

ADO.NET con Base de datos SQL Server

1. Modelo de objetos ADO.NET

ADO.NET es un conjunto de librerías orientadas a objetos que permiten interactuar con el origen de datos (data source). El origen de datos es una base de datos como también puede ser un archivo de texto, planilla electrónica o un archivo XML. En este tutorial tomaremos a ADO.NET como una forma de comunicarse con una base de datos.

El modelo de objetos ADO.NET proporciona una API (application programming interface) para acceder a los sistemas de bases de datos mediante la programación. El IDE de Visual Studio .Net cuenta con herramientas de programación visual que simplifican el proceso de utilizar una base de datos en un proyecto. También posee herramientas visuales para crear una base de datos de distintos proveedores de BD.

System.Data es el espacio de nombres raíz de la API de ADO.NET.

System.Data.OleDb es el espacio de nombres (workspace) que contiene clases diseñadas para trabajar con cualquier origen de datos (Access, Oracle, MySQL, etc.)

System.Data.SqlClient contiene clases optimizadas para trabajar con bases de datos de Microsoft SQL Server.

Para este tutorial utilizaremos bases de datos de SQL Server 2010 que viene incluido en el Visual C# 2010.

2. Creación de una Base de Datos de SQL Server desde Visual Studio .NET

Los pasos a seguir para crear una base de datos SQL, una vez que abrimos el Visual Studio, son:

- 1) abrir el explorador de servidores (menú Ver)
- 2) mouse derecho en Conexiones de datos dentro de la ventana del explorador de servidores
- 3) elegir agregar conexión ...
- 4) dejamos el origen de datos que por default es:
Archivo de base de datos de Microsoft SQL Server (SqlClient)
- 5) escribir el nombre de la base de datos que deseamos crear
- 6) clic en Aceptar
- 7) clic en Aceptar en respuesta a si desea crear la base de datos en la ventana que aparece
- 8) click en + que aparece con la nueva Base de Datos creada en la ventana del explorador de servidores
- 9) mouse derecho sobre Tablas
- 10) elegir Agregar nueva Tabla de la ventana subyacente
- 11) escribir nombres de campos y su tipo de datos para la nueva tabla en la nueva ventana que se abre
- 12) cerrar la ventana de los campos de la nueva tabla, cuando se hayan terminado de ingresar
- 13) click en Aceptar en la ventana que aparece preguntando si va a grabar los datos de la tabla
- 14) ingresar un nombre para la tabla nueva en la nueva ventana que le ingresemos un nombre para la tabla
- 15) click en Aceptar para grabar la tabla con sus campos y nombre
- 16) repetir los pasos 8 a 14 para agregar nuevas tablas a la base de datos

Para agregar datos a las tablas de la base de datos creada:

- click en + de Tablas del árbol de la base de datos recién creada en la ventana del explorador de servidores
- mouse derecho sobre el nombre de tabla a la cual queremos agregar datos
- elegir mostrar datos de tabla
- agregar datos en la ventana que se abre mostrando una tabla con el nombre de las columnas de la tabla seleccionada

- cerrar la ventana cuando se terminen de ingresar todos los datos de las filas

3. Mostrar en el formulario una consulta a Base de Datos SQL Server

Vamos a crear un proyecto del tipo aplicación Windows y mostrar los datos de la tabla Titulos de la Base de Datos Libros.mdf compuesta por las tablas que se muestran en la figura 1

