

	<p align="center">Instituto Gubernamental España Jesús Milla Selva Centro de Informática</p>	
<p align="center">II SEMESTRE Asignatura: Programación IV</p>		
<p align="center"><u>UNIDAD Nº 2:</u> Acceso a una Base de Datos con ASP.Net</p>	<p align="center"><u>TEMA:</u> Control GridView</p>	

Objetivos:

- Utilizar el control GridView para visualizar datos.
- Configurar de forma correcta el control GridView.
- Editar, modificar y consultar datos a partir de un GridView.

Introducción

El control GridView de ASP.Net permite visualizar datos en una tabla en pantalla, editar, modificar y borrar registros del mismo.

El GridView es un control extremadamente flexible para mostrar tablas multicolumna.

Cada registro de una consulta de un select configurado en un SqlDataSource genera una fila en la grilla. Cada campo en el registro representa una columna en la grilla.

El GridView es el control más poderoso que provee el ASP.Net. Veremos que este control trae funcionalidades ya implementadas para paginación, ordenamiento y edición de sus datos.

GridView

Datos de una tabla

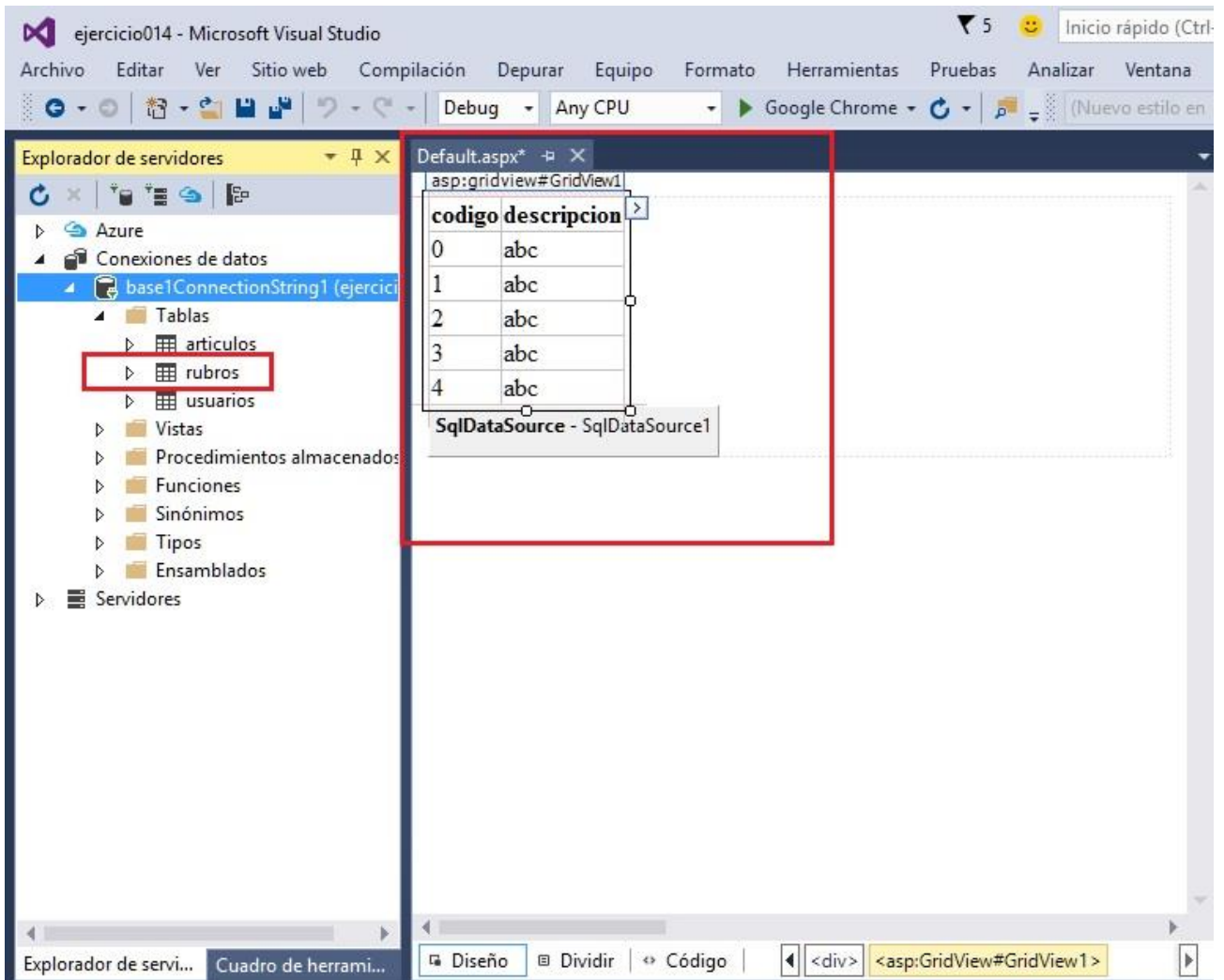
Crearemos un proyecto para probar el control GridView y las diferentes opciones que nos brinda.

