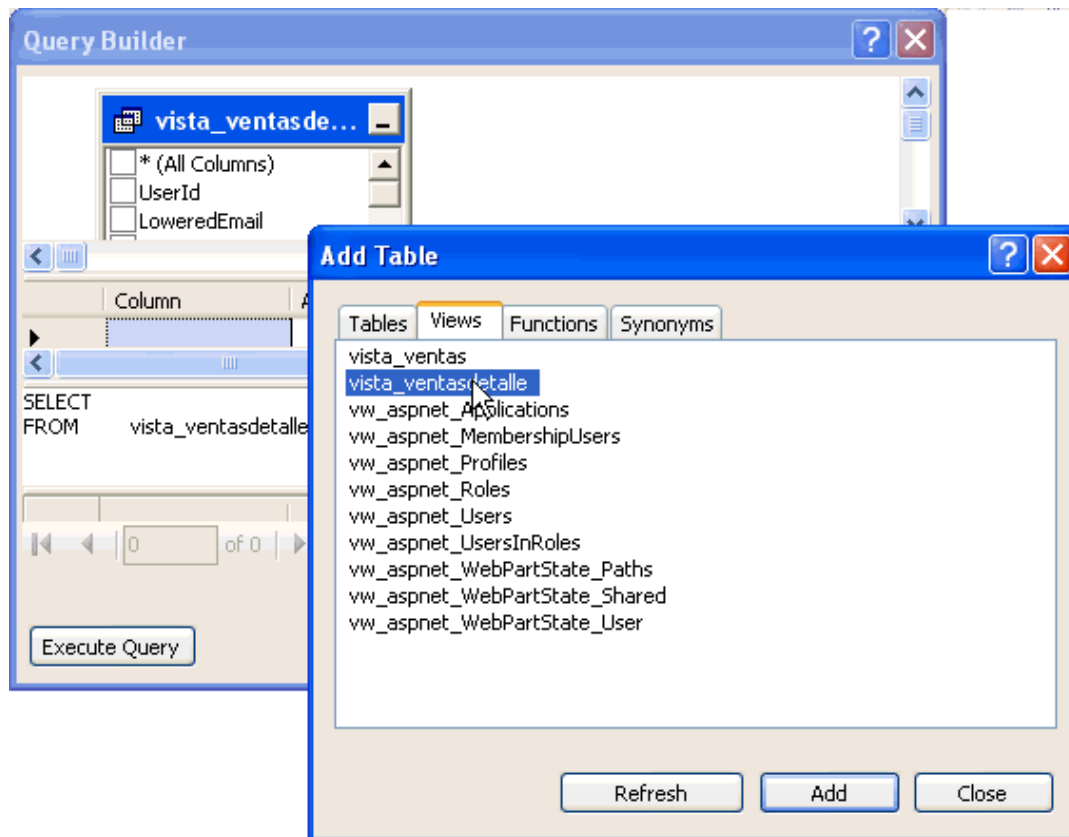
SELECT statement:

< Previous Next > Finish Cancel

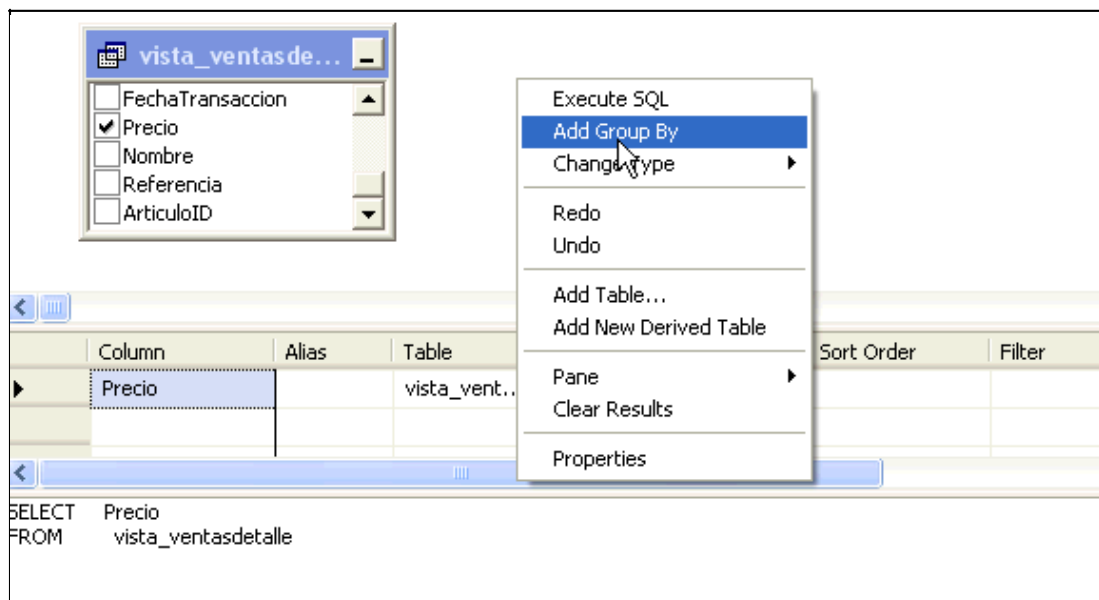
Que nos permitirá general una consulta SQL para poder realizar esta suma:



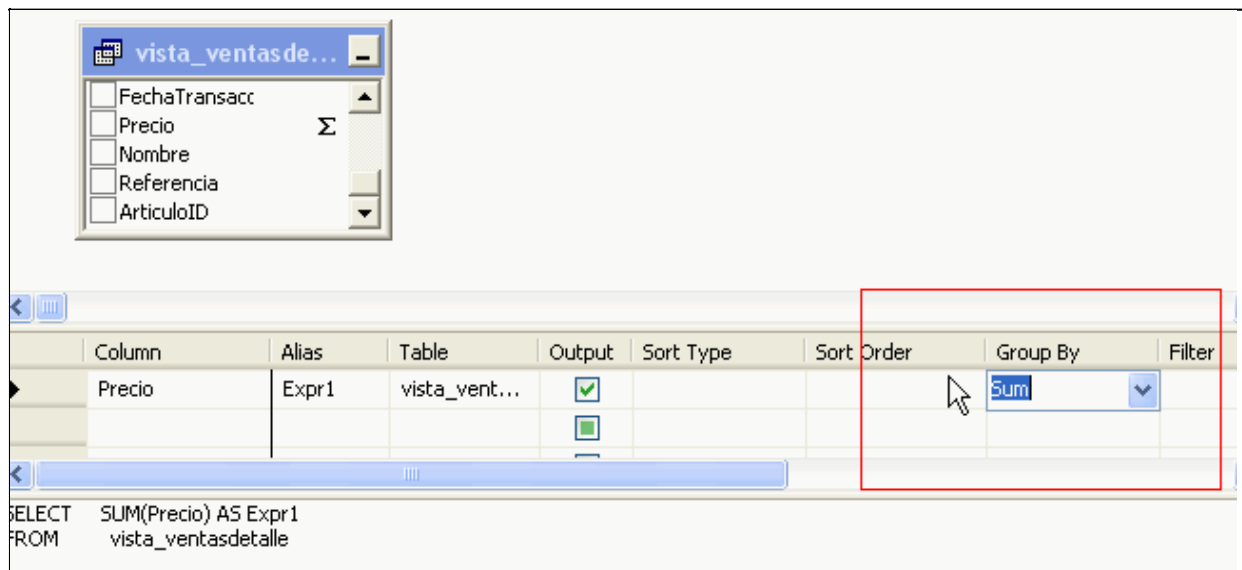
Como ves la versatilidad es total porque podemos indicar las sentencias SQL que queremos ejecutar para los cuatros procesos habituales en las bases de datos: altas, bajas, consultas y modificaciones. Pero ahora queremos utilizar el generador para que nos haga una "autosuma", si conoces Access te sonará bastante la siguiente pantalla. Asi que pulsamos en el botón de "Query Builder" y añadimos la yista:



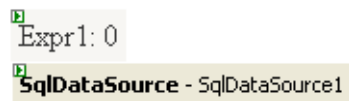
Ahora seleccionamos el campo del precio y seleccionamos con el botón derecho "Group By":



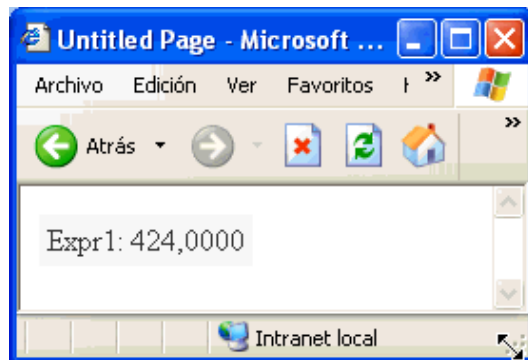
Que nos producirá un agrupamiento en la columna "Group By", la editamos para poner que el agrupamiento es para hacer un sumatorio:



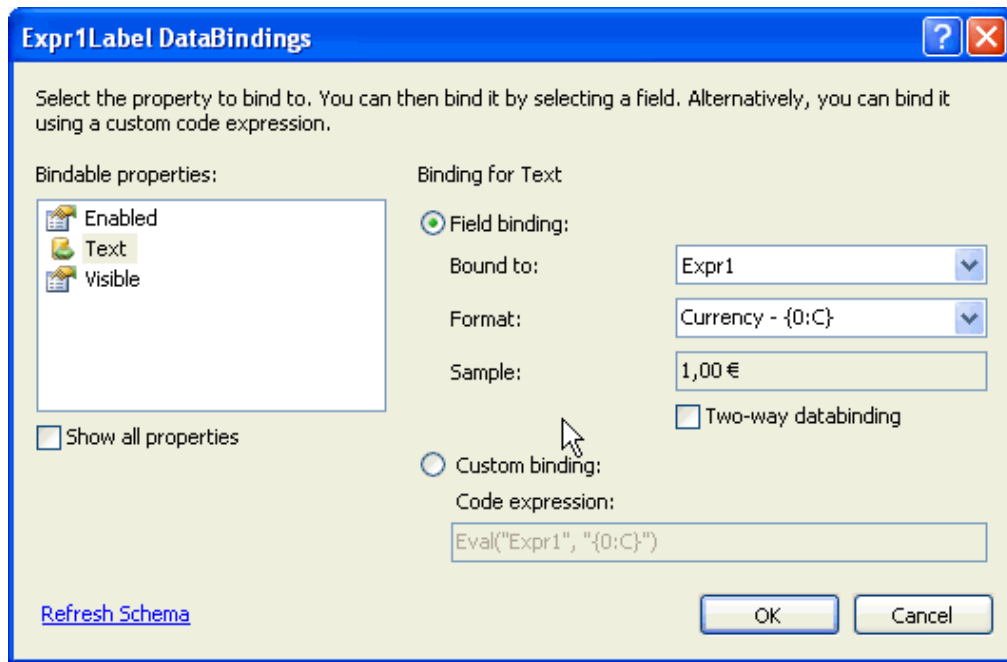
Finalizamos este generador y la selección de datos para volver a la página que ahora queda:



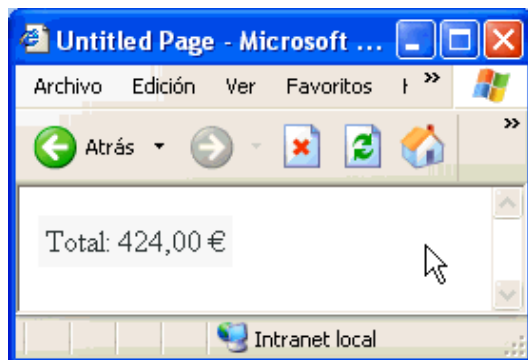
Y vista en el navegador:



Vamos a darle un poco de formato así que editamos su plantilla, luego con el botón derecho encima de la expresión que nos ha generado "Expr1" seleccionamos "Edit Data Bindings" para editar su origen de datos y así su formato:



Como ves en "Format" le he puesto el formato de moneda para que quede mejor presentado. También he desmarcado la opción de "Two-way databinding" que solo se necesita si vamos a editar este cuadro de texto. Pero como te dije antes, si quieres editar datos lo mejor es utilizar el otro control, este lo reservaremos para este tipo de cosas... bueno terminemos con este texto para quedar:



He puesto el literal de "total" y vemos que el campo de tipo moneda aparece ya con formato.