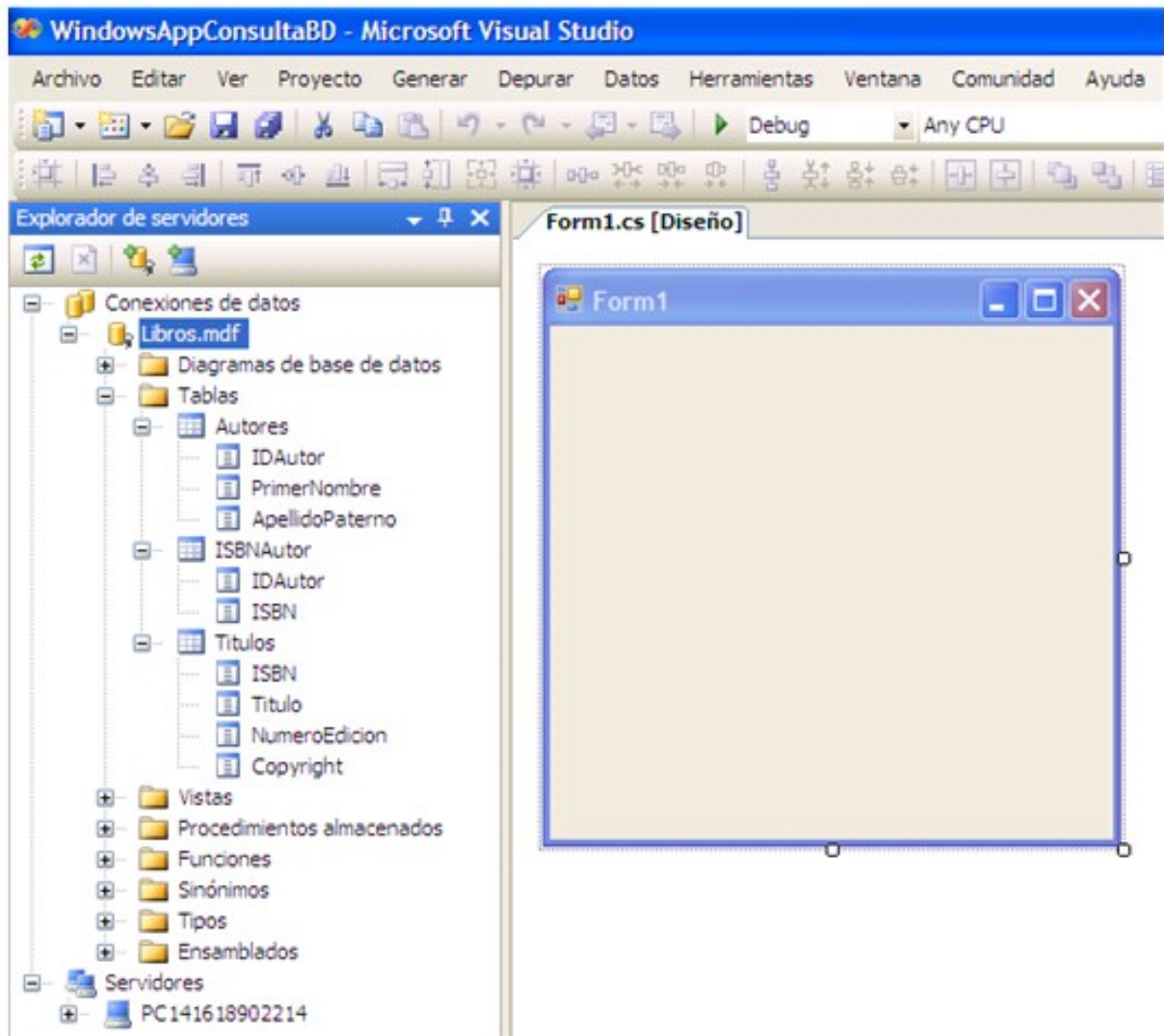


Figura 1

Para ello vamos a:

1. crear un proyecto nuevo del tipo aplicación Windows
2. abrir la ventana del explorador de servidores (menú Ver)
3. mouse derecho sobre Conexiones de datos dentro de la ventana del explorador de servidores
4. elegir agregar conexión ...
5. click en Examinar ...
6. elegir la base de datos Libros.mdf
7. click en Probar Conexión
8. ok si la conexión fué exitosa
9. click en Aceptar (aparece en el explorador de servidores la BD Libros.mdf)
10. click en Mostrar Orígenes de Datos (menú Datos)
11. click en Agregar nuevo origen de Datos de la ventana Orígenes de datos
12. click en Siguiente de la ventana Asistente para la configuración de orígenes de datos donde por default esta seleccionado Base de datos como origen de los datos
13. click en Siguiente de la ventana Asistente para la configuración de orígenes de datos donde por default esta seleccionado Libros.mdf para la conexión de datos
14. click en No en la ventana que pregunta si vamos a copiar el archivo BD al proyecto
15. click en Siguiente de la ventana Asistente para la configuración de orígenes de datos donde por default está tildada la opción de guardar la cadena de conexión
16. tildamos Tablas para elegir los datos de todas las tablas de la BD Libros
17. click en Finalizar

Aparece entonces la ventana de orígenes de datos según lo muestra la figura 2.

