

Accediendo a los Datos con ADO.NET (cap5)

En este capítulo veremos que es ADO.NET y algunas de sus clases más importantes, crearemos una Base de Datos y una tabla y haremos una conexión a una Base de Datos. Las pruebas serán con Visual Studio .NET y con Web Matrix.

5. 1. - Acceso a Datos con ADO.NET

En esta sección veremos ADO.NET y sus clases más importantes

5. 1. 1. - ¿Qué es ADO.NET?

¿Qué es ADO.NET?

ADO.NET. Provee un conjunto de clases para trabajar con datos. ADO.NET permite:

- Es un sucesor del modelo ADO anterior.
- Esta pensado para entornos desconectados
- Es un modelo de programación avanzado con soporte de XML
- Es un conjunto de clases, interfaces, estructuras y enumeraciones para administrar el acceso a datos dentro de Marco de Trabajo .NET

ADO.NET es una tecnología de acceso a datos que se basa en los objetos ADO (Objetos de Datos ActiveX) anteriores. Es una manera nueva de acceder a los datos construida sobre ADO. ADO.NET puede coexistir con ADO. Aunque usted no conozca ADO, en este capítulo va a poder usar ADO.NET y verá con ejemplos, como conectarse a una Base de Datos.

ADO.NET utiliza un **modelo de acceso pensado para entornos desconectados**. Esto quiere decir que la aplicación se conecta al origen de datos, hace lo que tiene que hacer, por ejemplo seleccionar registros, los carga en memoria y se desconecta del origen de datos.

ADO.NET es un conjunto de clases que usted utiliza **para acceder y manipular orígenes de datos** como por ejemplo, una base de datos en SQL Server o una planilla Excel.

ADO.NET utiliza XML como el formato para transmitir datos desde y hacia su base de datos y su aplicación Web.

5. 1. 2. - Usando el Espacio de Nombres

Usando espacio de nombres

- Use la sentencia Imports para importar el espacio de nombres

```
Imports System.Data
Imports System.Data.SqlClient
```

- Los espacios de nombre usados con ADO.NET incluyen:
 - System.Data
 - System.Data.SqlClient
 - System.Data.OleDb

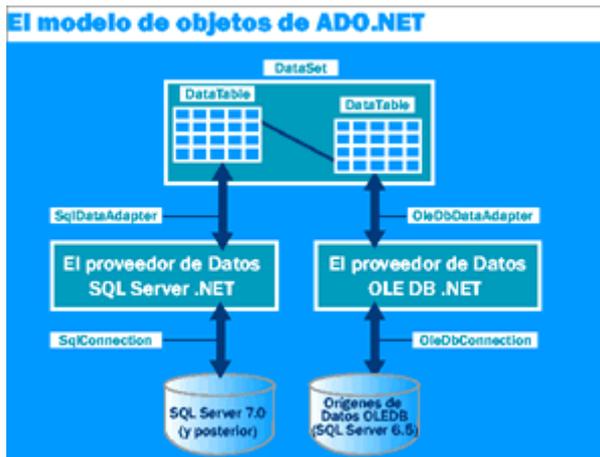
Hay 3 espacios de nombres que usted va a importar en un formulario Web si esta usando ADO.NET:

- **System.Data.**
- **System.Data.SqlClient.**
- **System.Data.OleDb.**

Recuerde la sentencia que usa VB.NET para importar el espacio de nombres:

```
Imports System.Data           'Siempre lo va a utilizar
Imports System.Data.SqlClient 'Para acceder a SQL Server
Imports System.Data.OleDb     'Para cualquier origen de datos
```

5. 1. 3. - El modelo de Objetos ADO.NET



Haga clic en la imagen para expandir

El modelo de objetos ADO.NET provee una estructura de acceso a distintos orígenes de datos. Tiene 2 componentes principales: El DataSet y el proveedor de Datos .NET

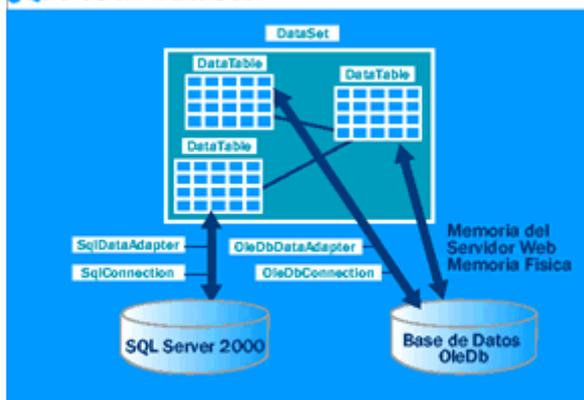
- El **DataSet**: Esta formado por uno o más objetos de tipo **DataTables**. fue pensado para acceder a datos independientemente del origen. Por ejemplo, un DataSet puede obtener datos de SQL Server, Oracle o de un archivo XML. Puede utilizar un objeto llamada DataView para ver los datos de distintas maneras.
- **El proveedor de Datos .NET**: Provee del enlace entre el Origen de Datos y el DataSet.

El cuadro muestra un ejemplo de Objetos provistos por distintos proveedores de datos .NET

Objetos de Proveedores de Datos .NET	Propósito	Objeto SQL Server 7.0 o 2000	Objeto para un origen OLEDB
Connection	Provee conectividad a un Origen de Datos	SqlConnection	OleDbConnection
Command	Provee acceso a comandos de Base de Datos como Select, Delete, Insert y Update	SqlCommand	OleDbCommand
DataReader	Provee acceso a datos de solo lectura	SqlDataReader	OleDbDataReader
DataAdapter	Utiliza el objeto Connection para enlazar un objeto DataSet con un Proveedor de Datos. También permite actualizar los Datos en el origen a partir de las modificaciones hechas en el DataSet.	SqlDataAdapter	OleDbDataAdapter

5. 1. 4. - ¿Qué es un DataSet?

¿Qué es un DataSet?



Haga clic en la imagen para expandir

Un DataSet guarda información en un entorno desconectado. Después de que usted establece una conexión con una Base de Datos entonces puede acceder a sus datos.

El DataSet es la principal forma de guardar datos cuando usted utiliza ADO.NET

El DataSet le permite a usted guardar datos que se obtuvieron de un origen de datos. **Los datos en un DataSet pueden ser manipulados sin necesidad que el formulario Web mantenga la conexión con el origen de datos.** La conexión se reestablece recién cuando usted necesita actualizar los cambios

El DataSet guarda la información en uno o más objetos de tipo DataTables. Cada DataTable puede ser poblada con datos de un único origen. Usted puede también establecer relaciones entre 2 objetos DataTables usando un objeto de tipo DataRelation.

5. 1. 5. - Accediendo a Datos con ADO.NET

Hay típicamente 3 pasos para acceder a los datos.

- Acceder al Origen de Datos y mostrar los datos en el formulario Web
- Manipular los Datos
- Retornar los datos para actualizar la base de datos.

Observe en el gráfico el escenario más común que se presenta cuando accede a los datos:



Haga clic en la imagen para expandir

Los datos siempre viajan en formato XML ASP.NET y ADO.NET transforman en forma automática XML en un DataSet.

5. 2. - Creando una Conexión a una Base de Datos

En esta sección veremos como establecer una conexión a una Base de Datos usando Visual Studio .NET y Web Matrix.

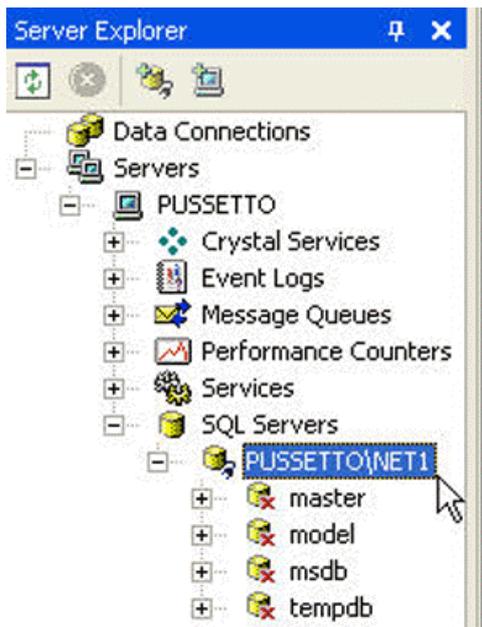
5. 2. 1. - Creando una Base de Datos y una Tabla con Visual Studio .NET.

El entorno de desarrollo provee de herramientas para hacer operaciones con un Origen de Datos tales como crear una Base de datos, crear una Tabla o Procedimientos Almacenados.

Vamos a ver como puede utilizar una herramienta llamada el Explorador de Servidores.

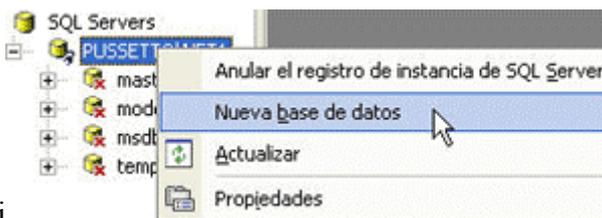
- Abra Visual Studio .NET con el proyecto llamado 'CursoVS'.
- Vaya a **Ver, Explorador de Servidores.**
- Haga clic en el símbolo + dentro de la estructura jerárquica para poder ver el servidor MSDE que tiene instalado en su máquina

El nombre del servidor es 'nombre de Máquina / Nombre de Instancia'

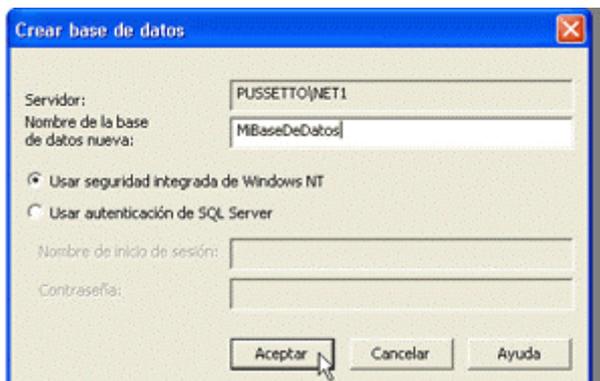


Vamos a crear una Base de Datos en el Servidor:

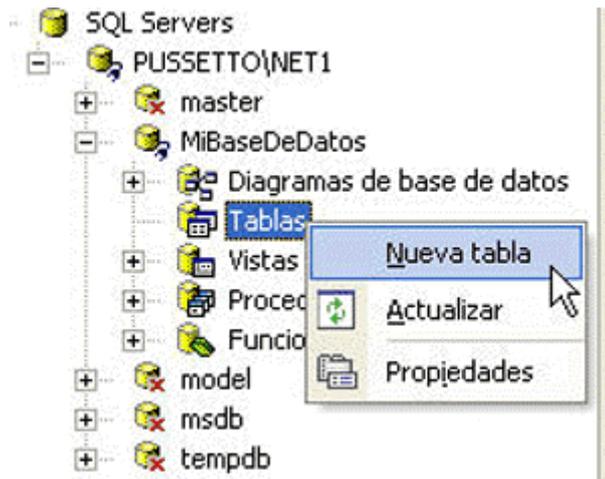
- Haga clic en el nombre de su servidor. Presione el botón derecho del Mouse. Seleccione **'Nueva Base de Datos'**



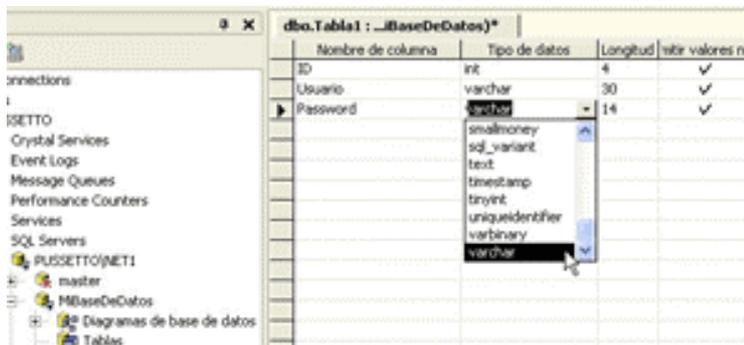
- En el campo que dice **'Nombre de la Base de Datos Nueva'** escriba **'MiBaseDeDatos'**. Presione **Aceptar**.



