

## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)

### Conceptos.

Con ASP.Net podemos comunicarnos a distintos gestores de base de datos como pueden ser SQL Server, Oracle, Access, MySQL etc.

Nosotros trabajaremos con el gestor de base de datos SQL Server por ser el más empleado cuando se utiliza la tecnología de ASP.Net en el desarrollo de sitios web dinámicos.

En esta clase especificaremos todo el código necesario para acceder al gestor de base de datos, desde la cadena de conexión hasta la implementación de los comandos SQL a enviar.

Si no tiene instalado el SQL Server puede seguir los pasos indicados [aquí](#).

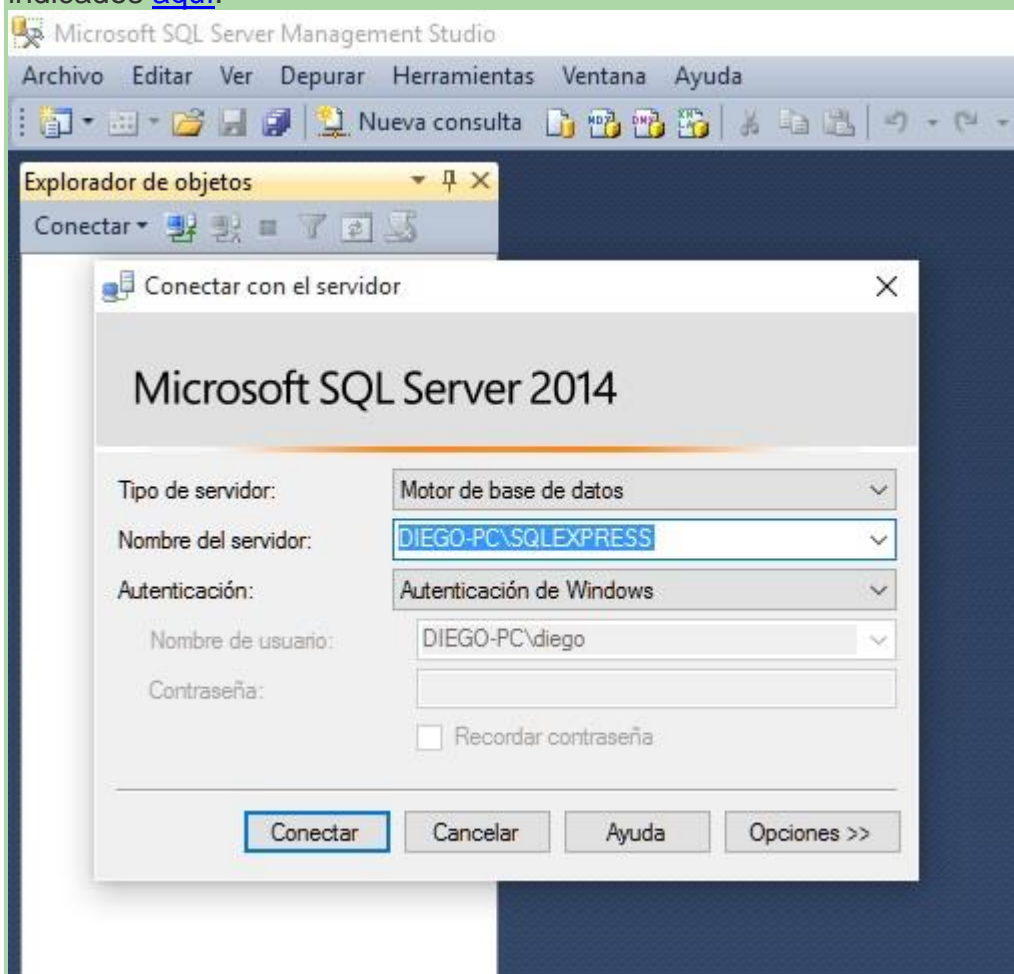
### ABM (Altas, Bajas y Modificaciones)

Crearemos una base de datos en SQL Server llamada 'base1' y dentro de la misma definiremos una tabla llamada usuarios con tres campos:

```
nombre varchar(30)    Clave primaria
clave  varchar(30)
mail   varchar(70)
```