Crear un sitio web en el Visual Studio llamado ejercicio014.

Agregar un Web Form llamado Default.aspx. Iremos al Explorador de servidores y seleccionaremos la tabla "rubros" y la arrastraremos al formulario web. Veremos que se generan dos objetos sobre la página:

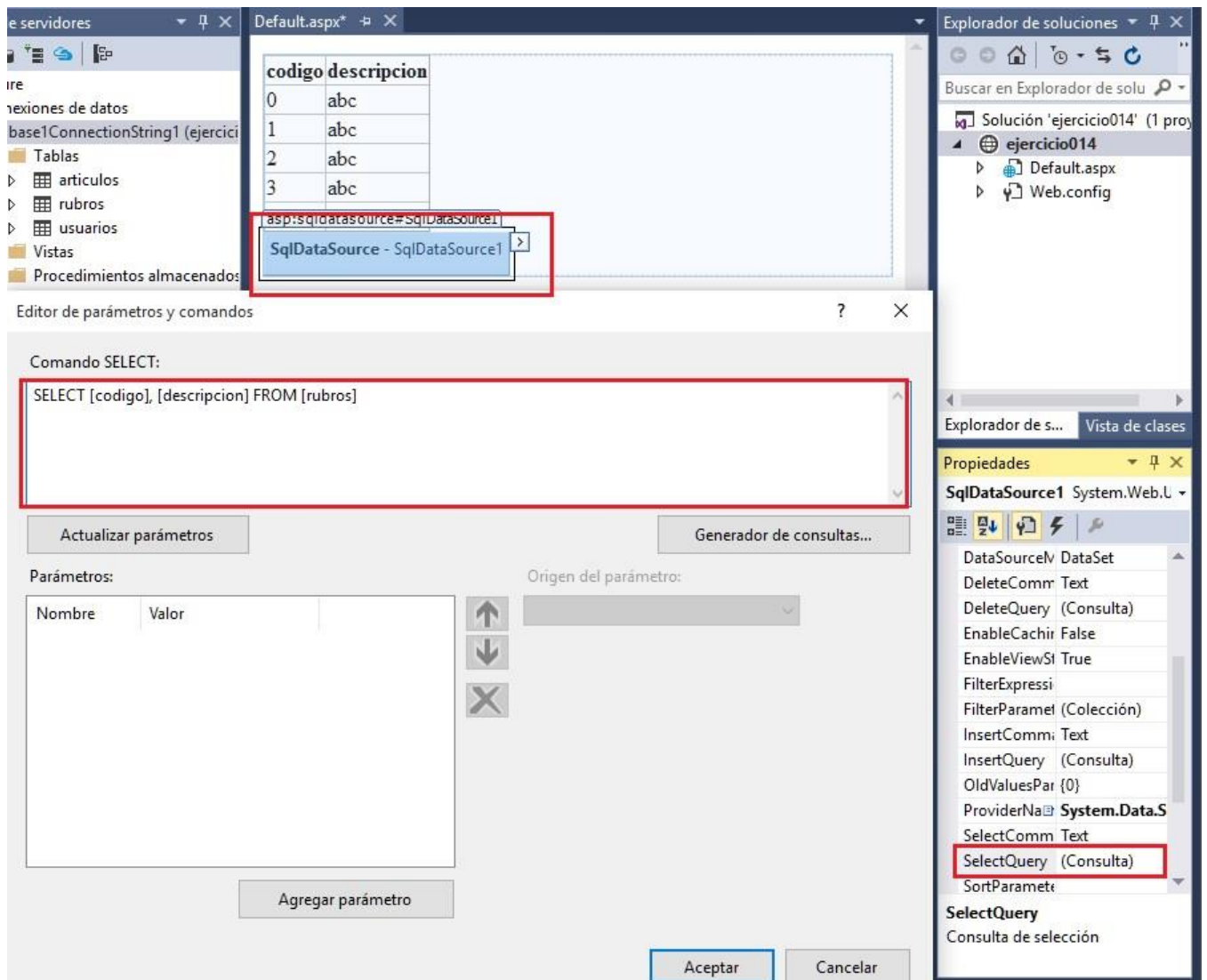
Un objeto de la clase GridView llamado GridView1.