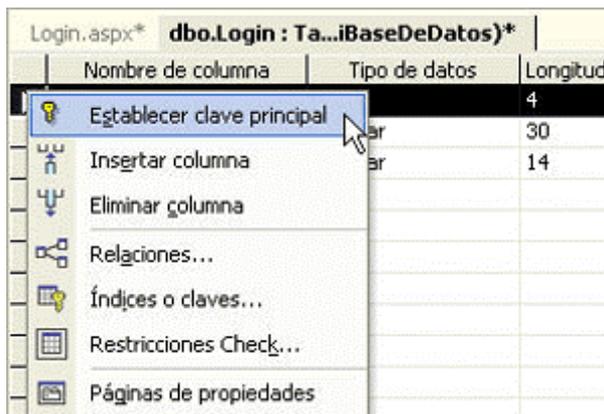
- Ahora vamos a crear una Tabla. Expanda la estructura y Haga clic con el botón derecho en el objeto llamado **'Tables'**. Va a ver la ventana de la figura.



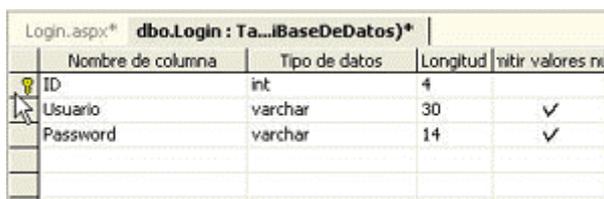
- Haga clic en **Nueva Tabla**.
- Escriba los nombres de las columnas, el tipo de datos y la longitud tal cual aparece en la imagen



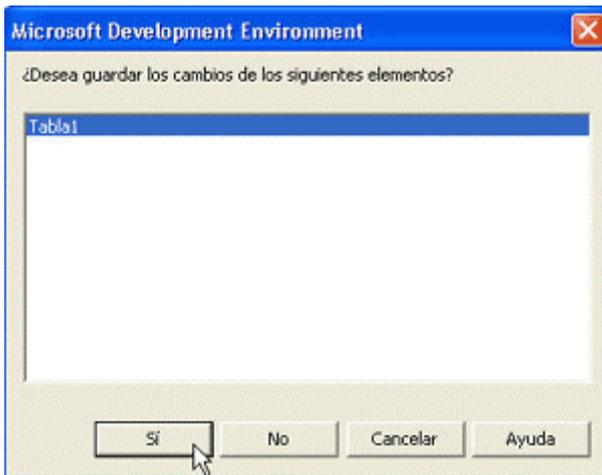
- Vamos a establecer a ID como clave principal de la tabla. Haga clic sobre la fila donde esta definido el campo ID. Presione el botón derecho del Mouse. Seleccione Establecer clave principal.



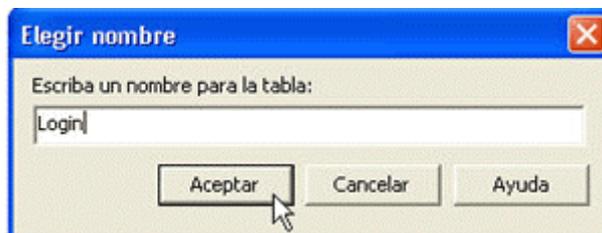
- Va a aparecer una llave amarilla al costado izquierdo del campo **ID**. Esto indica que es la clave principal de la tabla.



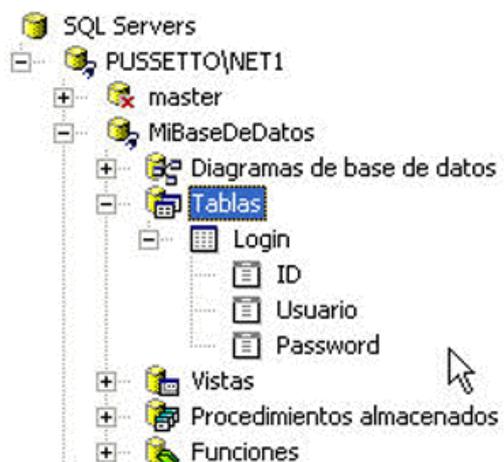
- Cierre la ventana. Va a aparecer el siguiente cuadro. Haga clic en **Si**.



- Escriba **'Login'** como nombre de tabla.
- Haga clic en **Aceptar**.



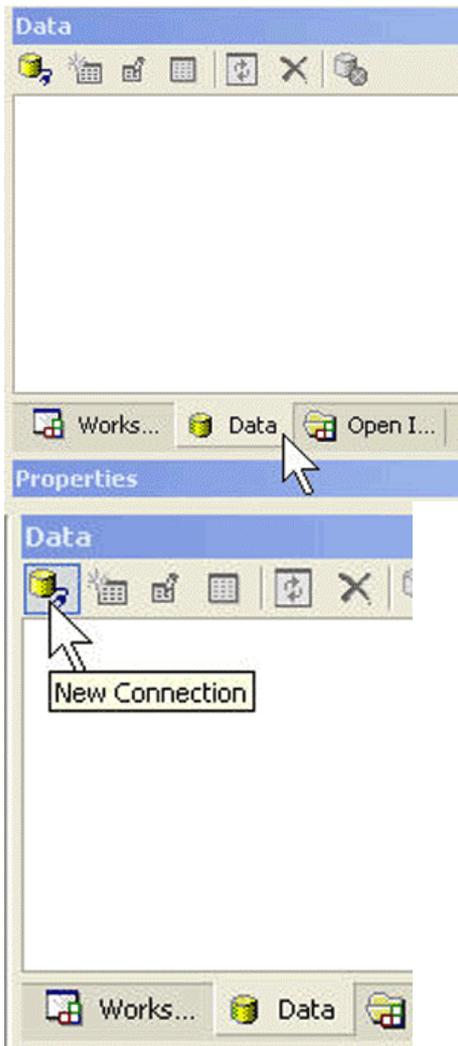
- Como resultado, puede ver su nueva **Base de Datos** y la Tabla **Login**.



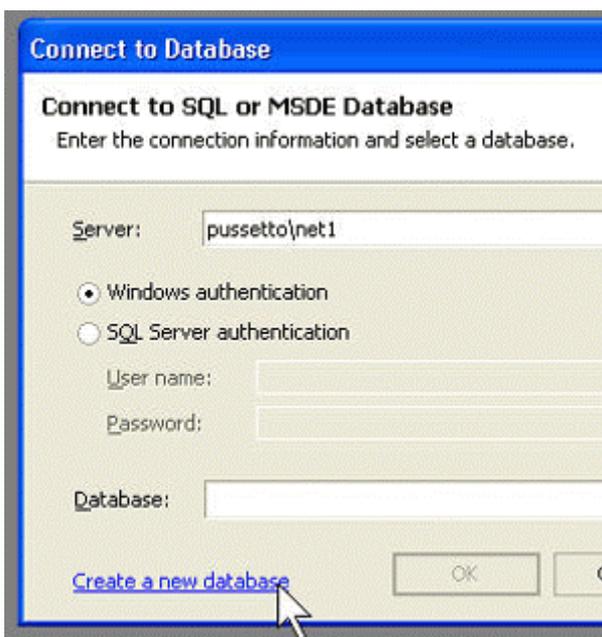
5. 2. 2. - Creando una Base de Datos y una Tabla con Web Matrix

Vamos a crear una Base de Datos llamada Doctores y una tabla también llamada Doctores con Web Matrix

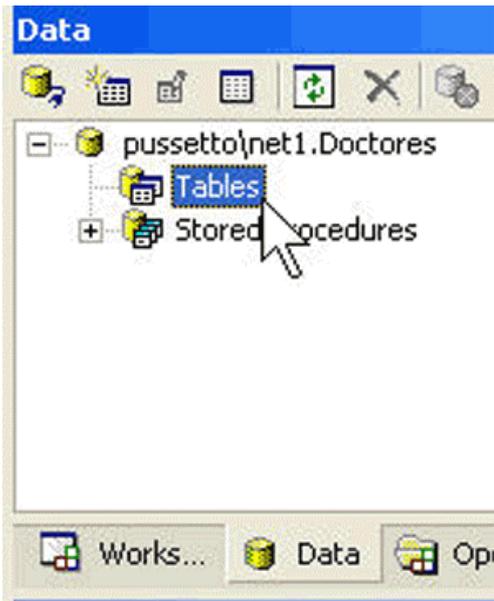
- Abra **Web Matrix**
- En la ventana llamada **'WorkSpace'**, seleccione el tab Data
- Haga clic en el icono que dice **'New Connection'**



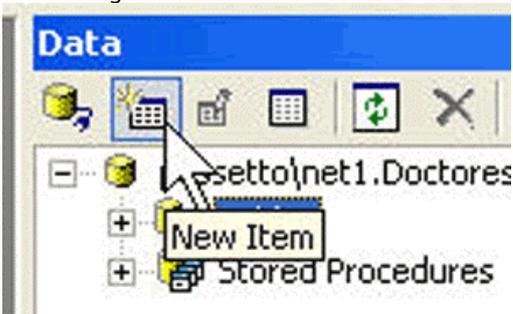
- Escriba el nombre de su servidor SQL. Recuerde que es Nombre de Maquina + NET1.
- Haga clic en el enlace Azul que dice **'Create a new database'**.



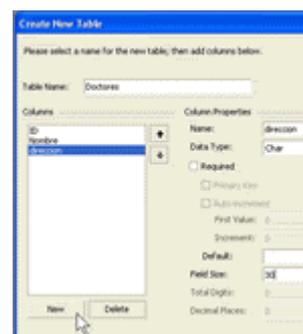
- Haga clic donde dice **'Tables'**



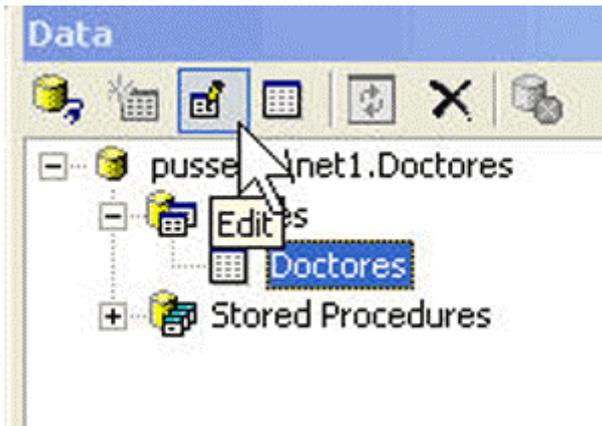
- Haga clic donde dice **'New Item'**



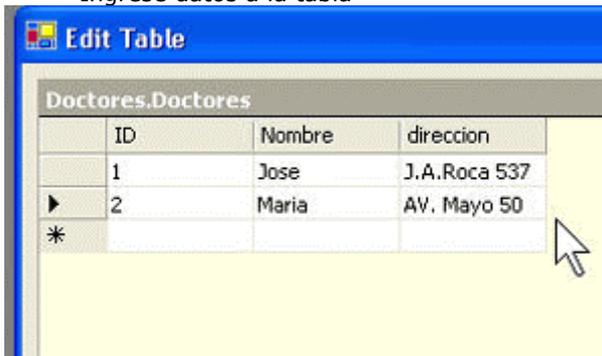
- Donde dice **'Table Name'** escriba **'Doctores'**
- Haga clic en el botón **'New'**
- Complete el campo Nombre como aparece en la figura
- Complete el campo Dirección como aparece en la figura



- Ahora vamos a agregar datos a la tabla. Haga clic en la tabla **Doctores** y Haga clic en el icono que dice **'Edit'**



- Ingrese datos a la tabla



- Cierre la ventana.

Hemos creado una Base de Datos llamada '**Doctores**' y una tabla llamada '**Doctores**' usando Web Matrix.