Además se agrega al árbol del proyecto un nodo llamado LibrosDataSet.xsd que se observa en la ventana del explorador de soluciones.

Para mostrar los datos de la tabla Titulos de la BD Libros en el formulario, lo haremos a través de un control DataGridView.

Utilizaremos las herramientas visuales, de la siguiente forma:

1. oprimir sin soltar mouse izquierdo sobre la tabla Titulos en la ventana de Orígenes de Datos

2. arrastrar y soltar el mouse en el formulario

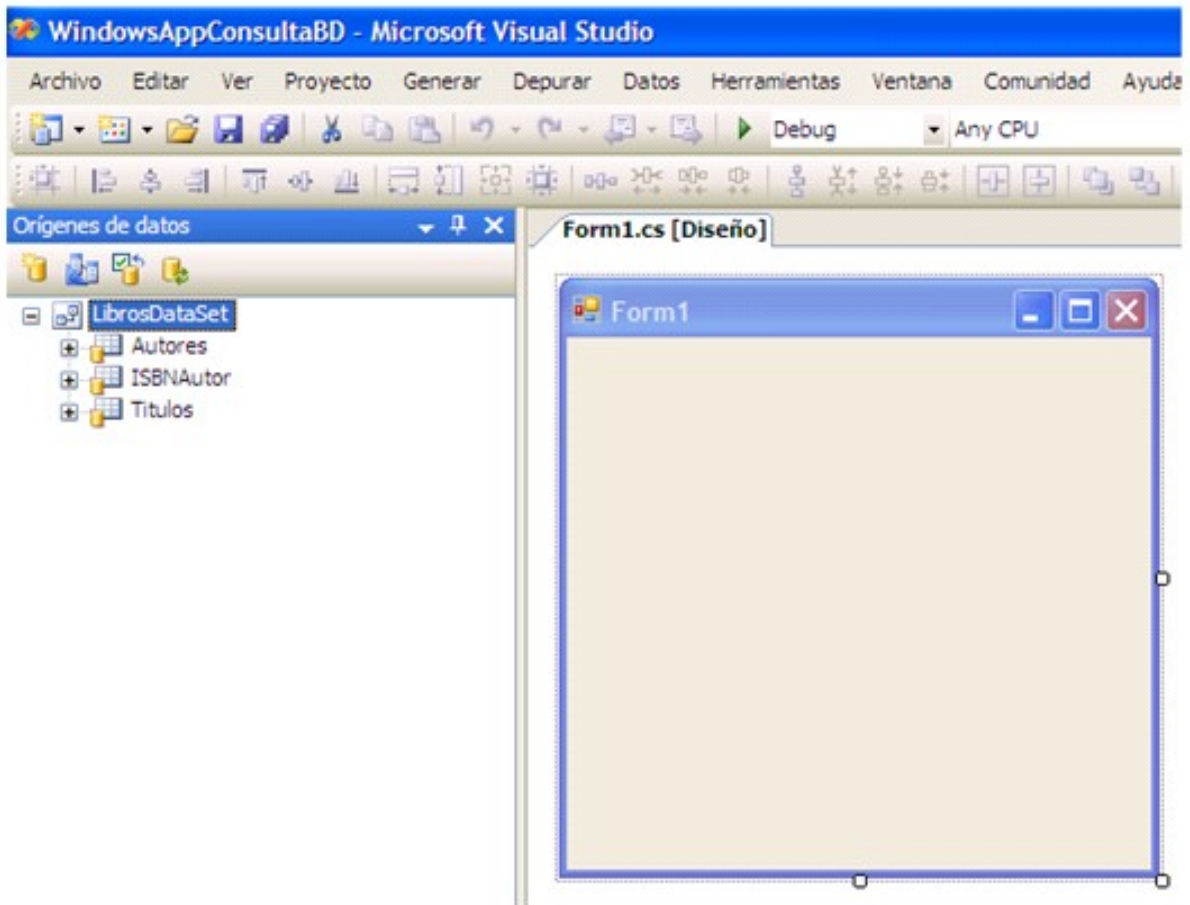


Figura 2

2. arrastrar y soltar el mouse en el formulario

Automáticamente se agrega al formulario un control DataGridView (TitulosDataGridView), con 4 columnas cuyos títulos son los nombres de los campos de la tabla Titulos como muestra la Figura 3. También se agregan al formulario 4 controles más:

- LibrosDataSet
- TitulosBindingSource
- TitulosTableAdapter
- TitulosBindingNavigator

Los 3 primeros controles no tienen una interface gráfica. El control TitulosBindingNavigator es el que aparece en la parte superior del formulario que contiene botones para ir al primero, siguiente, anterior o último registro de la tabla Titulos y también para agregar, eliminar un registro y grabar las modificaciones.