Un objeto de la clase SqlDataSource llamado SqlDataSource1.



Si seleccionamos el objeto SqlDataSource1 y observamos el contenido de la propiedad SelectQuery, veremos que ya está configurado el comando SELECT:

```
SELECT [codigo], [descripcion] FROM [rubros]
```



El comando `SELECT` indica rescatar todas las filas de la tabla `rubros`.

Podemos ver también que se han configurado automáticamente las propiedades `InsertQuery`, `DeleteQuery` y `UpdateQuery` con los valores:

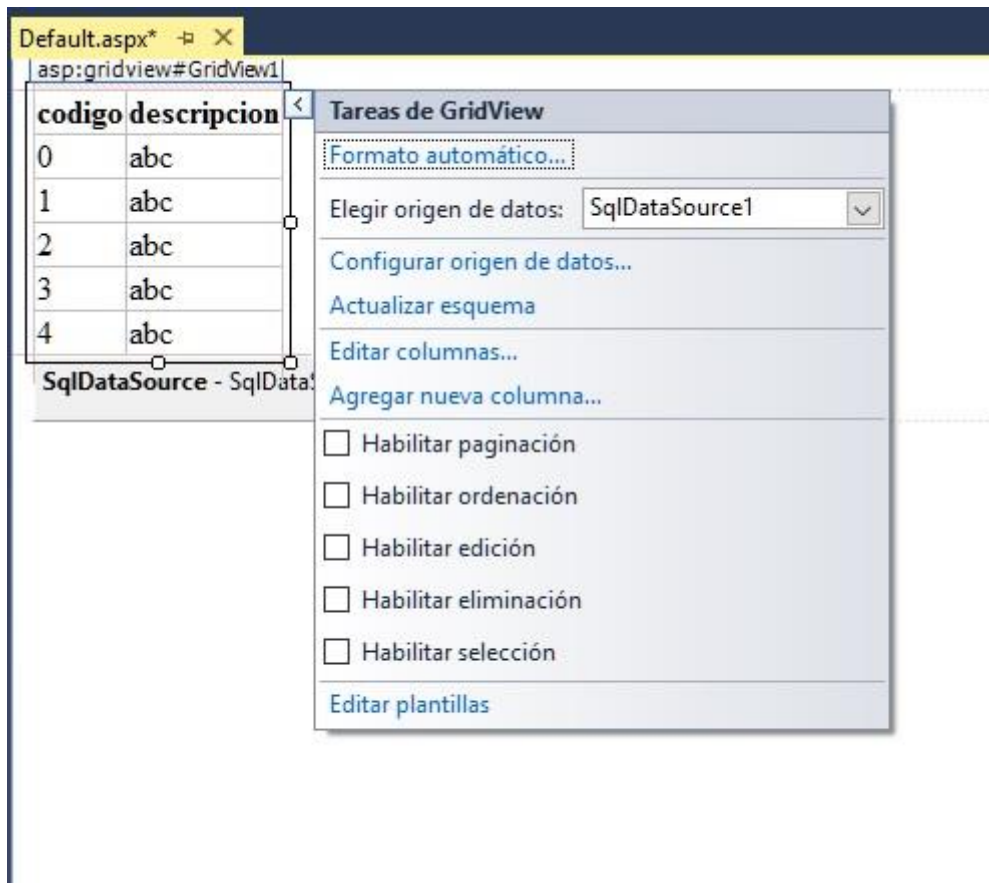
```
INSERT INTO [rubros] ([descripcion]) VALUES (@descripcion)
```

```
DELETE FROM [rubros] WHERE [codigo] = @codigo
```

```
UPDATE [rubros] SET [descripcion] = @descripcion WHERE [codigo] = @codigo
```