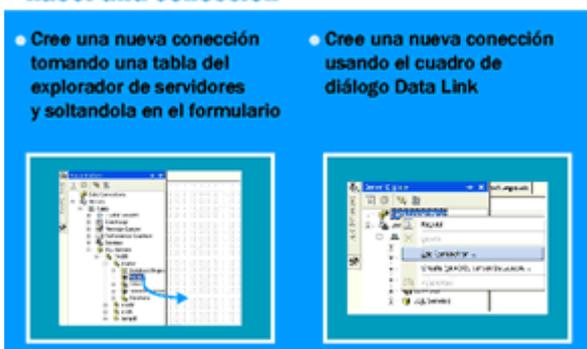
5. 2. 3 .a. Usando el Explorador de Servidores para establecer una conexión en Visual Studio.NET

Vamos ahora a ver 2 maneras de establecer una conexión con una Base de Datos con Visual Studio .NET.

1. Creando una Conexión desde el objeto Servers:

- Vaya al explorador de soluciones.
- Haga Doble click en el formulario llamado Login.
- Vaya al Explorador de Servidores

Usando el explorador de Servidores para hacer una conexión

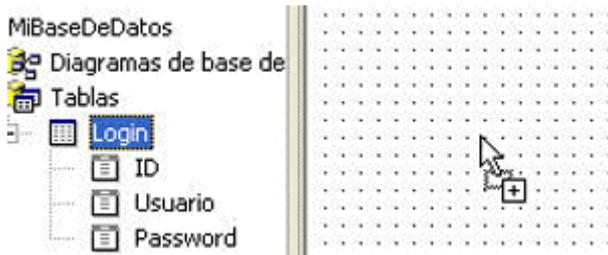


● Cree una nueva conexión tomando una tabla del explorador de servidores y soltandola en el formulario

● Cree una nueva conexión usando el cuadro de diálogo Data Link

- Ahora vamos a Tomar y Soltar la tabla de '**Logins**'.
- Presione el botón izquierdo del Mouse sobre la tabla **Login**.

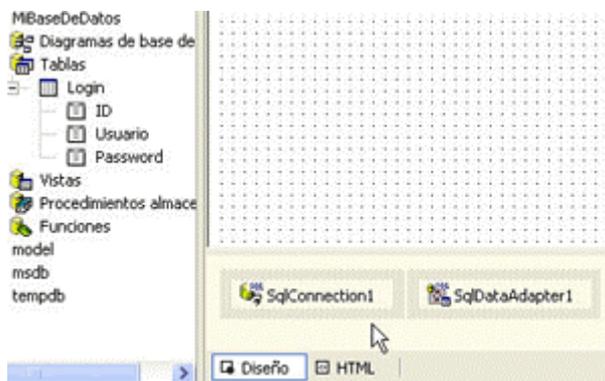
- Sin dejar de apretar el botón izquierdo, mueva el Mouse sobre el formulario.
- Suelte el botón izquierdo del Mouse.



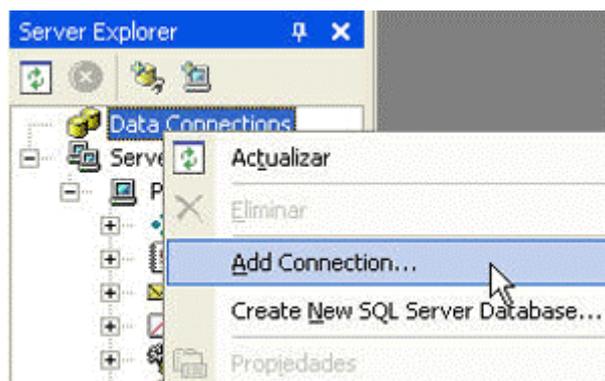
- Como resultado va a ver 2 controles visuales en la parte inferior del formulario. Uno de ellos es **SqlConnection**. El objeto visual para establecer la conexión con la base de Datos
- Para poder probar otra opción vamos a eliminar los 2 objetos que acabamos de agregar. Haga clic en SqlConnection1 y presione la tecla suprimir del su teclado.

2. Creando una Conexión con la ventana de dialogo 'Data Link' (Enlace de Datos):

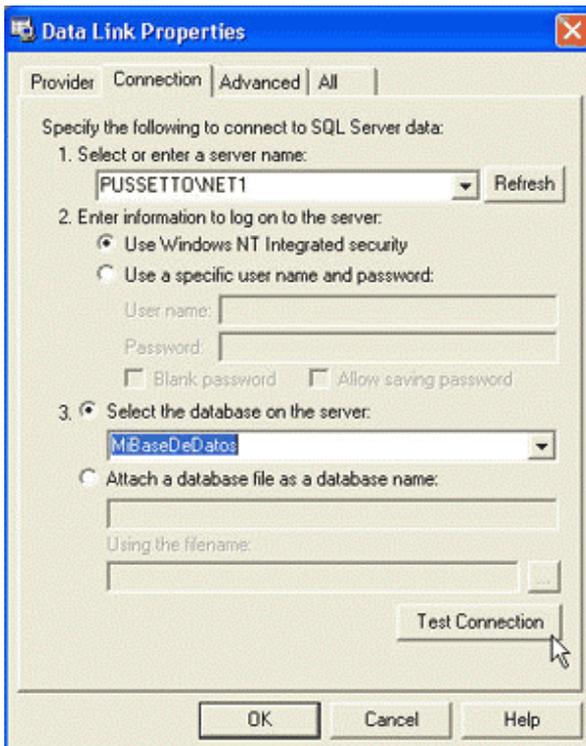
- Vaya al explorador de Servidores. Haga clic donde dice '**Data Connections**'.



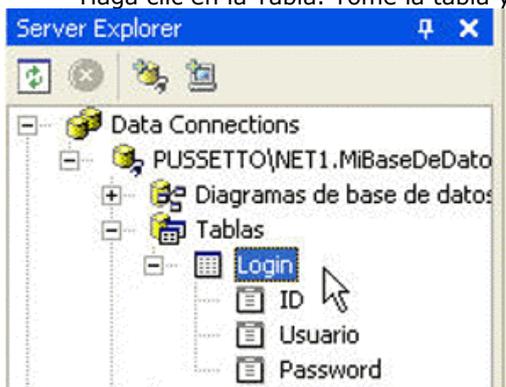
- Presione el botón derecho del Mouse. Seleccione '**Add Connection...**'



- Donde dice 1, seleccione el nombre de su servidor.
- Donde dice 2, Seleccione la primera opción.
- Donde dice 3, seleccione el nombre de la Base de Datos '**MiBaseDeDatos**'
- Haga clic en '**Test Connection**' para testear la conexión
- Haga clic en OK
- Expanda la conexión tal como se ve en la imagen

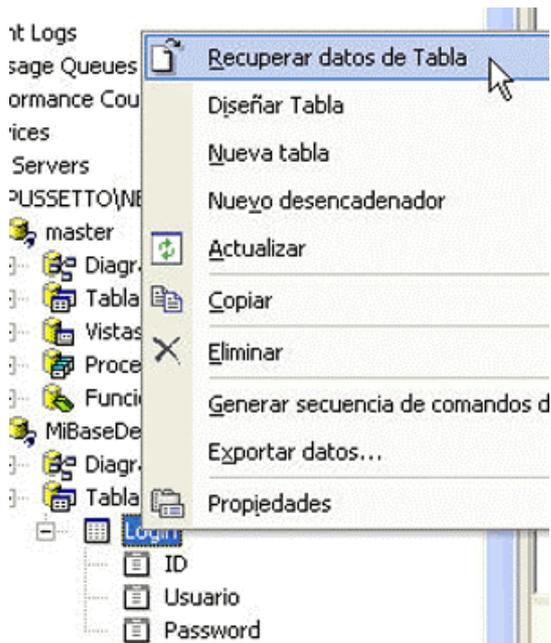


- Haga clic en la Tabla. Tome la tabla y suéltela sobre el formulario tal como lo hizo antes.



Ahora vamos a cargar algunos datos en la tabla

- Presione el botón derecho del Mouse sobre la tabla '**Login**'.
- Selecciones '**Recuperar datos de Tabla**'

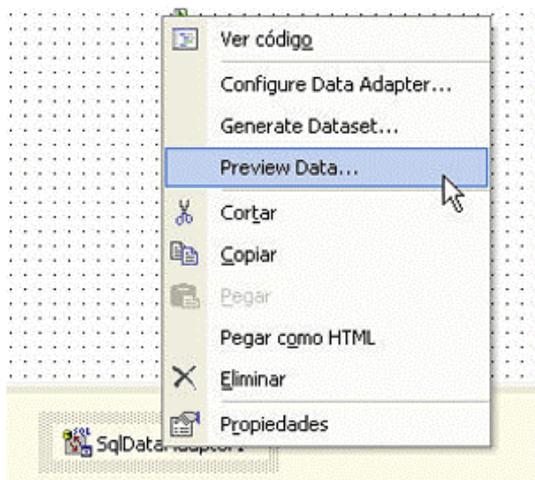


- Llene los campos de la tabla
- Cierre la tabla.

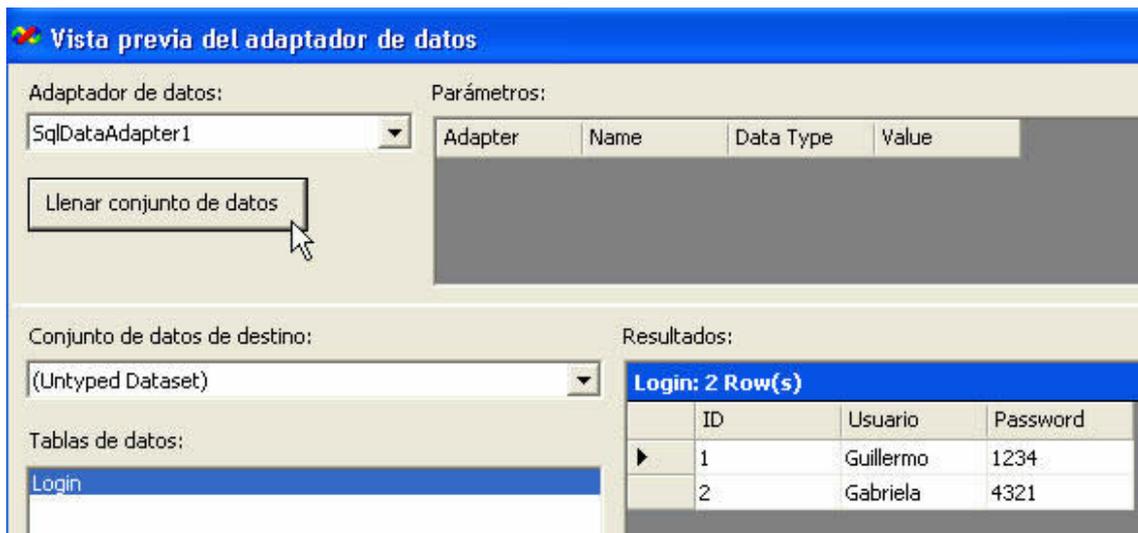
ID	Usuario	Password
1	Guillermo	1234
2	Gabriela	4321
*		

Finalmente, vamos a ver los datos.

- Haga clic con el botón derecho del Mouse en el objeto **SqlDataAdapter1**.
- Seleccione '**Preview Data**'



- Va a ver la imagen siguiente. Haga clic en '**Llenar conjunto de datos**'. Estos son los datos de la tabla **Login**. Al hacer click esta cargando un **DataSet**.



- Haga clic en Cerrar.