6.1 Mostrar subtotales

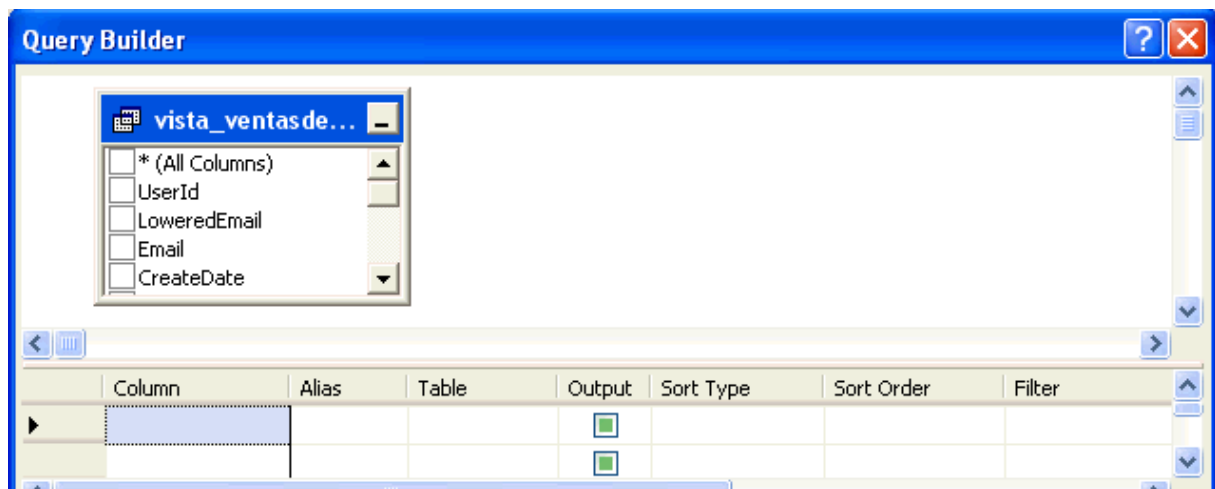
Mostrar un total es interesante pero lo mas habitual es mostrar subtotales agrupados por proveedor, cliente, articulo, ... Podemos añadir un control de cuadrícula en el mismo sitio que hemos añadido este del total. Bajamos el control con unos cuantos "intros" y ponemos un control "GridView" en la parte superior:

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

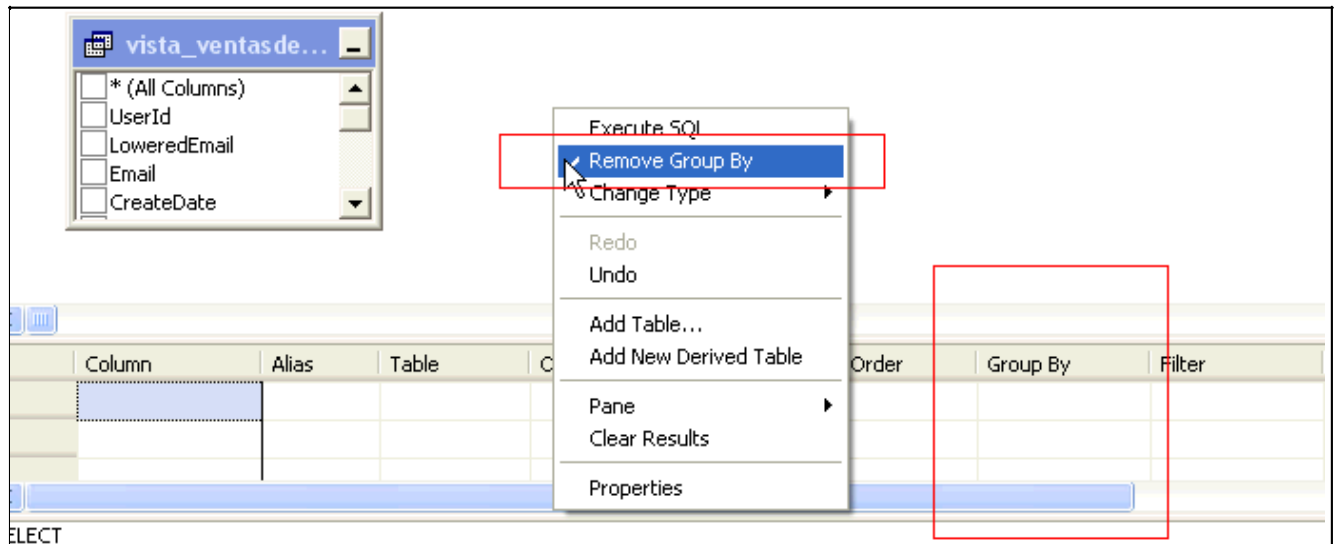
Total: 0,00 €

SqlDataSource - SqlDataSource1

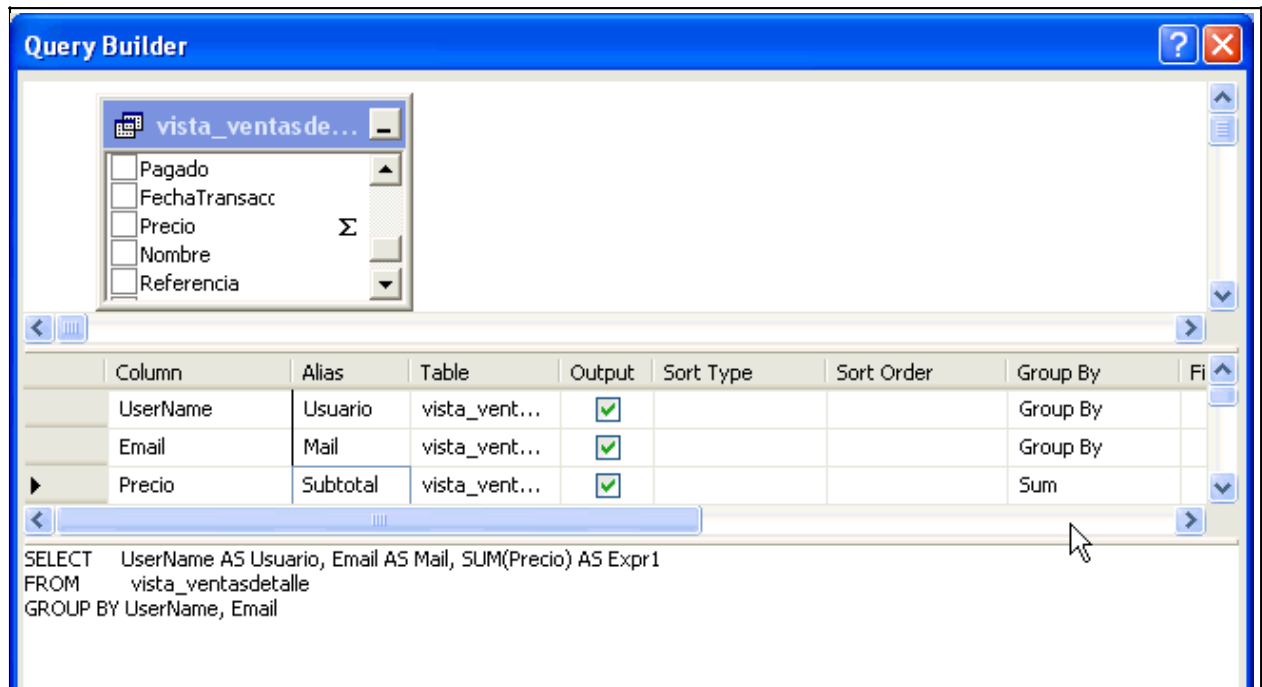
En la cuadrícula hacemos lo de antes, creamos la conexión habitual y yolyemos a crear la cadena SQL con el generador de consultas:



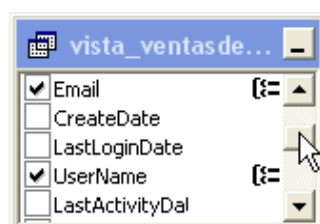
Añadimos la misma vista que el caso anterior. Desde luego que el criterio de agrupamiento es absolutamente libre, yamos a hacer un ejemplo pero lo puedes adaptar lógicamente a tu gusto. Vamos a hacer un subtotal por usuarios, asi que pulsamos otra vez con el botón derecho en el espacio donde está la vista en la parte de arriba para añadir una agrupación:



Esto lo que hace es añadir la columna del agrupamiento. Y ahora le decimos que agrupe por el nombre de usuario y por el mail para poner en el precio la función de sumar:

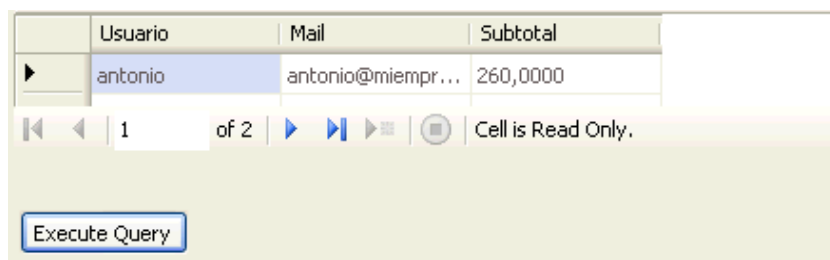


Fíjate en la instrucción SQL generada, como te has repasado el manual de SQL que te proporcioné hace unos capítulos no tendrás problema en interpretarla. Hemos utilizado la columna de "Alias" para que nos escriba un nombre distinto al del nombre del campo. Las llaves que aparecen en la parte de la vista (arriba) indican que esas van a ser las agrupaciones que ya a realizar:

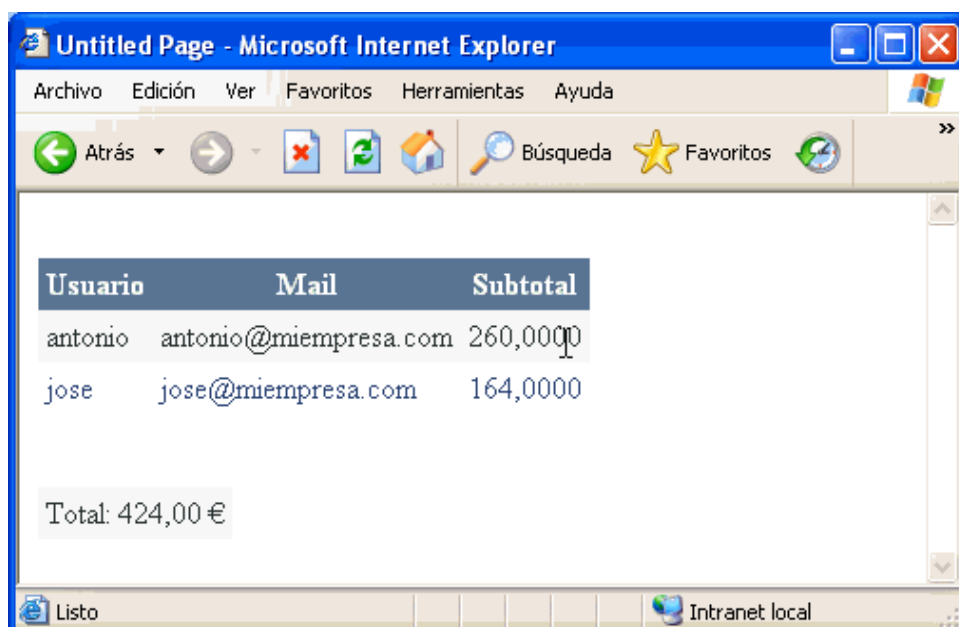


Perfiles y otros controles ASP.NET

En la parte de abajo tienes un botón para comprobar si la consulta generada devuelve los valores esperados:



Así que en la página obtendremos exactamente eso:

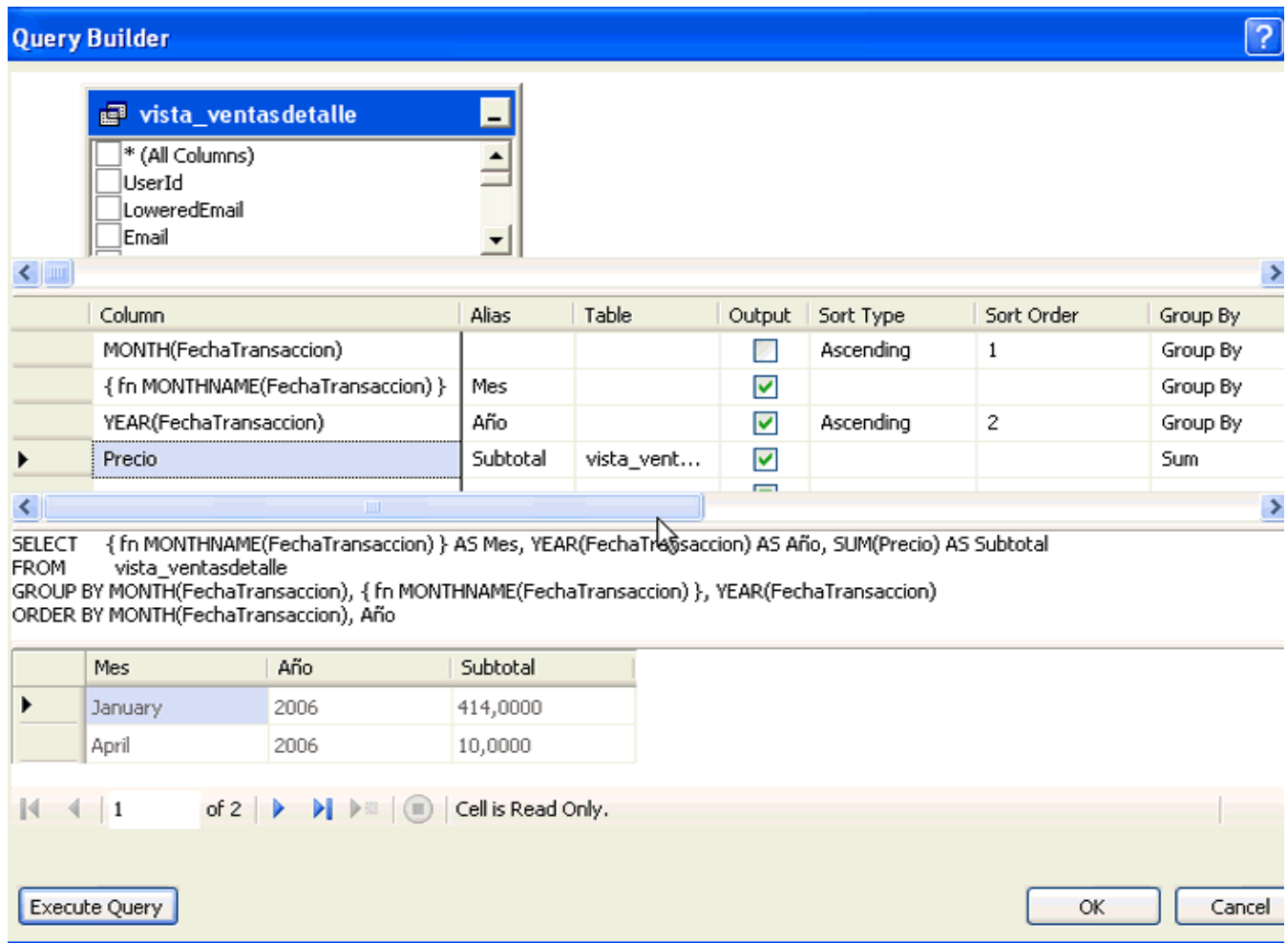


Subtotales con fechas

Si queremos hacer los subtotales con fechas, por ejemplos por meses necesitaremos utilizar algunas expresiones, con esto terminamos de ver las posibilidades de estos controles. Podemos hacer agrupaciones por estas expresiones:

- Month (fechatransaccion), lo utilizaremos para que nos lo orden por este valor, no marcamos la casilla de verificación de visible, así no se mostrará.
- Monthname (fechatransaccion). Extrae el nombre del mes para agrupar los resultados por él.
- Year (fechatransaccion). La utilizaremos junto con la del mes para la ordenación así se ordenará por el año y el mes.

La consulta quedará de esta forma:



Debajo puedes ver el resultado que obtendremos.

Hemos terminado de ver este interesante capítulo de acceso a datos, con esto tienes la base suficiente para hacer pequeñas explotaciones con resultados muy buenos. En el curso de ASP.NET 2.0 avanzado verás muchas mas posibilidades...

7. Despedida.

Bueno, hemos llegado al final de este curso... espero que haya sido de tu agrado y te sirva para empezar a trabajar en este interesante mundo. Ahora tienes la segunda parte de este curso con ejemplos muy interesantes y que tiene un nivel muy bueno, viendo prácticamente toda la tecnología necesaria de ASP.NET necesaria para la gran mayoría de los sitios Webs. Totalmente recomendable.

Ya ves que ha sido muy extenso, seguramente mucho mas de lo que te imaginabas, la idea es que te sirva de referencia porque hemos visto la gran mayoría de posibilidades que tiene ASP.NET 3.5. Ahora es el momento de hacer pequeñas explotaciones de bases de datos. Imagina que solo hubiese sido una "introducción" sin ver bases de datos o temas de seguridad, el curso se quedaría totalmente escaso. Es mejor así porque con esto tienes un nivel "básico-medio-casi avanzado" que seguramente cumpla con tus expectativas sobre el curso.