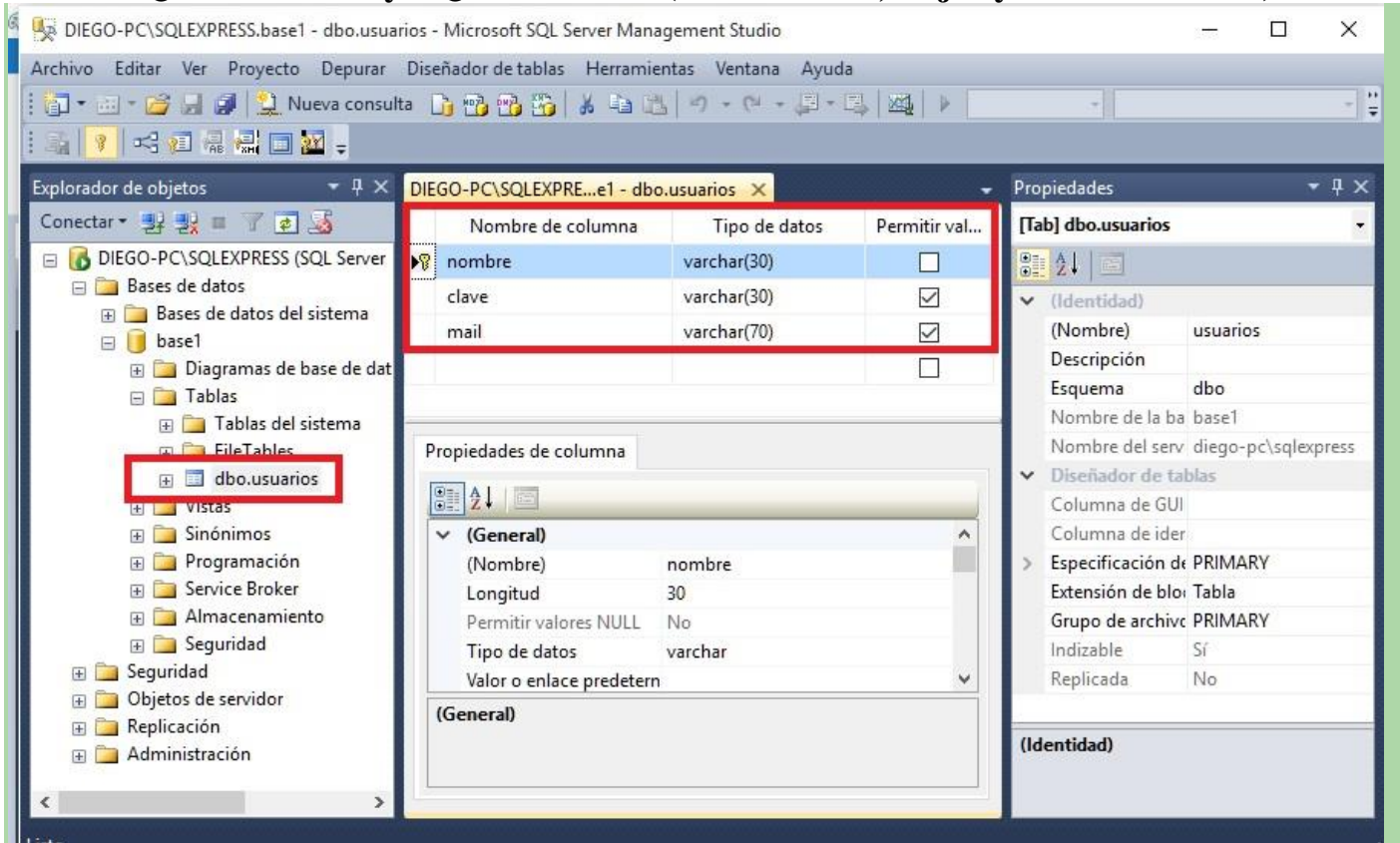
Para crear la base de datos podemos hacerlo desde el Microsoft SQL Server Management Studio:

Si no tiene instalado el Microsoft SQL Server Management Studio puede seguir los pasos indicados [aquí](#).



Creamos luego la tabla usuarios:

## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)



Ya tenemos creada la base de datos 'base1' y la tabla 'usuarios', crearemos un sitio web vacío de ASP.NET en el Visual Studio llamado ejercicio011.