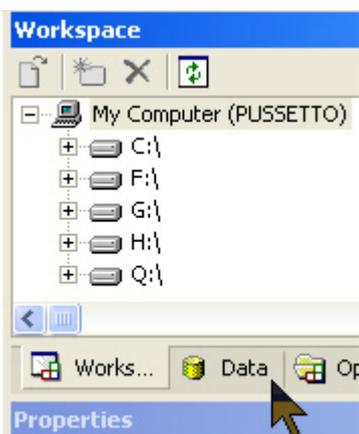
En resumen, hemos visto 2 maneras distintas de establecer una conexión con una Base de Datos en un Formulario Web

5. 2. 3. b. Práctica: Usando la ventana del Espacio de Trabajo (Workspace) para establecer una conexión a un Base de Datos en Web Matrix

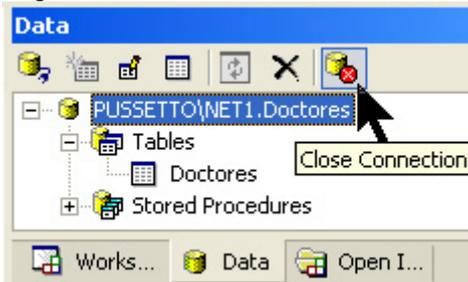
En esta sección vamos a ver como establecer una conexión con la base de datos 'Doctores que creamos en la práctica anterior 5. 2. 2. - Creando una Base de Datos y una Tabla con Web Matrix.

Primero vamos a cerrar la conexión con la Base de datos que creamos en la práctica anterior.

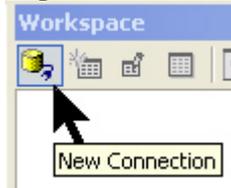
1. Vaya a la ventana llamada **'Workspace'** (espacio de trabajo).
2. Haga clic en el tab **'Data'**.



3. Haga clic sobre el nombre de su servidor SQL.
4. Haga clic en el icono **'Close connection'** (cerrar conexión).



- Haga clic en el icono **'New Connection'** (Nueva Conexión).



- En la casilla que dice **'Server'**, Escriba nombre de máquina + \ + nombre de instancia de SQL (si la tiene).
- En la casilla que dice **'Database'**, escriba 'Doctores' que es la base de datos que creamos en la práctica anterior 5. 2. 2. - Creando una Base de Datos y una Tabla con Web Matrix.

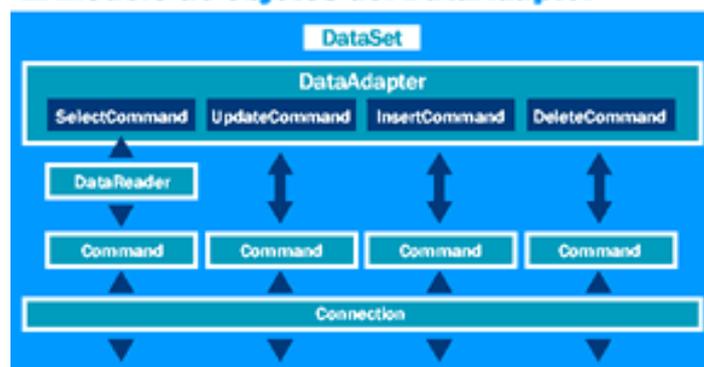


En esta práctica hemos visto como cerrar y como establecer una conexión con una Base de Datos usando Web Matrix.

5. 2. 4. - El modelo de objetos del DataAdapter

En la imagen puede ver como el DataAdapter hace de nexo entre una Base de Datos y el DataSet. El DataReader permite ver de distintas maneras los datos seleccionados de la Base de Datos. También se observa como el DataAdapter permite hacer las operaciones más comunes en una Base de Datos como por ejemplo, Seleccionar, Modificar, Borrar e Insertar elementos.

El modelo de objetos del DataAdapter



5. 2. 5. - Como Generar un DataSet con código

Usted puede crear un DataSet con código.

Con código debe primero declarar y crear un objeto de tipo DataSet

```
Dim ds As New DataSet()
```

Luego debe cargar el DataSet a partir de algún DataAdapter

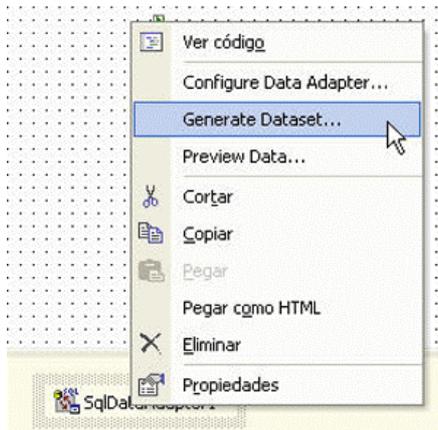
- DataAdapter1.fill(ds)

Eso es lo que hace el método fill del DataAdapter

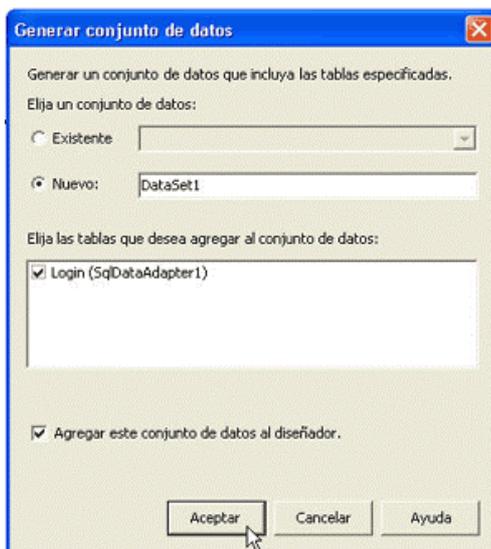
5. 2. 6. - Como Generar un DataSet con la Interfase Visual con Visual Studio .NET

Vamos a crear un DataSet.

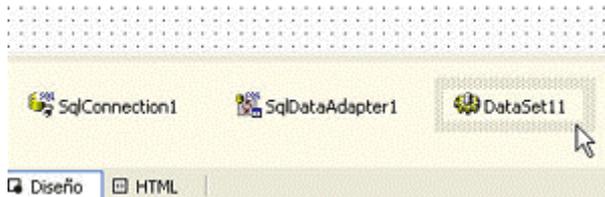
- Haga clic con el botón derecho del Mouse en el objeto **SqlDataAdapter1**.
- Seleccione '**Generate DataSet...**'



- Va a ver la pantalla que aparece en la imagen
- Al hacer click en **Aceptar** se crea un **DataSet** llamado DataSet1, con los datos de la tabla **Login** y agrega el objeto visual DataSet1 al formulario.



- En la imagen puede ver el control agregado.



5. 3. - Mostrando un Dataset en un control asociados a listas

5. 3. 1. - ¿Qué son los controles Asociados a Listas?

¿Qué son los Controles Asociados a Listas?

- Son controles que se conectan a un origen de datos y los muestran
- Incluyen los siguientes controles:
 - DropDownList
 - ListBox
 - CheckBoxList
 - RadioButtonList
 - DataGrid
 - DataList
 - Repeater

Hay dos tipos de controles asociados a datos. Los controles asociados simples como por ejemplo una casilla de texto y los controles asociados a listas. En la imagen tiene algunos de los controles asociados a listas.

Los controles asociados a listas son controles que se conectan con un origen de datos y luego los muestran. Ejemplo de esto es una grilla en un formulario ASP.NET

5. 3. 2. - Mostrando un Dataset en un control asociado a listas

Mostrando datos de un DataSet en los controles asociados a listas

• Configure las propiedades

Propiedad	Descripción
DataSource	El DataSet que contiene los datos
DataMember	El DataTable en el DataSet
DataTextField	El campo en el DataTable que se muestra
DataValueField	El campo en el DataTable que se convierte en el valor del ítem seleccionado en la lista

• Use el DataSet, entonces llame al método DataBind

```
DataAdapter1.Fill(ds)
lstEmployee.DataBind()
```

Haga clic en la imagen para expandir

En la imagen usted puede ver las propiedades que debe configurar en un control para poder asociarlo a un DataSet.

Puede observar también el código necesario para llenar el DataSet y luego para producir el enlace entre el control y el DataSet

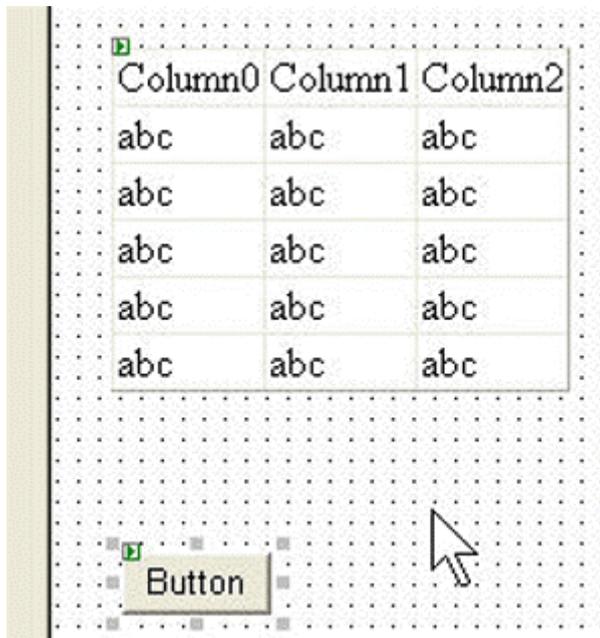
5. 3. 3. A - Práctica: Enlazando controles con una Base de Datos con Visual Studio .NET

Vamos a agregar un formulario Web a nuestro proyecto y lo vamos a llamar '**Medicos**'. Luego agregaremos un control de tipo Grilla. También agregaremos un **DataAdapter**, un objeto de tipo **Connection** y un **DataSet**. Enlazaremos el control y los objetos y ejecutaremos la aplicación.

1. Para agregar un formulario con un control de tipo DataGrid y un Button:

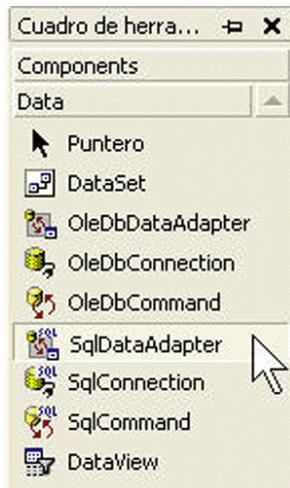
- Abra Visual Studio .NET con el proyecto '**VSCurso**'.
- Vaya al Explorador de Soluciones. Haga clic sobre '**VsCurso**'. Presione el botón derecho del Mouse. Seleccione Agregar, Agregar Nuevo Elemento

- En la platilla seleccione '**Web Forms**', como nombre escriba '**Medicos**'.
- Abra el Cuadro de Herramientas, y agregue un control de tipo '**DataGrid**' y otro de tipo '**Button**'. Modifique la propiedad Text del botón Button. Escriba '**Aceptar**'



2. Para agregar objetos de tipo DataAdapter, Connection y DataSet.

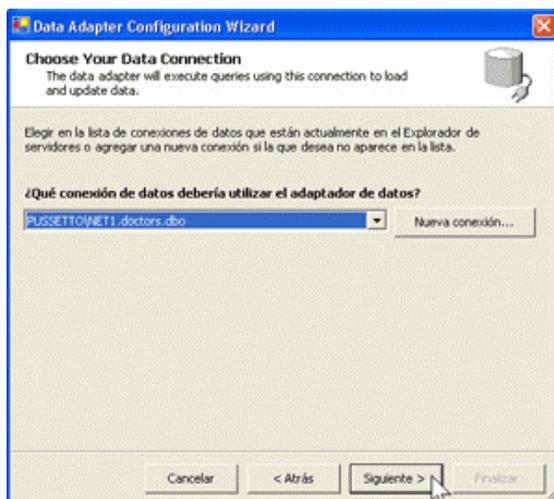
- Vaya al Cuadro de Herramientas, Seleccione el tab **Data**, Agregue un control de tipo DataAdapter al formulario.