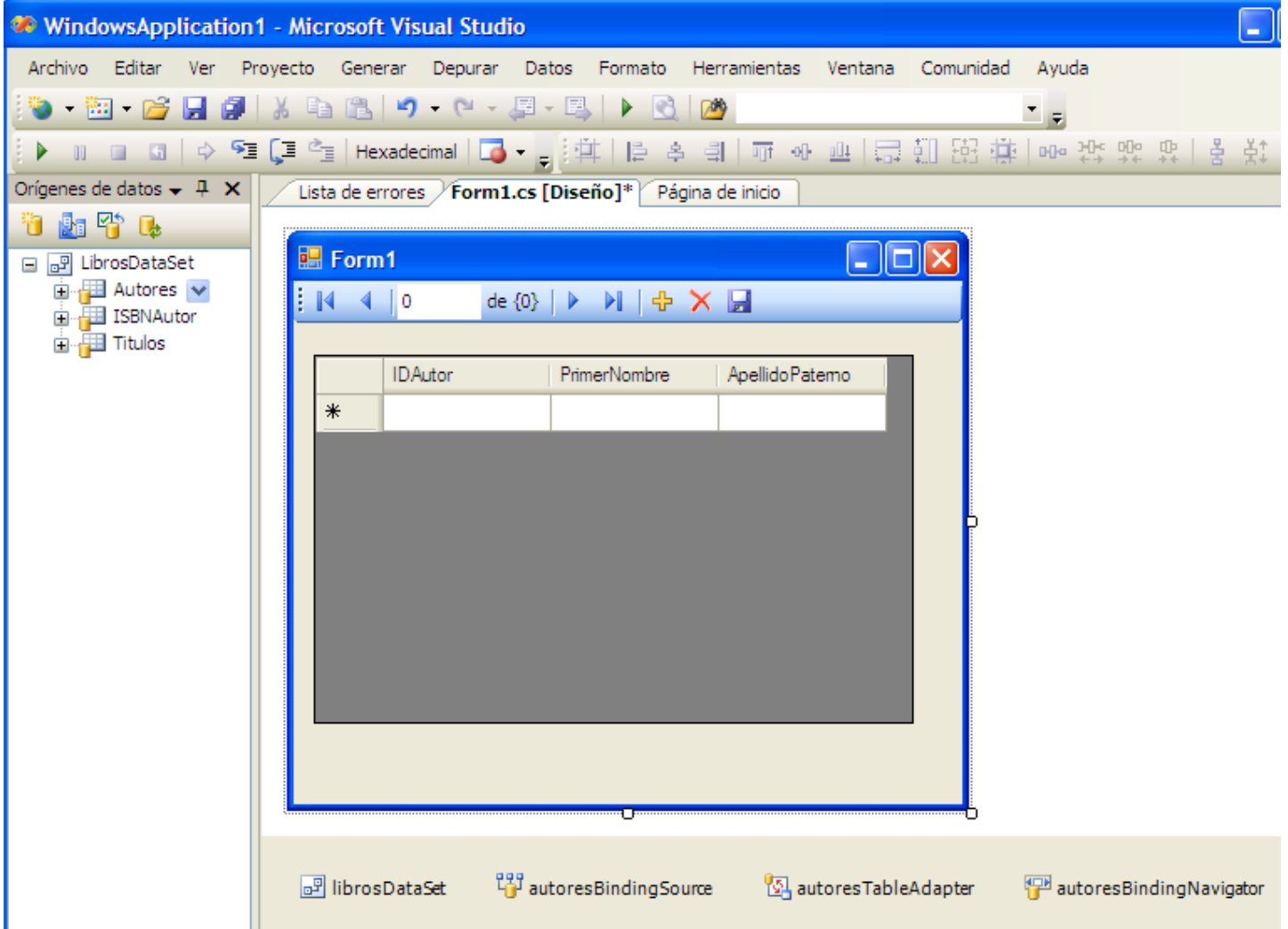


Figura 3

Si corremos la aplicación podemos observar los datos de los Autores en el DataGridView

Haciendo click en los controles contenidos en el autoresBindingNavigator podemos avanzar, retroceder, eliminar un registro, agregar un registro en la tabla de Autores y luego grabar las modificaciones.