Como podemos ver hasta este momento la herramienta Visual Studio .Net nos ha configurado en forma automática el control `SqlDataSource1`, sólo nos queda configurar el control `GridView1`.

Seleccionamos el control `GridView` y presionamos el botón presente en la parte superior derecha, el mismo nos muestra una serie de funcionalidades básicas del control:



Como podemos ver ya está configurado el origen de datos con el objeto `SqlDataSource1`.

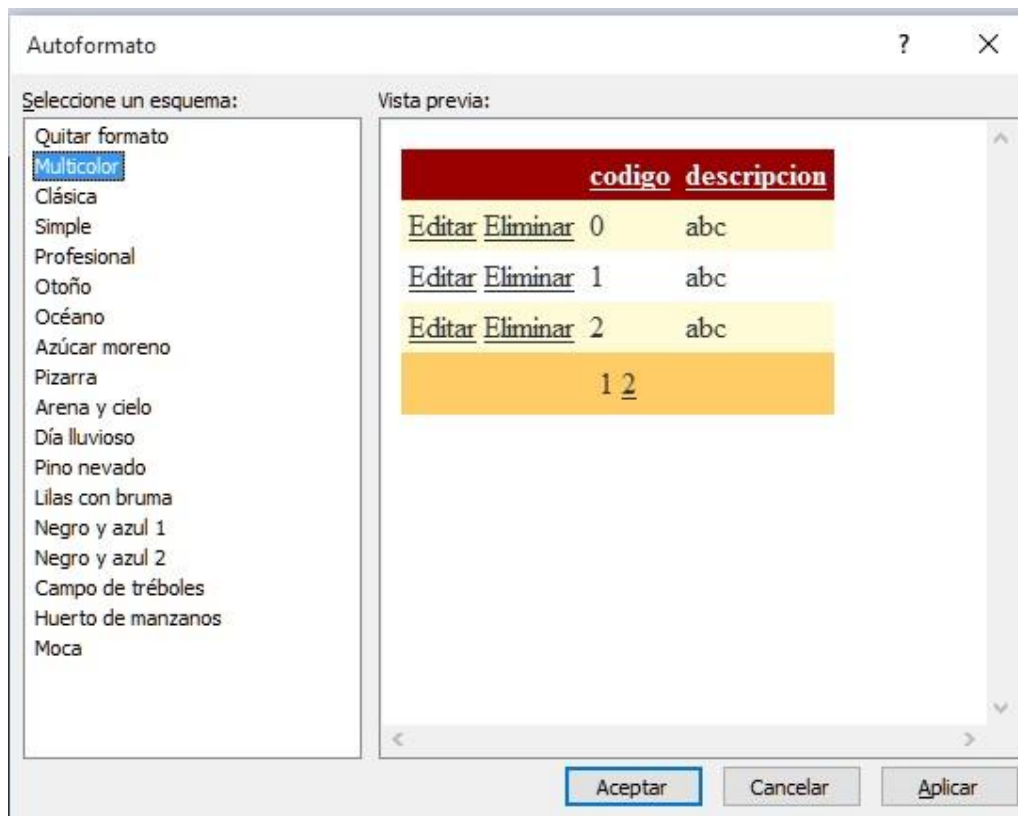
Habilitemos la paginación, ordenamiento, edición y eliminación.

Ejecutemos el proyecto y comprobaremos que tenemos en la página los datos de la tabla “rubros” con la capacidad de modificar y borrar registros. Además, está activa la paginación y ordenamiento por cualquiera de las dos columnas de la tabla (cambiamos la propiedad `PageSize` del objeto `GridView` por el valor 3)



Sin escribir una sola línea de código tenemos el mantenimiento de la tabla rubros (con la excepción del alta).