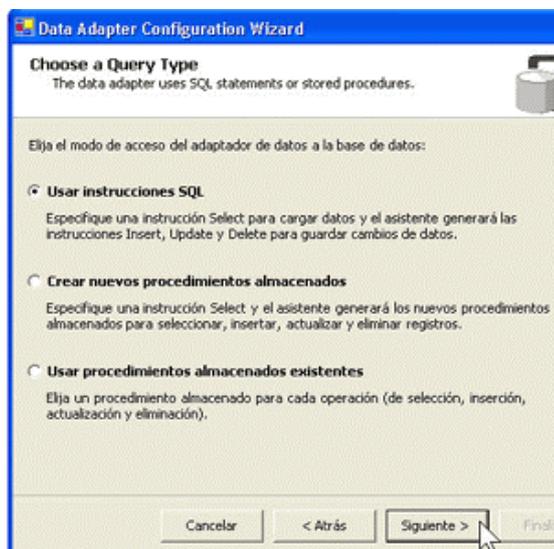
- Va a ver un asistente. Haga clic en **Siguiente**.
En la lista desplegable tenemos el objeto conexión con la Base de Datos de Doctores.
- Haga clic en **Siguiente**



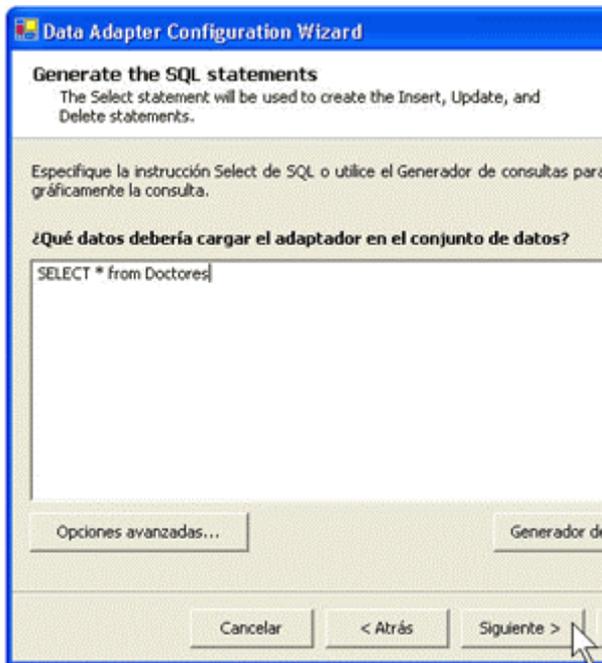
En esta pantalla puede seleccionar como quiere que el data seleccione los datos de la Base de Datos. Vamos a dejar la opción default.



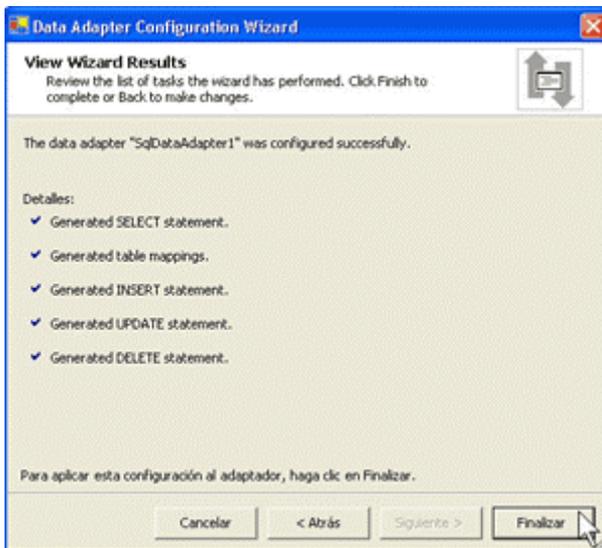
- Haga clic en **Siguiente**



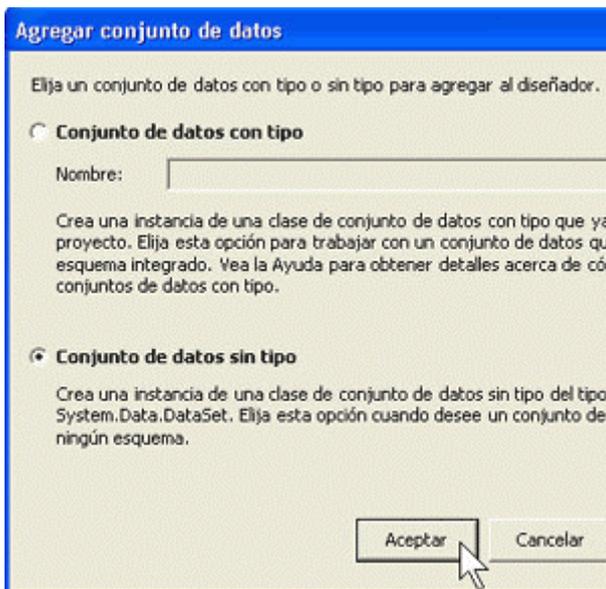
- Aquí debe escribir la sentencia para trabajar con los datos.
- Escriba **'SELECT * FROM DOCTORES'**
- Haga clic en **Siguiente**



- Haga clic en **Finalizar**



- Vaya al Cuadro de Herramientas y agregue un control de tipo DataSet al formulario
- Seleccione la opción que dice **'conjunto de datos sin tipo'**. Haga clic en **Aceptar**.

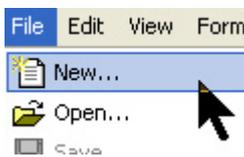


Hemos agregado un formulario, una grilla, un Button y los objetos DataAdapter, Connection y DataSet.

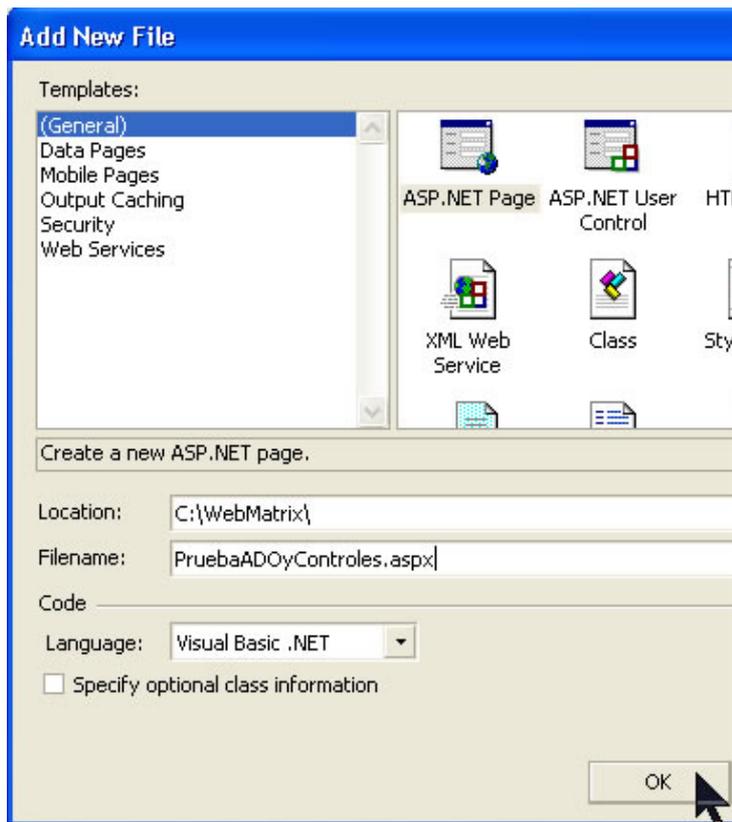
5. 3. 3. b. - Práctica: Enlazando controles con una base de datos en Web Matrix.

Vamos a asociar un control '**ListBox**' (caja de Lista) con la columna '**Nombre**' de la tabla '**Doctores**' en la Base de Datos '**Doctores**'

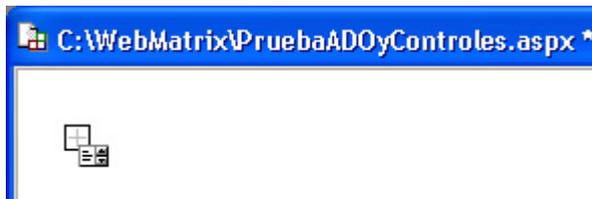
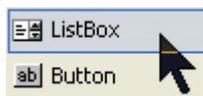
1. Vaya al menú de Web Matrix y seleccione '**File**' (archivo), '**New**'



2. En '**Templates**' (plantillas), seleccione '**(General)**'.
3. Deje el default '**ASP.NET Page**'.
4. En '**Location**' (ubicación) elija algún directorio.
5. En '**Filename**' (nombre), escriba Prue baADOyControles.aspx.
6. Haga clic en '**OK**'.



7. En la ventana '**ToolBox**' (Caja de Herramientas), tome un control de tipo '**ListBox**' y suéltelo en el formulario.



8. Tome el borde derecho del control '**ListBox**' y hágalo más grande.



9. Seleccione el tab '**All**' en la parte inferior del formulario



10. Modifique la línea 1, agregue '**Debug=true**', tal como se ve en la siguiente figura. Esto le va a permitir encontrar los errores, en el caso que su página los tenga.

```

C:\WebMatrix\PruebaADOyControles.aspx *
1 <%@ Page Language="VB" Debug="true" %>
2 <%@ Register TagPrefix="wmix" Namespace="Microsoft
3 <%@ Import Namespace="system.data" %>
4 <%@ Import Namespace="system.data.SqlClient" %>
5 <script runat="server">

```

11. Escriba el siguiente código. Lea los comentarios en verde que explican lo que hace cada instrucción. Tenga en cuenta en 'Data Source' de escribir el nombre de su servidor SQL + \ + nombre de la instancia si la tiene o como este caso (local)

```

7 Sub Page_Load(Sender As Object, E As EventArgs)
8
9 'se declara y se instancia un objeto de tipo connection
10 Dim con As New SqlConnection("data source='PUSSETTO\NET1'; & _
11 "integrated security=SSPI; Database='Doctores'")
12 'se declara y se instancia un objeto de tipo dataAdapter
13 Dim da As New SqlDataAdapter("SELECT * FROM DOCTORES", con)
14 'se declara un objeto de tipo DataSet
15 Dim ds As New DataSet()
16
17 'carga el dataset con los datos que obtiene el DataAdapter
18 da.Fill(ds, "Doctores")
19 'asigna el dataset a la propiedad dataSource (Origen de Datos) de la lista
20 ListBox1.DataSource = ds
21 'se asigna a la propiedad DataTextField (Campo de texto de datos) el valor
22 'de la columna nombre del dataset
23 ListBox1.DataTextField = ds.Tables("Doctores").Columns("Nombre").ToString()
24 'este método es necesario para enlazar los datos a la lista
25 ListBox1.DataBind()
26
27 end sub

```

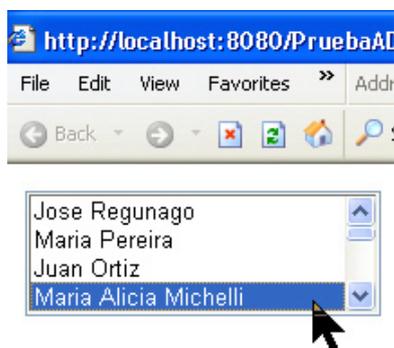
12. Haga clic en el icono 'Save File' (guardar archivo).



13. Haga clic en el icono 'Start' (comenzar).



Va a ver en su explorador de Internet la lista cargada con los nombres de los doctores cargados en la tabla 'Doctores'.



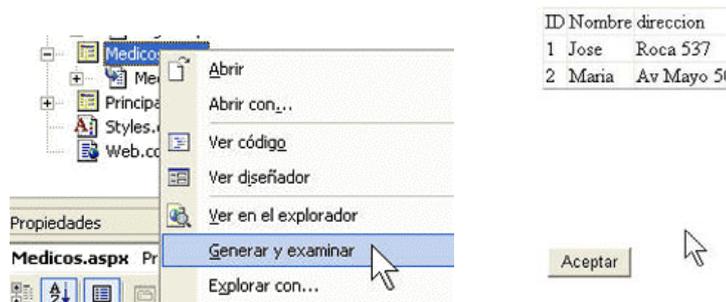
En esta práctica asociamos un control de tipo lista a una tabla en un origen de datos.

5. 3. 4. A- Práctica: Usando un Datagrid

Vamos a agregar el código para poder ver los datos en la grilla.