Perfiles y otros controles ASP.NET

Luego te meterás en el mundo de las páginas maestras para dar un aspecto homogéneo al web y poner los controles de navegación. Por último deberás meterte con la parte de los perfiles de usuario para crear zonas protegidas con formularios de validación.

Recuerda que no es necesario utilizar todo sino solo que necesites. Una aplicación web puede ser perfecta y potente utilizando solo la parte de las bases de datos sin necesidad de meterse con perfiles o páginas maestras así que repasa y practica poco a poco ya que requiere su aprendizaje despacio y sobre todo haciendo ejemplos sencillos.

Recuerda también que Internet es una fuente inagotable de ejemplos, dudas, problemas. Prácticamente todo lo que necesites lo tienes en Internet... en el curso adjunto unos cuantos enlaces para empezar a curiosar, algunos en español!.

Gracias por tu atención y ánimo con ASP.NET

Ejercicios

Ejercicio 1

Modifica el perfil de usuario para que soporte la fecha de alta en el portal.

Debes modificar el fichero de configuración web.config y los controles necesarios en la pantalla de edición del perfil:

```
<properties>
  <add name="Nombre" type="System.String"/>
  <add name="Apellidos" type="System.String"/>
  <add name="Direccion1" type="System.String"/>
  <add name="Direccion2" type="System.String"/>
  <add name="Ciudad" type="System.String"/>
  <add name="Provincia" type="System.String"/>
  <add name="CodigoPostal" type="System.String"/>
  <add name="Pais" defaultValue="España" type="System.String"/>
  <add name="FechaNacimiento" type="System.DateTime"/>
  <add name="TemaPreferido" defaultValue="Clasico"/>
  <add name="FechaAlta" type="System.DateTime"/>
</properties>
```

Y la modificación en la página:

```
Protected Sub btnenviar_Click(ByVal sender As Object, ByVal e As EventArgs)
    Profile.Nombre = txtnombre.Text
    Profile.Apellidos = txtapellidos.Text
    Profile.Direccion1 = txtdireccion1.Text
    Profile.Direccion2 = txtdireccion2.Text
    Profile.Ciudad = txtciudad.Text
    Profile.Provincia = txtprovincia.Text
    Profile.CodigoPostal = txtcodigopostal.Text
    Profile.FechaNacimiento = fecha.SelectedDate
    Profile.Pais = txtpais.Text
    Profile.TemaPreferido = lista_estilos.SelectedValue
    'La fecha de alta la pondremos a mano porque es la que
    'tiene el sistema en el momento de grabar:
    Profile.FechaAlta = Date.Today
End Sub
```

Ejercicio 2

Crea una página de consulta para la base de datos de usuarios. Podremos buscar por nombre, ciudad, sexo y además puede ser exacta o aproximada.

de MiEmpresa.Com

ContentPlaceHolder1 (Personalizado) asp:panel#Panel1

Consulta de usuarios

Buscar:

Tipo de búsqueda: ☒ Búsqueda exacta ☒ Aproximada

Provincia: ☐ [chk_provincia] DataBound

Sexo: ☒ Hombre ☐ Mujer

Buscar

SqlDataSource - SqlDataSource1

Se pedirán los datos y se hace un postback cruzado para construir la sentencia Select en una página destino. Las provincias se cargarán de la tabla de provincias con un origen de datos. Luego se muestra el resultado en otra página con un datagrid, por ejemplo...

La consulta con las provincias se hará solo si se activa la casilla de verificación.

Puedes hacer la página destino por código personalizando la sentencia SQL de la consulta y luego haciendo un databind con la cuadrícula.

Quedaría la pantalla de consulta así:

Consulta de usuarios

Buscar:

Tipo de búsqueda:

☐ Exacta
 ☒ Aproximada

Provincia:

☐ Barcelona

Sexo:

☒ Hombre
 ☐ Mujer

Y el resultado:

nombre	apellidos	direccion	fecha	localidad	provincia	sexo
Jose	Rodriguez	Jorge Vigón	19/03/1965 0:00:00	Logroño	La Rioja	<input checked="" type="checkbox"/>

SQL ejecutada: `select usuarios.nombre,usuarios.apellidos,usuarios.direccion,usuarios.fecha,usuarios.provincia=provincias.id and usuarios.nombre like '%j%' and usuarios.sexo=1`