Veamos otras características que podemos configurar en forma visual del control GridView. Desde el botón ">" que se encuentra en la parte superior derecha del control GridView1 podemos seleccionar la opción "Formato Automático..." para definir la presentación de la tabla con plantillas predefinidas de color y fuente:



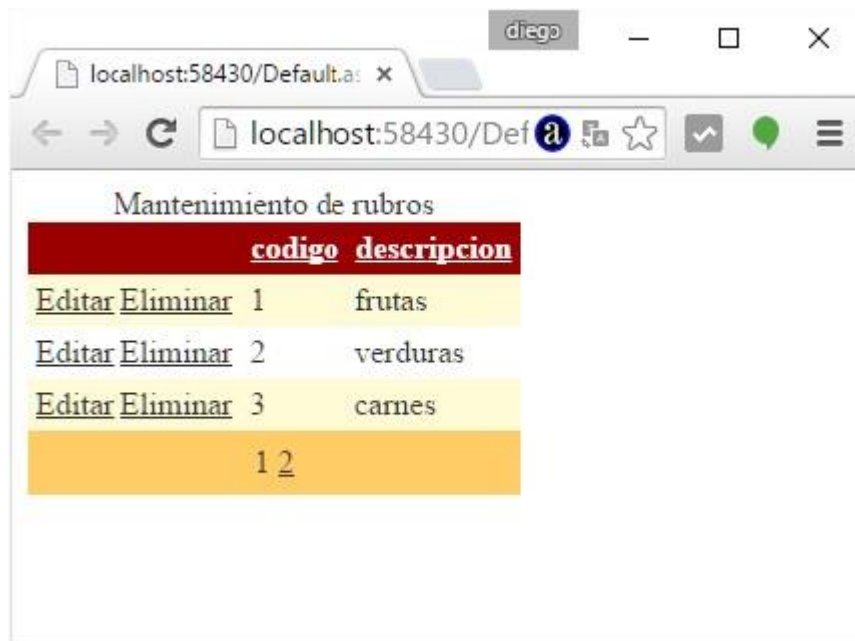
Luego de seleccionar el esquema para el GridView, presionamos aceptar y tenemos ya definido el nuevo formato de la grilla.

Desde la ventana de propiedades configuraremos las propiedades:

Caption: Es un título que aparece en la parte superior del GridView.

PageSize: Cantidad de registros a mostrar por página.

Luego de esto ejecutamos nuevamente y tenemos como resultado:



Datos de varias tablas

Continuamos con las tablas:

Tabla: articulos

```
codigo int Clave primaria e identidad.
descripcion varchar(50)
precio float
codigorubro int
```

Tabla: rubros

```
codigo int Clave primaria e identidad
descripcion varchar(50)
```