- Vaya al formulario. Haga Doble click en el botón **Aceptar**. En el evento click escriba el siguiente código.
- `sqlDataAdapter1.Fill(DataSet1)`
- `DataGrid1.Datasource = DataSet1`
- `DataGrid1.DataBind()`
- Vaya al explorador de soluciones. Presione el botón derecho del Mouse sobre el archivo '**Medicos.aspx**'. Seleccione '**Generar y Examinar**'
- Va a aparecer la ventana del formulario. Haga clic en el botón **Aceptar**.

Va a ver los datos de la tabla Doctores en la grilla



5. 3. 4. b. - Práctica: Usando un DataGrid en Web Matrix

Vamos a usar una plantilla para hacer un formulario que permite editar una grilla con datos.

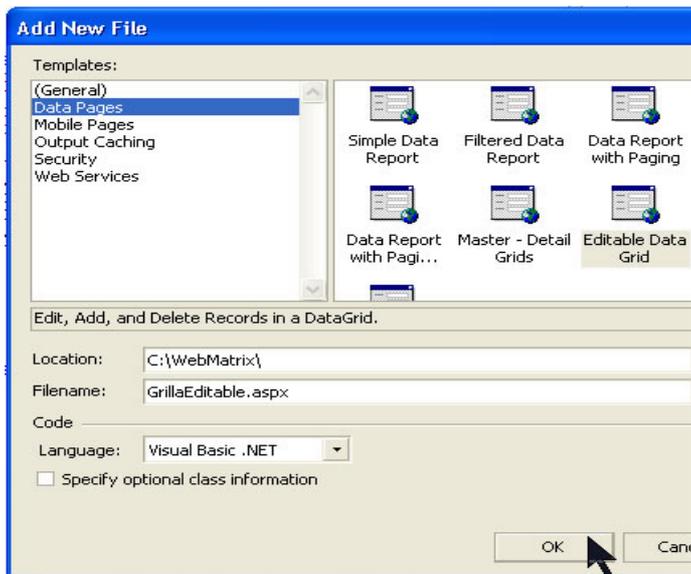
1. Vamos a usar una plantilla para hacer un formulario que permite editar una grilla con datos.

```

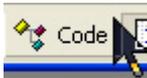
78 'obtenga los valores de las casillas de texto
79 Dim ID As String = CType(e.Item.Cells(2).Controls(0), TextBox).Text
80 Dim Nombre As String = CType(e.Item.Cells(3).Controls(0), TextBox).Text
81 Dim Direccion As String = CType(e.Item.Cells(4).Controls(0), TextBox).Text
82 Dim Ciudad As String = CType(e.Item.Cells(5).Controls(0), TextBox).Text
83
84 'actualice el objeto command
85 Dim myConnection As New SqlConnection(ConnectionString)
86 Dim UpdateCommand As SqlCommand = new SqlCommand()
87 UpdateCommand.Connection = myConnection
88
89 'si selecciona agregar Nuevo, se inserta un nuevo registro
90 'sino se modifica la tabla
91 If AddingNew = True Then
92     UpdateCommand.CommandText = "INSERT INTO doctores(ID, Nombre, Direccion, " & _
93         "Ciudad) VALUES (@ID, @Nombre, @Direccion, @Ciudad)"
94 Else
95     UpdateCommand.CommandText = "UPDATE doctores SET Nombre = @Nombre, " & _
96         "Direccion = @Direccion, Ciudad = @Ciudad WHERE ID = @ID"
97 End If
98
99 UpdateCommand.Parameters.Add("@ID", SqlDbType.VarChar, 11).value = id
100 UpdateCommand.Parameters.Add("@Nombre", SqlDbType.VarChar, 40).value = Nombre
101 UpdateCommand.Parameters.Add("@Direccion", SqlDbType.VarChar, 20).value = Direccion
102 UpdateCommand.Parameters.Add("@Ciudad", SqlDbType.VarChar, 20).value = Ciudad

```

2. En '**Templates**' (plantillas), seleccione '**Data Pages**' (páginas de datos).
3. Seleccione la plantilla '**Editable Data Grid**' (Grilla de datos editable).
4. En '**Location**', seleccione algún directorio.
5. En '**Filename**', escriba **GrillaEditable.aspx**.
6. Haga clic en '**OK**'.



- Haga clic en el tab '**Code**' (código) del formulario.



- Modifique '**connectionString**' y '**SelectCommand**' tal como se ve en la figura. En Server escriba el nombre correcto de su servidor.

```

4  <script runat="server">
5
6  'Actualice el string de conexión y el objeto command
7  Dim connectionString As String = "server='pussetto\net1'; " & _
8  "database=doctores;trusted_connection=true"
9  Dim selectCommand As String = _
10 "SELECT ID, Nombre, Direccion, ciudad from Doctores"
11
12 Dim isEditing As Boolean = False
13
14 Sub Page_Load(Sender As Object, E As EventArgs)
15
16     If Not Page.IsPostBack Then
17
18         'Enlaza los datos a la grilla únicamente la
19         'primera vez que la página es llamada
20
21         BindGrid()
22
23     End If
24
25 End Sub

```

- Modifique el evento '**CheckIsEditing**'. Cambie el mensaje (en este caso en la línea 49-52), tal como se ve en la figura).

```

41 Sub CheckIsEditing(commandName As String)
42
43 If DataGrid1.EditItemIndex <> -1 Then
44
45 ' estamos editando un registro
46 If commandName <> "Cancel" And commandName <> "Update" Then
47
48 ' si el usuario hizo modificaciones no se van a guardar
49 Message.Text = "sus cambios no han sido guardados." & _
50 "Por favor apriete Actualizar para guardar sus cambios, " & _
51 "o cancelar para descartar las modificaciones, " & _
52 "antes de seleccionar otro item."
53
54 isEditing = True
55
56 End If
57
58 End If
59
60 End Sub

```

10. Modifique el evento '**DataGrid_Update**', en este caso las modificaciones son en las líneas **78-81, 91-100**.

```

74 Sub DataGrid_Update(Sender As Object, E As DataGridCommandEventArgs)
75
76 ' actualice la Base de Datos con los nuevos valores
77
78 ' obtenga los valores de las casillas de texto
79 Dim ID As String = CType(e.Item.Cells(2).Controls(0), TextBox).Text
80 Dim Nombre As String = CType(e.Item.Cells(3).Controls(0), TextBox).Text
81 Dim Direccion As String = CType(e.Item.Cells(4).Controls(0), TextBox).Text
82 Dim Ciudad As String = CType(e.Item.Cells(5).Controls(0), TextBox).Text
83
84 ' actualice el objeto command
85 Dim myconnection As New SqlConnection(ConnectionString)
86 Dim UpdateCommand As SqlCommand = new SqlCommand()
87 UpdateCommand.Connection = myConnection
88
89 ' si selecciona agregar Nuevo, se inserta un nuevo registro
90 ' sino se modifica la tabla
91 If AddingNew = True Then
92 UpdateCommand.CommandText = "INSERT INTO doctores(ID, Nombre, Direccion, " & _
93 "Ciudad) VALUES (@ID, @Nombre, @Direccion, @Ciudad)"
94
95 Else
96 UpdateCommand.CommandText = "UPDATE doctores SET Nombre = @Nombre, " & _
97 "Direccion = @Direccion, Ciudad = @Ciudad WHERE ID = @ID"
98
99 End If
100 UpdateCommand.Parameters.Add("@ID", SqlDbType.VarChar, 11).Value = id
101 UpdateCommand.Parameters.Add("@Nombre", SqlDbType.VarChar, 40).Value = Nombre
102 UpdateCommand.Parameters.Add("@Direccion", SqlDbType.VarChar, 20).Value = Direccion
103 UpdateCommand.Parameters.Add("@Ciudad", SqlDbType.VarChar, 20).Value = Ciudad
104
105 ' execute the command

```

11. Modifique el evento '**DataGrid_Delete**', en este caso la modificación es en la línea **152-15**

```

Sub DataGrid_Delete(Sender As Object, E As DataGridCommandEventArgs)
    ' borra un registro
    If Not IsEditing Then
        ' el valor clave para el registro se encuentra en la colección DataKeys
        Dim keyValue As String = CStr(DataGrid1.DataKeys(e.Item.ItemIndex))

        ' actualiza el valor del objeto SqlCommand
        Dim myConnection As New SqlConnection(ConnectionString)
        Dim DeleteCommand As New SqlCommand("DELETE from Doctóres " & _
            "where ID='" & keyValue & "'", myConnection)

        ' execute the command
    End If
End Sub

```

12. Modifique el evento 'AddNew_Click', en este caso la modificación es en la línea 196

```

183 Sub AddNew_Click(Sender As Object, E As EventArgs)
184     ' agrega un nuevo registro al final de los datos y habilita la edición
185     CheckIsEditing("")
186     CheckIsEditing("")
187     CheckIsEditing("")
188     CheckIsEditing("")
189     If Not IsEditing = True Then
190         ' setea la bandera para que poder insertar un nuevo registro
191         AddingNew = True
192         AddingNew = True
193         AddingNew = True
194         AddingNew = True
195         ' primero obtiene los datos
196         Dim myConnection As New SqlConnection(ConnectionString)
197         Dim myCommand As New SqlDataAdapter(selectCommand, myConnection)
198         Dim ds As New DataSet()
199         myCommand.Fill(ds)
200         myCommand.Fill(ds)
201         myCommand.Fill(ds)
202         ' agrega un registro al final de los datos
203         Dim rowValues As Object() = {"", "", "", ""}
204         ds.Tables(0).Rows.Add(rowValues)
205     End If
End Sub

```

13. Haga clic en el tab 'HTML'.



14. Modifique el tag <h2>, Escriba 'Grilla Editable'.

```

<body style="FONT-FAMILY
    <h2>Grilla Editable
</h2>

```

15. Busque el tag <asp:LinkButton .

```

<asp:LinkButton id="LinkButton1"
<br />

```

16. Modifique la propiedad **Text**, escriba '**Agregar Nuevo Item**'.

```
Text="Agregar Nuevo Item"
```

17. Guarde el formulario.



18. Ejecute la aplicación.



Grilla Editable

		ID	Nombre	Direccion	ciudad
Edit	Delete	1	Jose Regunago	Roca 537	Capital
Edit	Delete	2	Maria Pereira	Av Mayo 50	Capital
Edit	Delete	3	Juan Ortiz	San Martin 23	Sunchales
Edit	Delete	4	Maria Alicia Michelli	Alegria 200	Colonia Bossi
Edit	Delete	5	Celina Staiger	Santa Fe 98	Colonia Bossi
Edit	Delete	6	Jose Milanese	Rivadavia 90	El Bolson

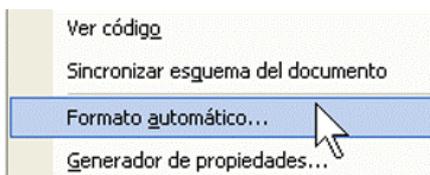
1 2 3

[Agregar Nuevo Item](#)

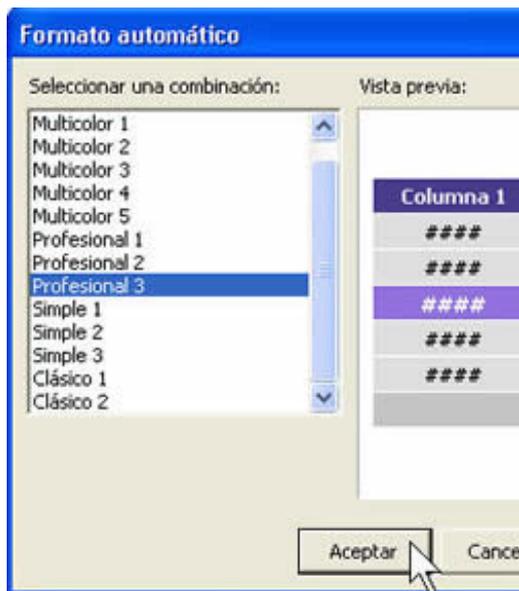
5. 3. 5. A- Práctica: Modificando algunas propiedades de un control DataGrid

Vamos a modificar algunas propiedades de la grilla.