En la figura 4 se muestra la aplicación en ejecución, con los datos de la tabla Autores en el DataGridView

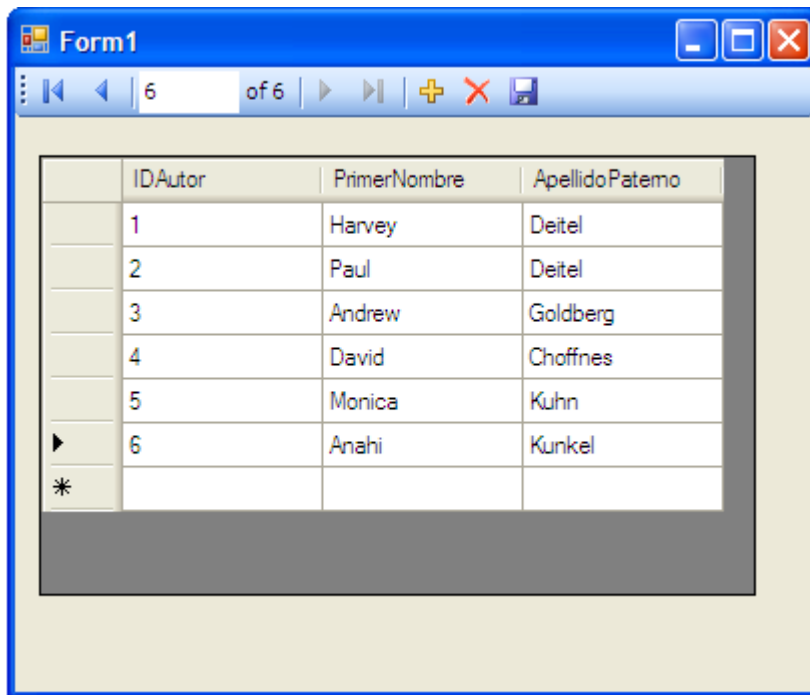


Figura 4

3. Mostrar en el formulario una consulta con parámetros que involucra varias tablas de la Base de Datos SQL Server

Supongamos que queremos mostrar en el DataGridView los títulos de libros de un determinado Autor cuyo IDAutor se ingresa en tiempo de ejecución, como lo muestra la figura 5