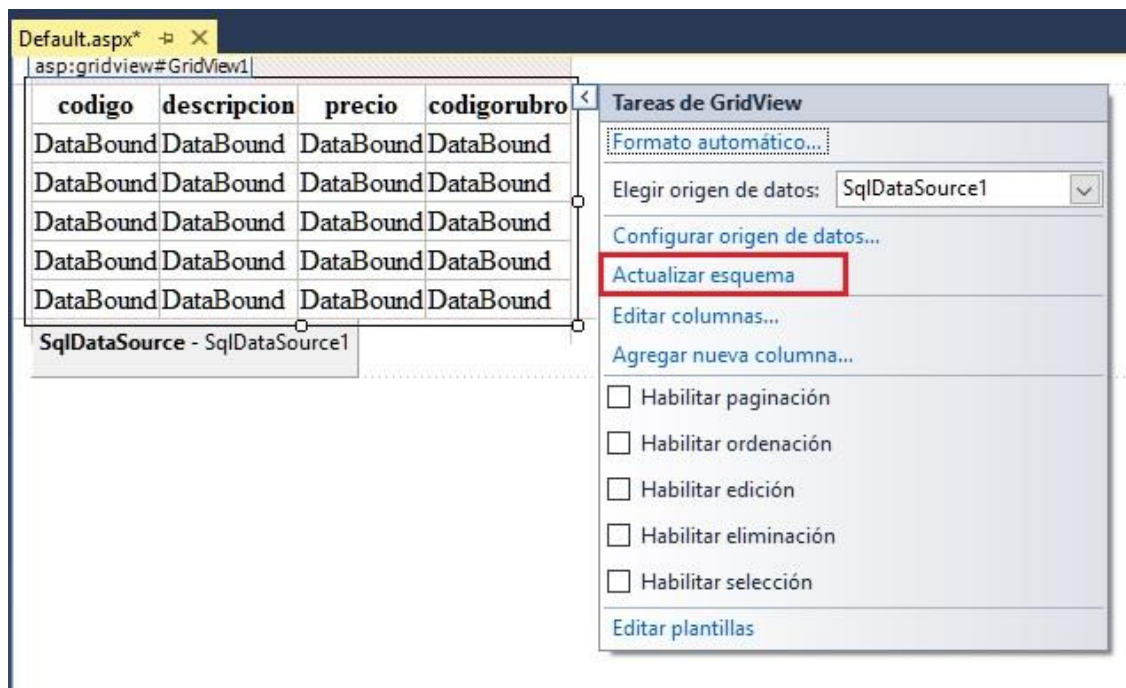
Crearemos un nuevo Sitio Web llamado "ejercicio015" y agregaremos un webform llamado Default.aspx. Seleccionamos desde el Explorador de servidores la tabla articulos y la disponemos dentro del webform. El entorno del Visual Studio .Net nos genera un objeto de la clase GridView y otro de la clase SqlDataSource.

El objetivo final es mostrar el código del artículo, su descripción, su precio y finalmente la descripción del rubro (no el código de rubro).

1. Primero seleccionamos el control `SqlDataSource1` y configuramos la propiedad `SelectQuery` con el comando `Select` que rescata los datos haciendo el emparejamiento por la columna `codigorubro` de la tabla `articulos` y `codigo` de la tabla `rubros`:

```
SELECT ar.codigo,
       ar.descripcion as descriarticulo,
       precio,
       ru.descripcion as descrirubro
from articulos as ar
join rubros as ru on ru.codigo=ar.codigorubro
```

2. Luego de configurar la propiedad `SelectQuery` debemos actualizar el esquema del `SqlDataSource1`, esto lo hacemos seleccionando el objeto sobre el formulario y seleccionamos la opción "Actualizar esquema".



Con esto logramos que se refresquen las columnas a mostrar en el `GridView1`.