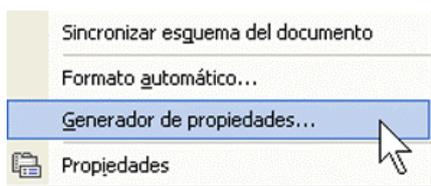
- Haga clic en la grilla. Presione el botón derecho del Mouse. Seleccione '**Formato automático**'



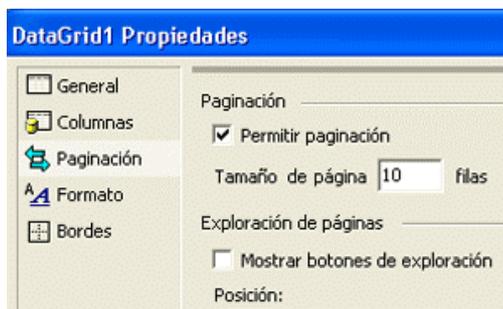
- Seleccione algún formato. Haga clic en Aceptar.



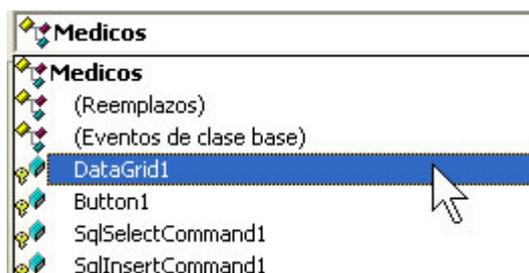
- Haga clic en la grilla. Presione el botón derecho del Mouse. Seleccione '**Generador de Propiedades**'



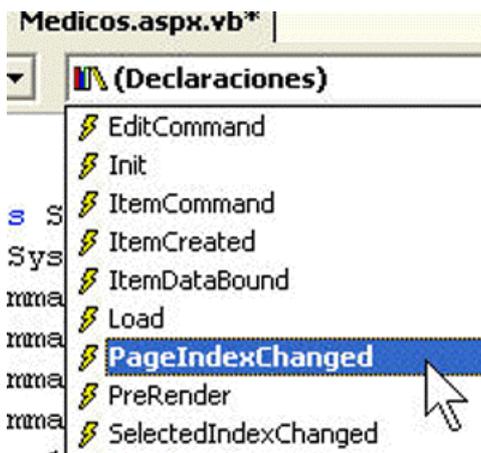
- Vaya al tab '**Paginación**'. Haga clic en el botón que dice '**Permitir Paginación**'. Haga clic en **Aceptar**



- Haga Doble click en el formulario. En la lista desplegable superior izquierda seleccione '**DataGrid1**'



- En la lista desplegable superior derecha seleccione **'DataGrid1_PageIndexChanged'**



- Escriba en el evento **'DataGrid1_PageIndexChanged'** el siguiente código :

```
Datagrid1.CurrentPageIndex = e.NewPageIndex  
SqlDataAdapter1.Fill (DataSet1)  
DataGrid1.DataBind()
```

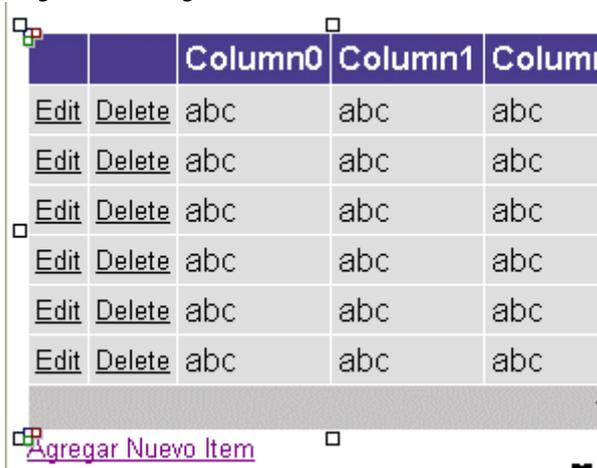
- Vaya al explorador de soluciones. Presione el botón derecho del Mouse sobre el archivo **'Medicos.aspx'**. Seleccione **'Generar y Examinar'**

Verá en el formulario, las modificaciones en la grilla. Si quiere ver como hace la paginación agregue más datos a la tabla de doctores.

5. 3. 5. b. - Práctica: Modificando algunas propiedades de un control DataGrid en Web Matrix

Vamos a modificar algunas propiedades de la grilla que usamos en la práctica anterior.

- Haga clic en la grilla.

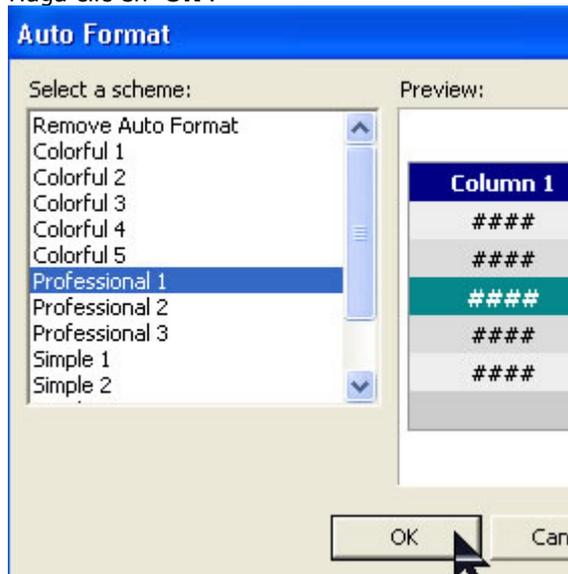


- En la ventana **'Properties'** (propiedades), Haga clic en **'Auto Format'** (autoformato).



- Donde dice **'Select a scheme'** (seleccione un esquema), pruebe distintas opciones. Por ejemplo, **'Professional 1'**.

4. Haga clic en 'OK'.



5. En la ventana 'Properties' (propiedades), Haga clic en 'Property Builder' (constructor de propiedades).



6. Donde dice 'Header and footer', seleccione 'Show footer'. (mostrar pie de página).

7. Haga clic en 'OK'.



8. Ejecute la aplicación.



Grilla Editable

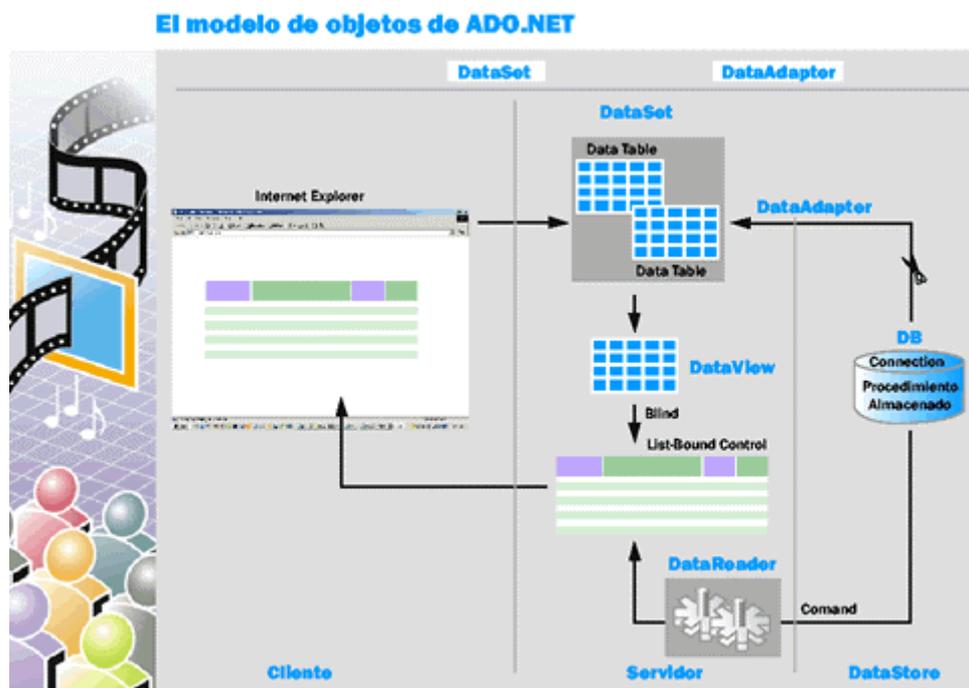
		ID	Nombre
Edit	Delete	1	Jose Regunago
Edit	Delete	2	Maria Pereira
Edit	Delete	3	Juan Ortiz
Edit	Delete	4	Maria Alicia Michelli
Edit	Delete	5	Celina Staiger
Edit	Delete	6	Jose Milanese
			1 2 3

[Agregar Nuevo Item](#)

Va a ver como la grilla se modificó de acuerdo a los cambios que hizo.

5. 3. 6 - El modelo ADO.NET

En la imagen puede ver como interactúan los objeto del modelo de ADO.NET



5. 3. 7. - DataSets vs. DataReaders

En la imagen puede ver la diferencia entre los objetos DataSets y DataReaders

Usando DataSets vs. DataReaders

DataSet	DataReader
Acceso a datos de Lectura/Escritura	Lectura únicamente
Puede incluir varias tablas y distintos orígenes de datos	Basado en una sola sentencia SQL de una sola Base de Datos
Modelo Desconectado	Modelo conectado
Se puede asociar a muchos controles	Se puede asociar a un solo control
Se puede recorrer los datos hacia adelante y atrás	Únicamente hacia adelante
Acceso mas lento	Acceso más rápido
Soportado por las herramientas de Visual Studio .NET	Codificado manualmente

5. 3. 8. - Accediendo a datos con DataSets

En esta sección veremos con código como establecer una conexión con un servidor y manipular los datos usando distintos objetos del modelo ADO.NET

5. 3. 8. 1. - Estableciendo una Conexión

Veamos el código necesario para establecer una conexión a una base de datos

```
Dim strConn As String = "data source=localhost; " & _
    "initial catalog=MiBaseDeDatos; integrated security=true"
Dim conn As New SqlConnection(strConn)
```

En el cuadro tiene algunos parámetros que usa el objeto Connection y su descripción

Parámetro	Descripción
Connection Timeout	Es el tiempo máximo que espera el objeto para poder establecer una conexión
Data Source	Es el nombre del Servidor para establecer la conexión
Integrated Security	Permite usar las cuentas del sistema operativo y cuentas locales en el servidor SQL Server para establecer la conexión
Inicial Catalog	Es el nombre de la base de datos a la cual se va a conectar

5. 3. 8. 2. -Creando un DataAdapter

El objeto DataSet representa una copia local de los datos obtenidos desde algún origen de datos. Es útil para tener una copia local de los datos que los formularios Web pueden utilizar.

El **DataAdapter** sirve de enlace entre un origen de datos y el DataSet. Hay 2 tipos:

- **OLEDBDataAdapter:** Provee de acceso a cualquier origen de Datos
- **SqlDataAdapter:** Provee acceso únicamente a SQL Server 7.0 o posterior

Creando un un objeto DataAdapter

Guardar la consulta en un SqlDataAdapter

```
Dim da As New SqlDataAdapter ("select * from Doctores", conn)
```

El SqlDataAdapter configura la propiedad SelectCommand

```
da.SelectCommand.CommandText  
da.SelectCommand.Connection
```

Si lo requiere configure las propiedades InsertCommand, UpdateCommand, y DeleteCommand

Cuando un SqlDataAdapter se conecta a un origen de datos puede ejecutar acciones. EL siguiente cuadro resume las 4 acciones:

Propiedad	Función
SelectCommand	Obtiene registros de un origen de Datos
InsertCommand	Inserta registro en un origen de datos
UpdateCommand	Modifica registros en un origen de datos
DeleteCommand	Borra registros en un origen de datos

Vemos el código

```
Dim strConn As String = "data source=localhost; " & _  
"initial catalog=MiBaseDeDatos; integrated security=true"  
  
'crea una conexión  
Dim conn As New SqlConnection(strConn)  
  
'crea un objeto de tipo SqlDataAdapter  
Dim da as New SqlDataAdapter( Select * from Doctores, Conn )
```

5. 3. 8. 3. - Creando un DataSet con código

El los datos de un DataSet se cargan partir de objetos DataTables. Primero se debe declarar un objeto DataSet y luego se lo carga con los datos obtenidos a partir de algún objeto SqlDataAdapter.