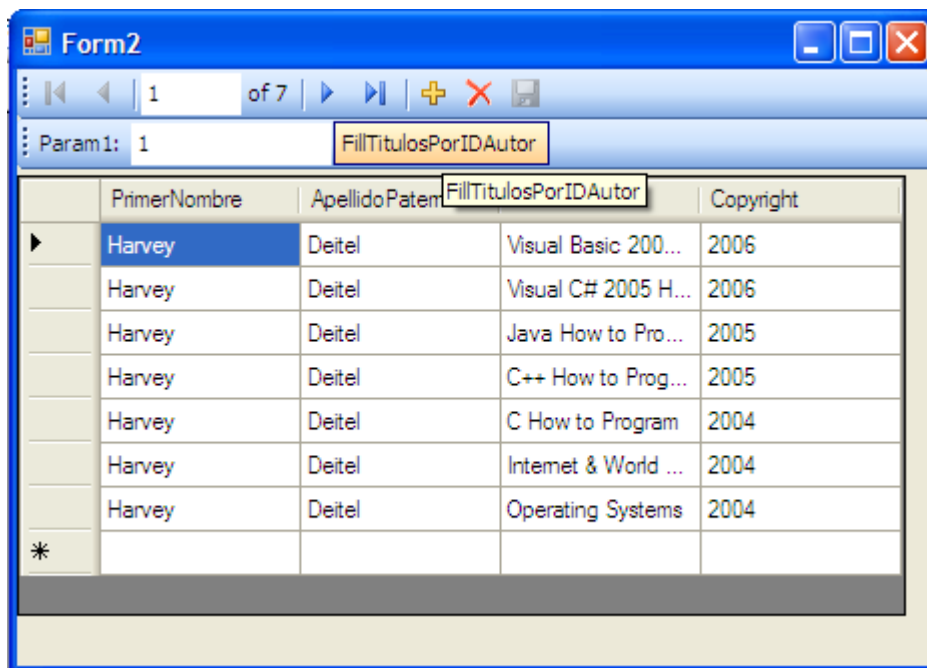


Figura 5

Entonces seguimos los siguientes pasos:

1. mouse derecho sobre LibrosDataSet en la ventana de Origenes de Datos
2. elegir Editar DataSet con el Diseñador
3. mouse derecho sobre una zona libre del Diseñador
4. elegir agregar ► TableAdapter ...
5. click en siguiente en la ventana del Asistente para la configuración del TableAdapter donde por default esta seleccionado LibrosConnectionString para la conexión de datos
6. click en siguiente donde por default está seleccionado Usar instrucciones SQL
7. click en Generador de consultas ...
8. click en agregar tres veces para agregar las 3 tablas (Autores, ISBNAutor, Titulos)
9. tildamos en las tablas los campos IDAutor, PrimerNombre, ApellidoPaterno, Titulo, Copyright
10. destildamos el campo IDAutor en la columna Resultados de la grilla
11. escribimos ? en la celda de la columna Filtro del campo IDAutor
12. click en Ejecutar consulta quedando el siguiente código para la consulta

```
SELECT      Autores.PrimerNombre, Autores.ApellidoPaterno,
            Titulos.Titulo, Titulos.Copyright

FROM        Autores
            INNER JOIN
                ISBNAutor ON Autores.IDAutor = ISBNAutor.IDAutor
            INNER JOIN  Titulos ON ISBNAutor.ISBN = Titulos.ISBN

WHERE       (Autores.IDAutor = @Param1)
```

13. ingresar 1
14. Aceptar (ver figura 6)
15. Aceptar para volver al asistente de configuración de TableAdapter
16. Siguiente
17. cambiar el nombre de los métodos Fill y Get por FillTitulosPorIDAutor y GetDataTitulosPorIDAutor
18. Finalizar

19. click en el nombre de DataTable1 para cambiarlo por DataTableTitulosPorIDAutor que aparece en el Editor del Diseñador
20. agregar un nuevo formulario Form2 a la aplicación desde el explorador de Soluciones
21. ir a la ventana de Origenes de Datos
22. mouse izq sin soltar sobre DataTableTitulosPorIDAutor
23. arrastrar y soltar sobre el Form2
24. correr la aplicación desde el Form2 escribiendo en el código del archivo Program.cs


```
Application.Run(new Form2());
```
25. ingresar 1 para IDAutor en param1
26. click en FillTitulosPorIDAutor (ver Figura 5)