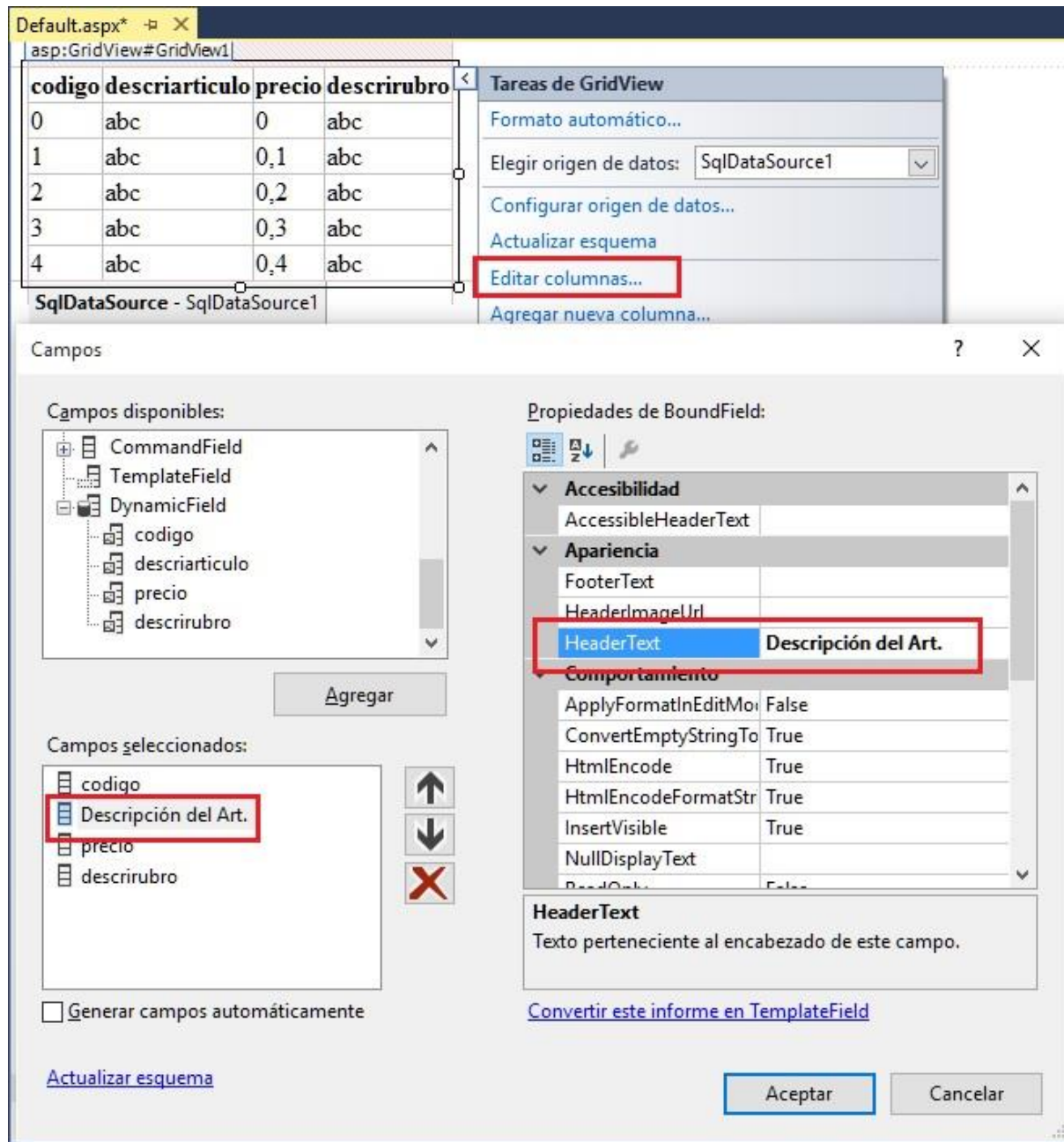
Si ejecutamos, podemos ver que ya tenemos la tabla que rescata todos los artículos y asociado a cada artículo la descripción del rubro al que pertenece:



codigo	descriarticulo	precio	descrirubro
2	manzanas	17	frutas
1002	naranjas	7	frutas
1003	bananas	13	frutas
1004	papas	5	verduras
1005	lechuga	9	verduras

Ahora configuraremos algunas propiedades para dar un formato y presentación más adecuada de los datos:

- A. Ya vimos que seleccionando el GridView1 y mediante la opción “Formato automático...” podemos definir el estilo de presentación de la grilla.
- B. Ahora entramos a la opción “Editar columnas...” y seleccionamos el campo a configurar:



Cambiamos los títulos de las columnas de cada campo (por ejemplo, en el campo descriarticulo mostraremos el título "Descripción del Art." Modificando la propiedad HeaderText.

De forma similar cambiar los otros títulos de las columnas de la grilla:



codigo	Descripción del Art.	precio	rubro
2	manzanas	17	frutas
1002	naranjas	7	frutas
1003	bananas	13	frutas
1004	papas	5	verduras
1005	lechuga	9	verduras

- C. La propiedad `Visible` de cada columna nos permite configurar si la columna se muestra o no.
- D. La propiedad `DataFormatString` nos permite configurar la apariencia de números y fechas. Por ejemplo, si queremos que el precio aparezca con el símbolo de moneda debemos configurar la propiedad `DataFormatString` con el valor: `{0:C}`



codigo	Descripción del Art.	precio	rubro
2	manzanas	17,00 €	frutas
1002	naranjas	7,00 €	frutas
1003	bananas	13,00 €	frutas
1004	papas	5,00 €	verduras
1005	lechuga	9,00 €	verduras

Algunos ejemplos de formato de campos:

Tipo	Formato	Ejemplo
Moneda	{0:C}	\$120.50
Porcentaje	{0:P}	40%
Decimales fijos	{0:F4}	12.4567
Notación científica	{0:E}	1.233E003
Fecha corta	{0:d}	12/25/2005

Formato de filas individuales de acuerdo con ciertas condiciones

Cuando se grafica la tabla podemos capturar el evento Row DataBound y configurar como mostrar dicha fila de la tabla.