'el siguiente código crea un DataSet y lo carga con Datos

```
Dim ds As New DataSet ()  
da.fill (ds, "Doctores")  
  
'para acceder a un objeto DataTable dentro de un SqlDataAdapter debe usar  
el siguiente código  
ds.Tables ("Doctores")  
  
'para acceder al valor de un columna en un objeto DataTable use el  
código  
ds.Tables ("Doctores") . Rows (x) . ("Nombre")  
  
'x es el índice del elemento dentro del objeto DataTables  
'nombre es el nombre de la columna en el elemento x  
  
'Veamos como recorrer un objeto DataTable y mostrar sus campos  
Dim r As DataRow  
Dim str As String  
For Each r In ds.tables("Doctores").rows
```

```

str = r(0) 'agrega el valor de la primera columna
str += " " 'espacio en blanco
str += r ("Nombre") 'agrega el valor del Campo llamado Nombre
Response.Write (str) 'devuelve al cliente el string que armó
Next

```

'el operador de string += concatena lo que tiene la variable a la derecha del operador con el string que se pone a la izquierda del operador

5. 3. 9. - El concepto de colección

Recordemos que un arreglo es un conjunto de elementos del mismo tipo. Una colección es un conjunto de elementos de cualquier tipo. Sus elementos son accedidos mediante un índice entero. En ADO.NET el primer elemento de una colección tiene índice 0.

Ejemplos de colecciones en ADO.NET son:

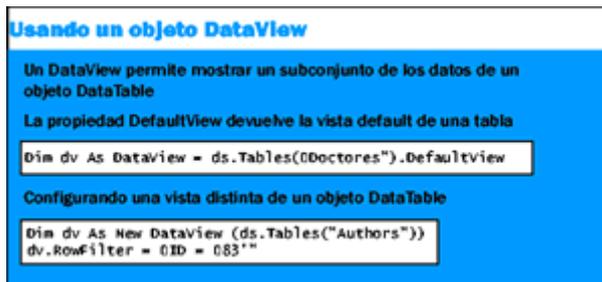
DataSet : Es una colección de DataTables
 DataTable: Es una colección de DataRow
 DataRow : Es una colección de DataColumnn

5. 3. 10. - Que es y como se usa un objeto DataView

Un objeto DataView permite mostrar un objeto DataTable bajo distintos parámetros. Por ejemplo, los datos se pueden ordenar y filtrar.

El objeto DataView luego puede ser asociado a un control

En la imagen tiene el código necesario para crear un objeto DataView.



5. 3. 11 - ¿Qué es un objeto DataReader?

Cuando usted debe recuperar una gran cantidad de registros de un origen de datos el objeto DataTable puede usar demasiada memoria y recursos. El objeto DataReader permite usar menos recursos y acceder más rápidamente a los datos. El costo de esto es que puede ser recorrido únicamente hacia adelante y sus datos no pueden ser modificados Además la conexión al origen de datos debe hacerse en forma explícita El objeto DataReader puede ser derivado de 2 clases: SqlDataReader y OleDbDataReader. La imagen muestra una comparación entre el uso de un DataTable y un DataSet

5. 3. 12 - ¿Cómo crear un objeto DataReader?

En la imagen puede ver los pasos para crear y usar un objeto DataReader . El objeto Command utiliza una conexión y ejecuta algún tipo de operación en el origen de datos, por ejemplo trae datos.

Veamos un ejemplo con código,

```

'se crea la conexión
Dim conn As New SqlConnection("data source=localhost;" & _
"Integrated Security=true; initial catalog=Doctores") 'se crea un objeto
de tipo command
Dim cmdDoctores As New SqlCommand("select * from doctores", conn)

'se declara crea un objeto de tipo dataReader
Dim dr As SqlDataReader

```

```
'el resultado de la ejecución del objeto command es un DataReader
dr = cmdDoctores.ExecuteReader()
Do while dr.Read()
Response.Write( dr("ID") + " " + dr("Nombre") )
Loop
```

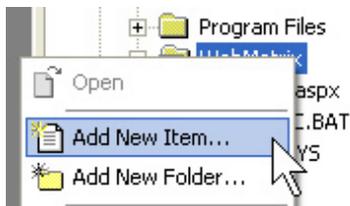
```
dr.close()
conn.close()
```

Observe la estructura del loop. El método Read del objeto DataReader va leyendo los datos uno a uno. A medida que se leen los datos son mostrados al cliente usando el método write del objeto response.

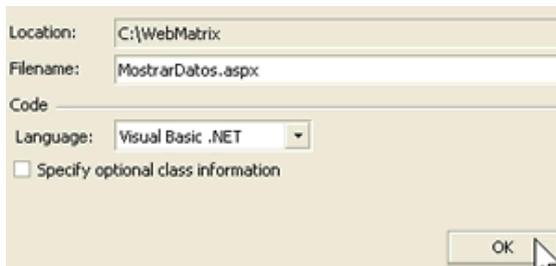
5. 4. - Accediendo a Datos con Web Matrix

Vamos a crear un formulario y ver 2 maneras de mostrar datos usando ADO.NET.

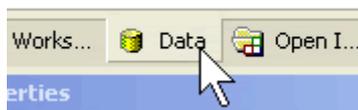
- Vaya al entorno de desarrollo de Web Matrix.
- En la ventana 'workspace' presione el botón derecho del Mouse sobre el folder '**Web Matrix**'



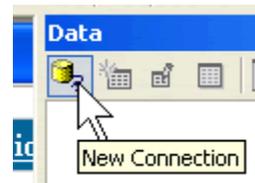
- Agregue un formulario Web. Como nombre escriba '**MostrarDatos**'.



- En la ventana '**workspace**' seleccione el tab '**data**'.



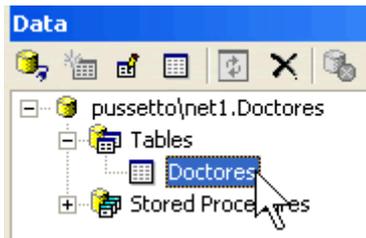
- Haga clic en el icono que dice '**New Connection**'



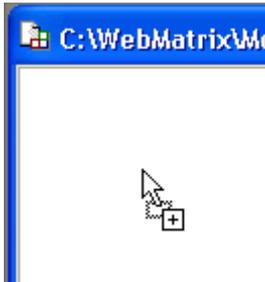
- Seleccione la Base de Datos llamada '**Doctores**' e su servidor. Recuerde '**nombre de máquina + nombre de**

Seleccione la Base de Datos llamada '**Doctores**'.

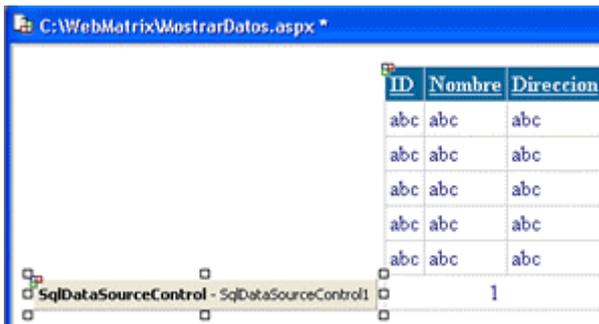
- Expanda los objetos para ver la base de datos '**Doctores**'.



- Tome el objeto '**Doctores**' y suéltelo en el formulario.



- Se va a agregar automáticamente un objeto llamado '**SqlDataSourceControl**' y un objeto llamado '**MxDataGridControl**'.

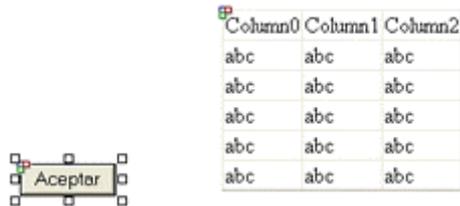


- Ejecute la página. Va a ver los datos de la tabla de '**Doctores**' en una grilla.
- Cierre la ventana.

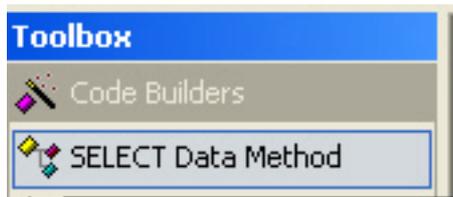


Ahora vamos a usar otro método para mostrar datos en un formulario.

- Agregue un control de tipo **'Datagrid'** al formulario y un control de tipo **'Button'**. Modifique la propiedad Text del control 'Button' para que sea **'Aceptar'**. La parte inferior del formulario **'MostrarDatos'** va a quedar como se muestra en la figura.
- Haga Doble click en el botón **'Aceptar'**.



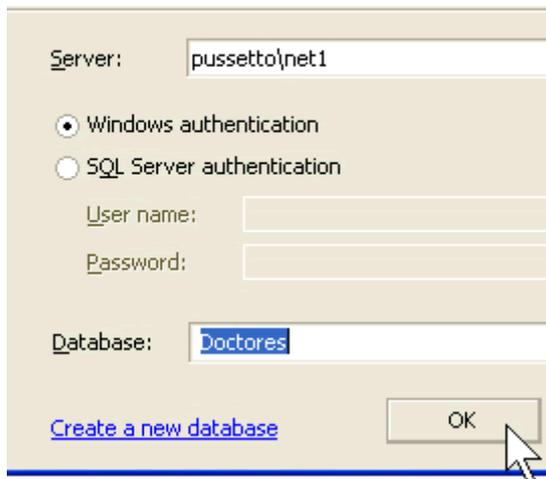
- En el cuadro de herramientas. Seleccione el botón llamado **'SELECT Data Method'** que se encuentra en el tab **'Code Builders'**.



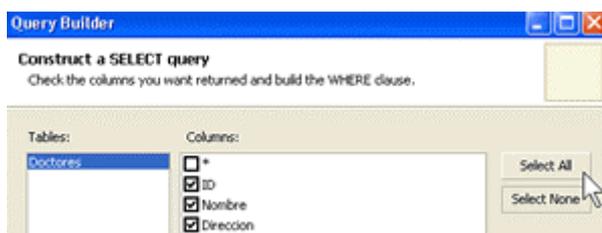
- Seleccione la Base de datos de **'Doctores'** antes

Connect to SQL or MSDE Database

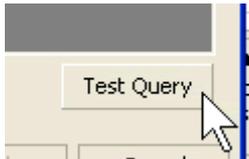
Enter the connection information and select a database



- Va a ver una ventana para armar una consulta. Presione el botón que dice **'Select All'**.



- Haga clic en el botón llamado **'Test Query'**.



- Va a ver los datos de la tabla **'Doctores'**
Query Preview
Test your query by entering some test data.

	ID	Nombre	Direccion
▶	1	Jose	Roca 537
	2	Maria	Av Mayo 50
*			

- Deje las opciones que aparecen por default. Haga clic en el botón **'Finish'**
Enter a caption for this WizardPanel
Enter a description for this WizardPanel

This CodeBuilder will generate a public method with :
should the method be called?
MyQueryMethod

The method can return either an System.Data.SqlClient
System.Data.DataSet. Which do you want?
 DataSet
 DataReader

- Vaya a la vista diseño del formulario. Haga Doble click en el botón **'Aceptar'** y escriba el código que aparece en la figura.

```
C:\WebMatrixMostrarDatos.aspx *  
* Insert page code here  
Sub Button1_Click(sender As Object, e /  
datagrid1.datasource = MyQueryMethod()  
datagrid1.databind()  
End Sub
```

- Ejecute el formulario. Va a ver primero la grilla como antes. Luego Haga clic en **'Aceptar'**. Se va a llenar la otra grilla con datos.

Hemos visto como mostrar datos en un formulario usando ADO.NET con Web Matrix.

5. 5. - Revisión

En este capítulo vimos el modelo de objetos de ADO.NET. En el próximo capítulo veremos como hacer nuestros propios servicios Web.