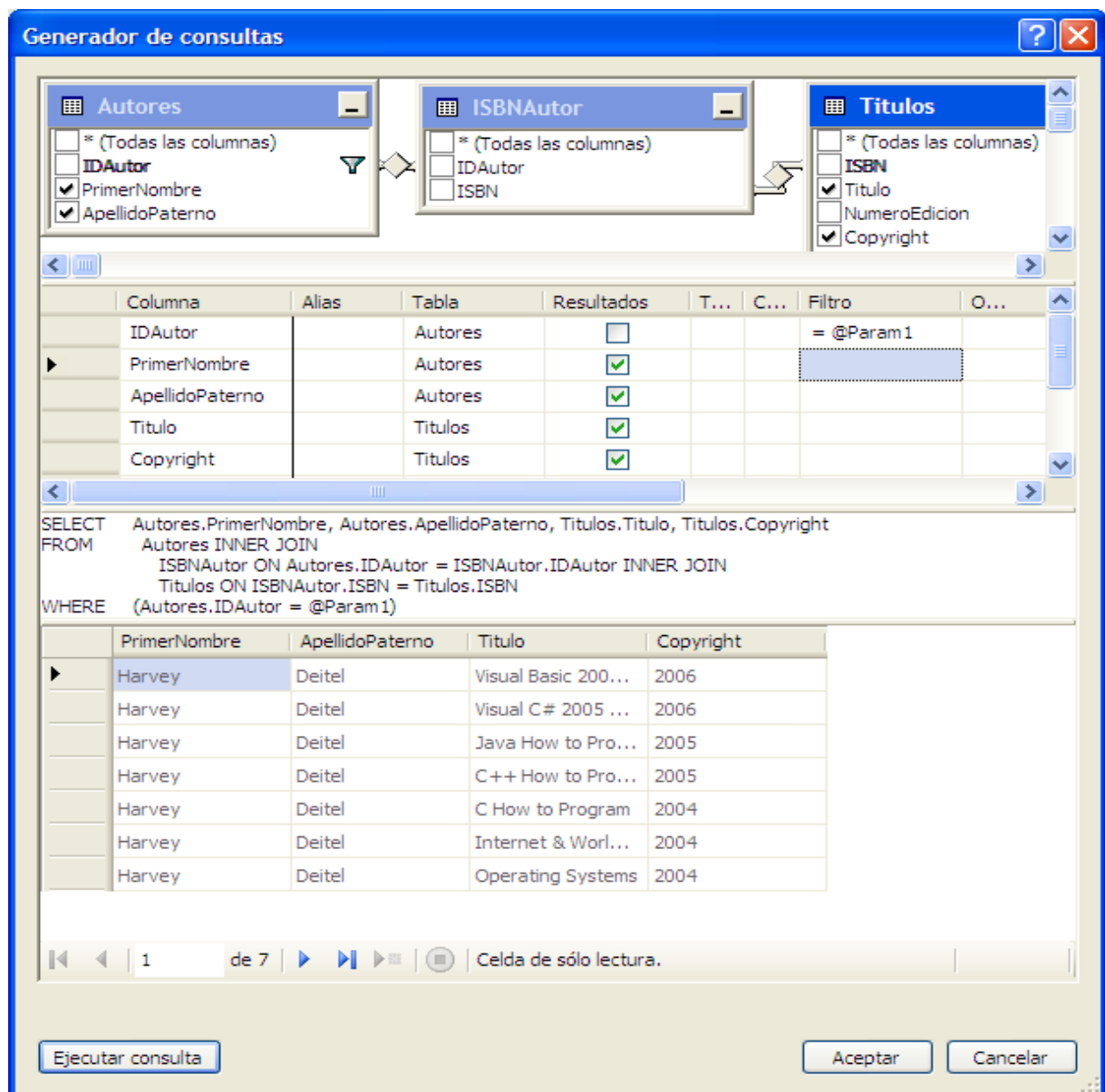


Figura 6

4. Mostrar en el formulario un ComboBox con los datos de varios campos de una tabla de una BD de SQL Server

Si queremos mostrar en un ComboBox los datos de varios campos de por ej. la tabla Autores como muestra la figura 7

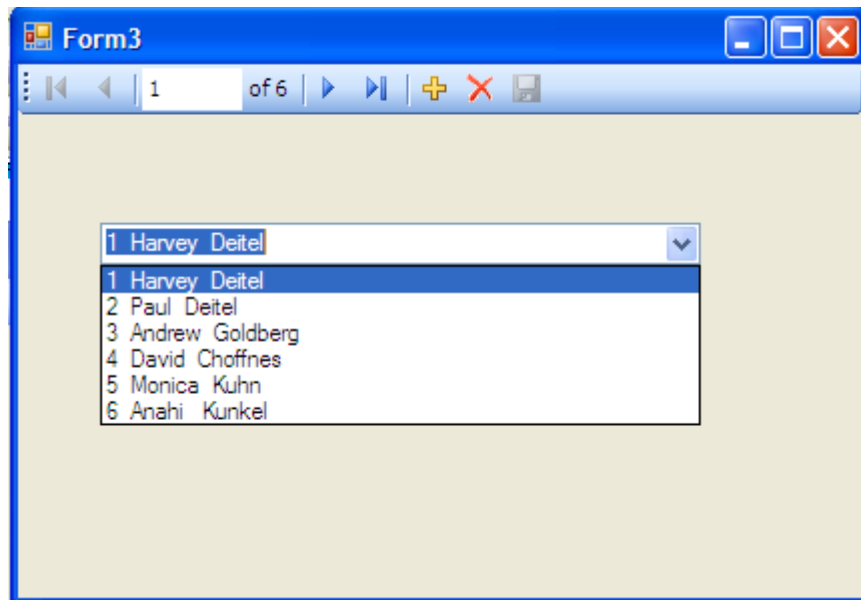


Figura 7

procedemos de la siguiente forma:

1. Agregamos un nuevo formulario Form3 desde el explorador de soluciones
2. procedemos igual que en el punto 3 desde los pasos 1 a 6
3. escribimos la siguiente instrucción SQL

```
SELECT
```

```
CAST(IDAutor AS varchar(5)) + ' ' + PrimerNombre + ' ' + ApellidoPaterno  
AS expr1, IDAutor
```

```
FROM Autores
```

4. Siguiendo
5. cambiar el nombre de los métodos Fill y Get por FillIDNomApAutores y GetDataIDNomApAutores
6. Finalizar
7. click en el nombre de DataTable1 para cambiarlo por DataTableIDNomApAutores que aparece en el Editor del Diseñador

8. ir a la ventana de Origenes de Datos
9. mouse der sobre DataTableIDNomApAutores y en el menú desplegable seleccionar ComboBox. Si no figura como opción ir a personalizar ... tildar ComboBox y aceptar
10. mouse izq sin soltar sobre DataTableIDNomApAutores
11. arrastrar y soltar sobre el Form3
12. correr la aplicación desde el Form3 escribiendo en el código del archivo Program.cs

`Application.Run(new Form3());`

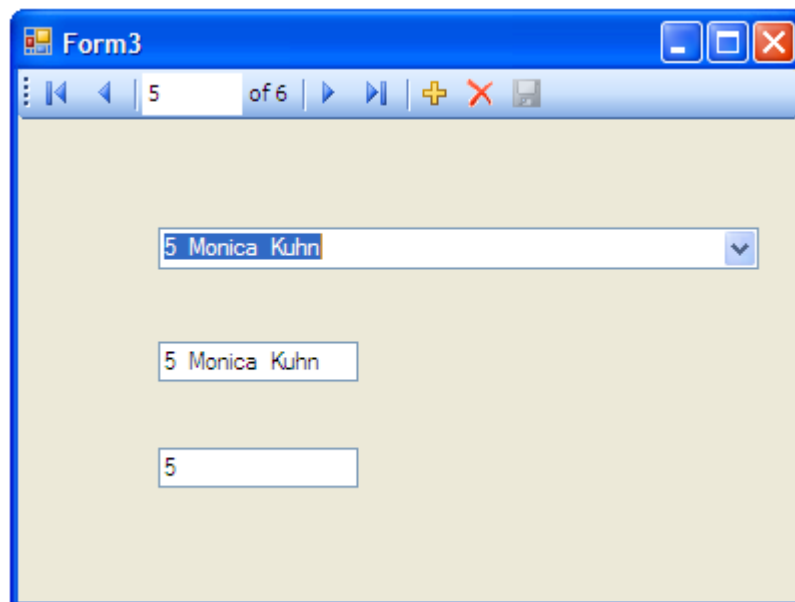


Figura 8

En la figura 8 se ha agregado al Form3 dos textbox
 El textBox1 muestra el texto seleccionado en el ComboBox y
 el textBox2 muestra el IDAutor asociado
 En el método que se ejecuta cuando se produce el evento de cambio de selección es el que sigue:

```
private void dataTableIDNomApAutoresComboBox_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        textBox1.Text = ((DataRowView)dataTableIDNomApAutoresComboBox.SelectedItem)["expr1"].ToString();
        textBox2.Text = ((DataRowView)dataTableIDNomApAutoresComboBox.SelectedItem)["IDAutor"].ToString();
    }
    catch (NullReferenceException ee)
    {
    }
}
```

Para recuperar el texto de la opción elegida en el ComboBox sería:

```
((DataRowView)dataTableIDNomApAutoresComboBox.SelectedItem)["expr1"].ToString();
```

Para recuperar el IDAutor de la opción seleccionada en el ComboBox sería:

```
((DataRowView)dataTableIDNomApAutoresComboBox.SelectedItem)["IDAutor"].ToString();
```