Agregaremos 5 Web Form llamados:

Default.aspx

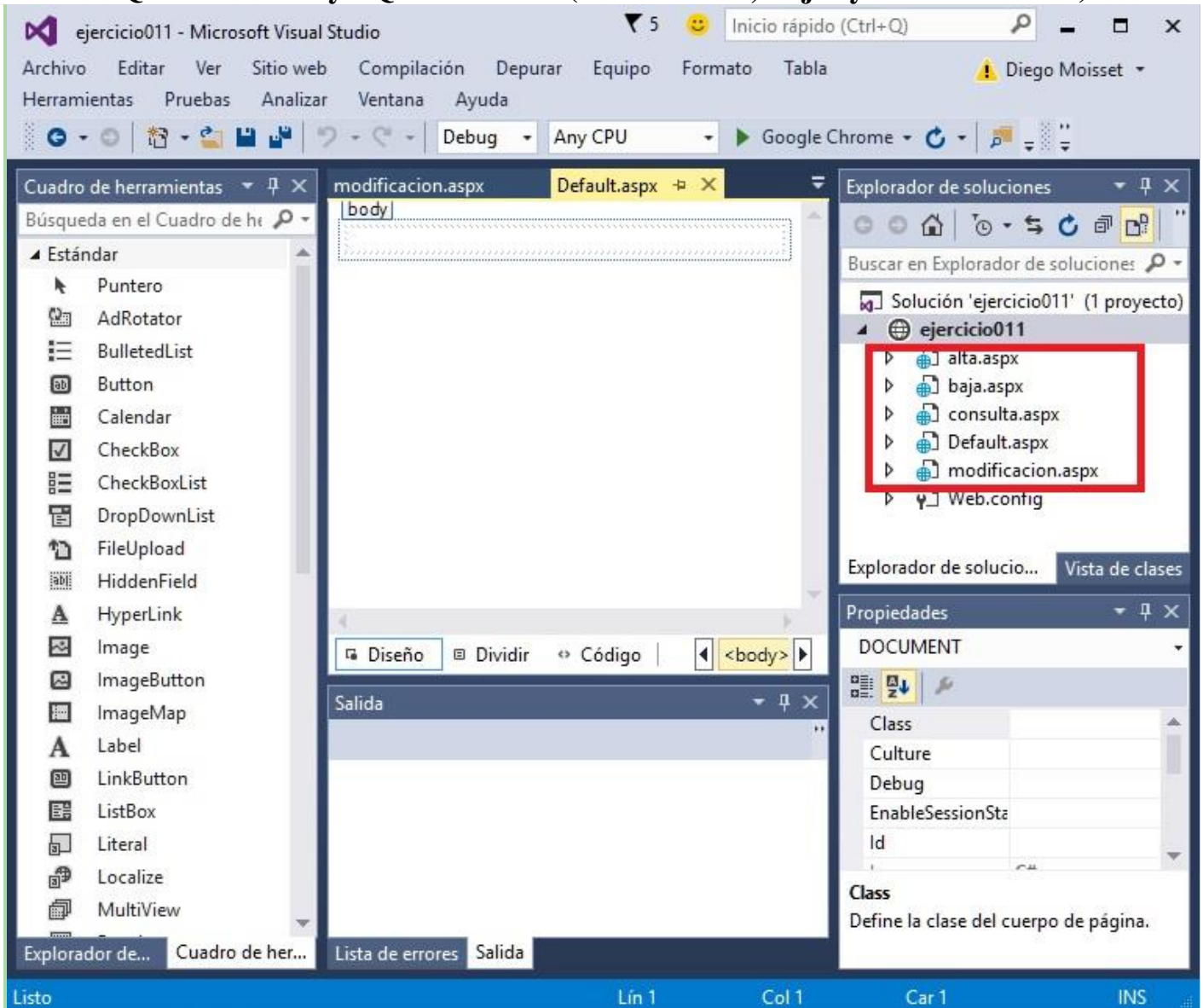
alta.aspx

consulta.aspx

baja.aspx

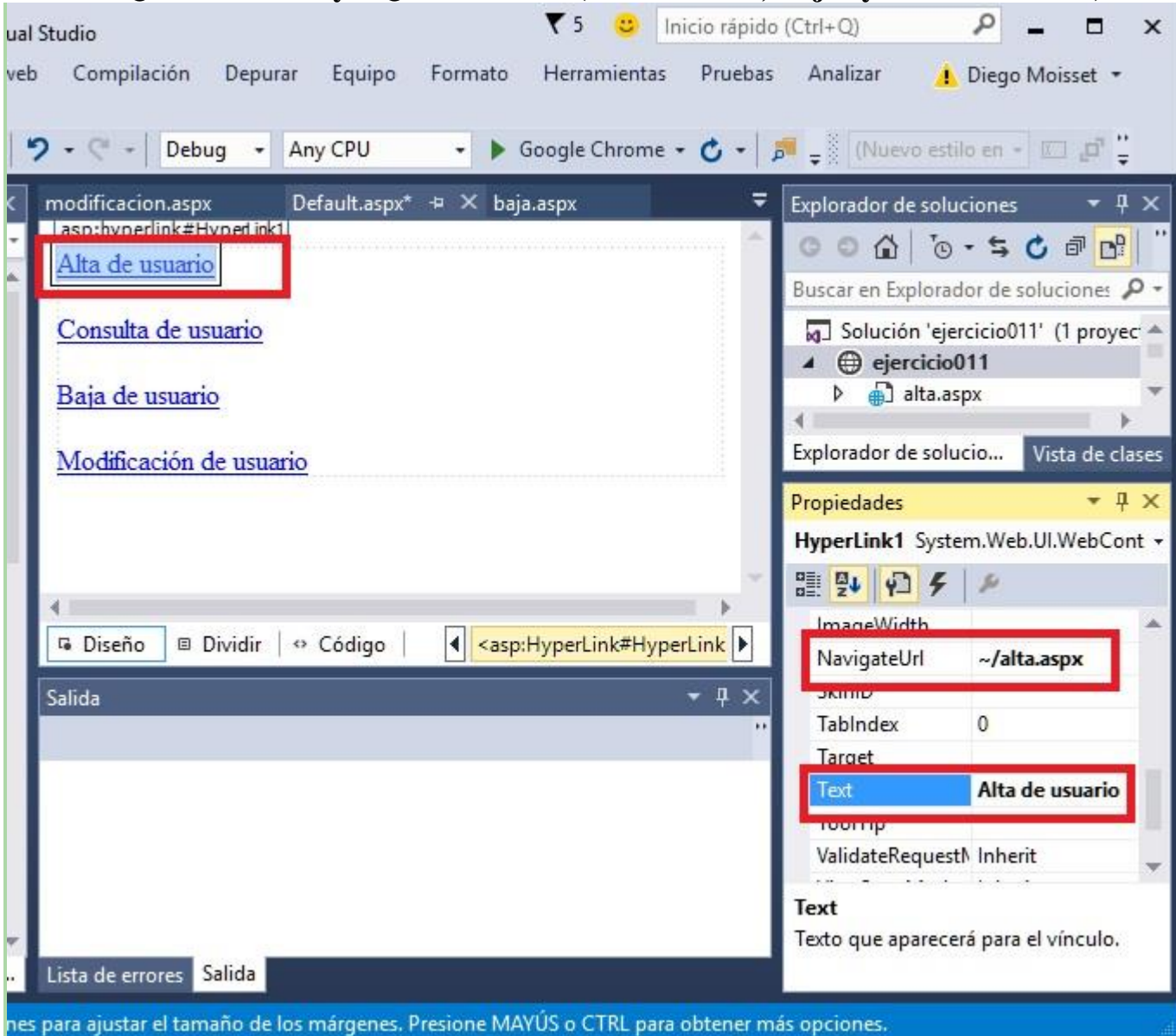
modificacion.aspx

## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)



La página Default.aspx solo tendrá los hipervínculos a las otras páginas que tendrán por objetivo efectuar una el alta de usuarios, otra la consulta, otra la baja y la última la modificación:

## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)



Para crear esta interfaz insertaremos cuatro objetos de la clase HyperLink, como mínimo debemos inicializar las propiedades text (es el texto que mostrará el hipervínculo en el navegador y la propiedad NavigateUrl que indica el nombre de la página que debe cargar el navegador cuando se presione el hipervínculo)

### Definición de la cadena de conexión con la base de datos en el archivo Web.config

Web.config es el archivo principal de opciones de configuración para una aplicación web en ASP.NET.

El archivo es un documento XML que define información de configuración concerniente a la aplicación web. El archivo Web.config contiene información que controla la carga de módulos, configuraciones de seguridad, configuraciones del estado de la sesión, opciones de compilación y el lenguaje de la aplicación.