A modo de ejemplo mostraremos de color amarillo las filas de los artículos con precio superior a 8. Para esto codificamos el evento Row DataBound del GridView1.

The screenshot shows a Visual Studio IDE with a web application project named 'ejercicio015'. The main window displays a GridView1 control with the following data:

codigo	Descripción del Art.	precio	rubro
0	abc	0,00 €	abc
1	abc	0,10 €	abc
2	abc	0,20 €	abc
3	abc	0,30 €	abc
4	abc	0,40 €	abc

The first row (codigo 0) is highlighted in yellow. Below the GridView1 control, a SqlDataSource1 is visible. The Properties window on the right shows the 'Gridview1' control with the 'RowDataBound' event selected in the 'Events' list.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void GridView1_RowDataBound(object sender,
GridViewRowEventArgs e)
    {
        if (e.Row.RowType == DataControlRowType.DataRow)
        {
            double precio;
            precio = (double)DataBinder.Eval(e.Row.DataItem,
"precio");
            if (precio > 8)
            {
                e.Row.ForeColor = System.Drawing.Color.Red;
                e.Row.BackColor =
System.Drawing.Color.Yellow;
                e.Row.Font.Bold = true;
            }
        }
    }
}

```

Con el if verificamos si el evento se disparó para una fila de datos de la grilla (ya que este método se dispara cuando dibuja la cabecera (DataControlRowType.Header), el pie de grilla (DataControlRowType.Footer) etc.

Luego rescatamos el valor del campo precio y verificamos con un nuevo if si el precio supera 8, en caso afirmativo modificamos el color de fondo (BackColor) y de frente de la fila.



codigo	Descripción del Art.	precio	rubro
1	peras	12,00 €	frutas
2	manzanas	4,00 €	frutas
3	bananas	6,00 €	frutas
4	papas	5,00 €	verduras
5	lechuga	15,00 €	verduras

Selección de una fila del GridView y posterior extracción de sus datos

En muchas situaciones es necesario que el usuario seleccione una fila de la grilla para reflejar dicho dato en otra parte de la página o hacer otra consulta.

Para poder implementar esta característica del GridView llevaremos a cabo los siguientes pasos:

1. Cambiaremos el valor de la propiedad `SelectedRowStyle.BackColor` por amarillo (es decir que cuando seleccionemos la fila el color de fondo de la misma se activará con este valor)
2. En el menú de opciones que se despliega en la parte derecha del GridView 1 activaremos el CheckBox "Habilitar selección"
3. Dispondremos una Label en el webform para mostrar el valor seleccionado de la grilla (solo a modo de ejemplo)
4. Para el evento `SelectedIndexChanged` del GridView 1 codificaremos el siguiente código:

```
protected void GridView1_SelectedIndexChanged(object sender,
EventArgs e)
{
    Label1.Text =
this.GridView1.Rows[GridView1.SelectedIndex].Cells[1].Text;
}
```

El objeto `Rows` del `GridView` almacena una colección de filas, mediante el valor devuelto por la propiedad `SelectedIndex` de la grilla podemos acceder a la celda que almacena el código del artículo.

	codigo	Descripción del Art.	precio	rubro
Seleccionar	1	peras	12,00 €	frutas
Seleccionar	2	manzanas	4,00 €	frutas
Seleccionar	3	bananas	6,00 €	frutas
Seleccionar	4	papas	5,00 €	verduras
Seleccionar	5	lechuga	15,00 €	verduras

Esta información nos es muy útil para mostrar información adicional sobre el registro en otro control, por ejemplo.