El archivo Web.config contiene también la cadenas de conexión a la base de datos.

Debemos agregar al archivo el elemento connectionStrings:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<!--
```

```
Para obtener más información sobre cómo configurar la aplicación de ASP.NET, visite  
http://go.microsoft.com/fwlink/?LinkId=169433
```

```
-->
```



## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)

```
<configuration>

  <system.web>
    <compilation debug="true" targetFramework="4.5.2" />
    <httpRuntime targetFramework="4.5.2" />
  </system.web>

  <connectionStrings>
    <add name="cadenaconexion1"
        connectionString="Data Source=DIEGO-PC\SQLEXPRESS;Initial
Catalog=base1;Integrated Security=true;" />
  </connectionStrings>

</configuration>
```

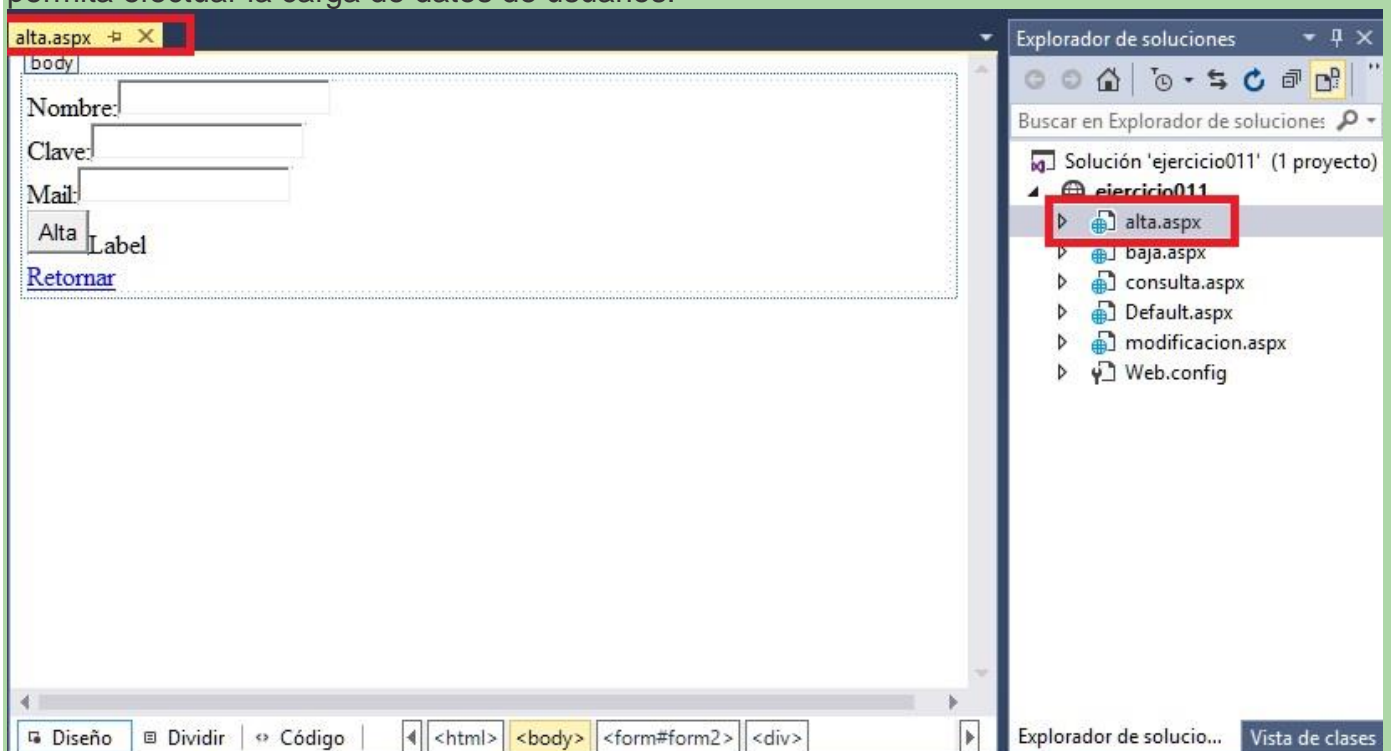
Como vemos en la propiedad connectionString indicamos en Initial Catalog el nombre de la base de datos que hemos creado en SQL Server 'base1'.

En la propiedad Data Source usted debe fijarse como se llama la instancia de SQL Server que instaló en su máquina, esto puede hacerlo cuando arranca el SQL Server Management Studio donde dice "Nombre del servidor"

Luego recuperaremos esta cadena de conexión mediante el nombre 'cadenaconexion1'.

### Altas

Activemos desde el Visual Studio la pestaña alta.aspx para elaborar la interfaz visual que nos permita efectuar la carga de datos de usuarios:



Como podemos ver disponemos tres controles de tipo TextBox, el que solicita el ingreso de la clave modificamos la propiedad TextMode con el valor Password, los otros dos los dejamos con el valor SingleLine.

Disponemos un objeto de la clase Button y una Label donde mostraremos un mensaje si el alta se efectuó correctamente.

Por último disponemos un objeto de la clase HyperLink configurando la propiedad NavigateUrl con la dirección de la página principal (Default.aspx)

Ahora codificamos el evento clic del botón de alta:

## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

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

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        string s =
System.Configuration.ConfigurationManager.ConnectionStrings[
"cadenaconexion1"].ConnectionString;
        SqlConnection conexion = new SqlConnection(s);
        conexion.Open();
        SqlCommand comando = new SqlCommand("insert into
usuarios(nombre,clave,mail) values('" +
        TextBox1.Text + "',''" + this.TextBox2.Text +
        "',''" +
        TextBox3.Text + "')", conexion);
        comando.ExecuteNonQuery();
        Label1.Text = "Se registro el usuario";
        conexion.Close();
    }
}
```

Lo primero que debemos hacer es importar el espacio de nombres donde se encuentra definida la clase `SqlConnection`:

```
using System.Data.SqlClient;
```

Al presionar el botón, primero extraemos la cadena de conexión que tenemos almacenada en el archivo `Web.config`:

```
string s =
System.Configuration.ConfigurationManager.ConnectionStrings["cadenaconexion1"].Connecti
onString;
```

Creamos un objeto de la clase `SqlConnection` indicando como parámetro la cadena de conexión que rescatamos anteriormente:

```
SqlConnection conexion = new SqlConnection(s);
```

Abrimos la conexión:

```
conexion.Open();
```

Creamos un objeto de la clase `SqlCommand` creándolo con los datos cargados en los controles `TextBox`:

```
SqlCommand comando = new SqlCommand("insert into usuarios(nombre,clave,mail)
values('" +
        TextBox1.Text + "',''" + this.TextBox2.Text + "',''" +
        TextBox3.Text + "')", conexion);
```

## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)

Pedimos a SQL Server que ejecute el comando especificado anteriormente:

```
comando.ExecuteNonQuery();
```

Cerramos la conexión:

```
conexion.Close();
```

## Consultas

Seleccionamos del Explorador de soluciones la página consulta.aspx y procedemos a elaborar la siguiente interfaz visual (disponemos un TextBox, un Button, una Label y un HyperLink:



El código del evento click del botón es:

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

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

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        string s =
System.Configuration.ConfigurationManager.ConnectionStrings[
"cadenaconexion1"].ConnectionString;
        SqlConnection conexion = new SqlConnection(s);
```

## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)

```
conexion.Open();
SqlCommand comando = new SqlCommand("select
nombre,clave,mail from usuarios " +
    " where nombre='" + this.TextBox1.Text + "'",
conexion);
SqlDataReader registro = comando.ExecuteReader();
if (registro.Read())
    this.Label1.Text = "Clave:" + registro["clave"]
+ "<br>" +
    "Mail:" + registro["mail"];
else
    this.Label1.Text = "No existe un usuario con
dicho nombre";
conexion.Close();
}
```

Para poder recuperar los datos lo hacemos creando un objeto de la clase SqlDataReader e inicializándolo mediante la llamada del método ExecuteReader de la clase SqlCommand:

```
SqlDataReader registro = comando.ExecuteReader();
```

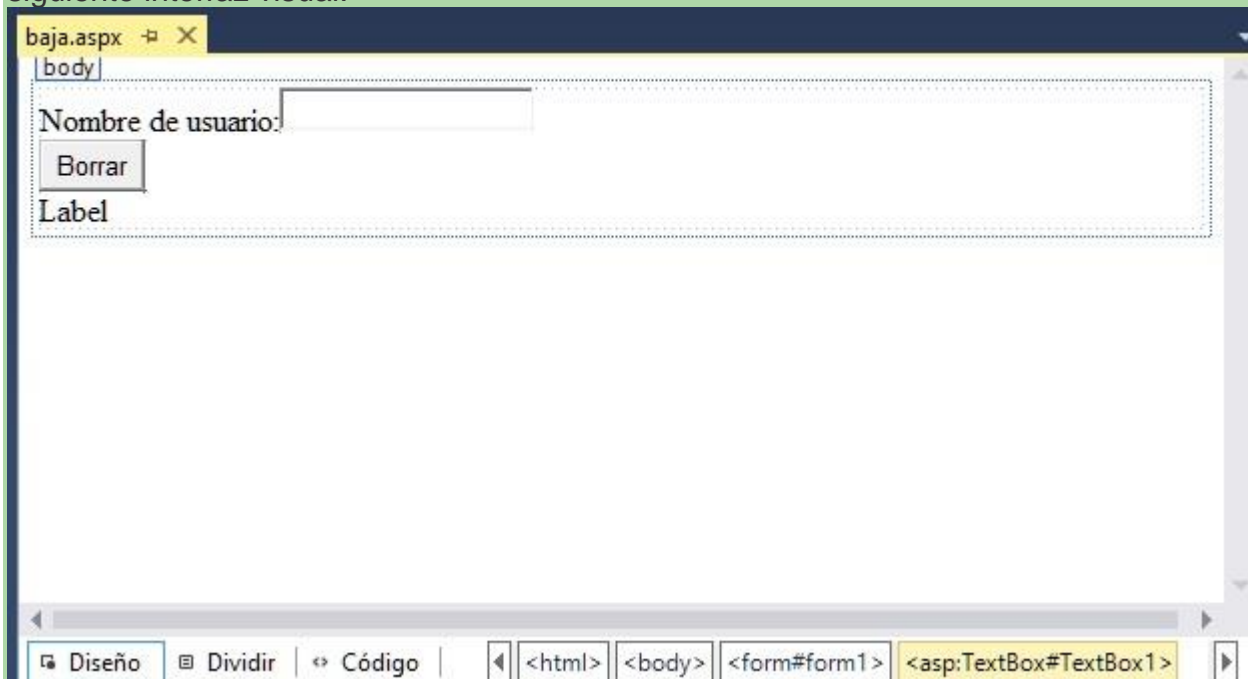
Luego recorremos el SqlDataReader (como este caso puede retornar cero o una fila lo hacemos mediante un if:

```
if (registro.Read())
```

Si el método Read retorna true luego podemos acceder a la fila recuperada con el select.

### Baja

Seleccionamos del Explorador de soluciones la página baja.aspx y procedemos a elaborar la siguiente interfaz visual:



Luego el código a ejecutar cuando se presiona el botón ?Borrar? de la página es:

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
```



## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)

```
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

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

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        string s =
System.Configuration.ConfigurationManager.ConnectionStrings[
"cadenaconexion1"].ConnectionString;
        SqlConnection conexion = new SqlConnection(s);
        conexion.Open();
        SqlCommand comando = new SqlCommand("delete from
usuarios where nombre='" + this.TextBox1.Text + "'",
conexion);
        int cantidad = comando.ExecuteNonQuery();
        if (cantidad == 1)
            this.Label1.Text = "Se borró el usuario";
        else
            this.Label1.Text = "No existe un usuario con
dicho nombre";
        conexion.Close();
    }
}
```

El método ExecuteNonQuery retorna un entero y representa la cantidad de filas borradas de la tabla.

### Modificaciones

Por último implementaremos la modificación de datos. Seleccionamos del Explorador de soluciones la página modificacion.aspx y procedemos a elaborar la siguiente interfaz visual:

## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)

modificacion.aspx\* baja.aspx

Nombre de usuario:

Buscar Label

Clave:

Mail:

Modificar Label

[Retornar](#)

Diseño | Dividir | Código | <body> <form#form1> <div> <asp:Button#Button1>

Para efectuar la modificación de datos de un usuario procederemos primero a la búsqueda de los datos actuales.

Luego el código para los eventos clic de los dos botones es:

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

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

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        string s =
System.Configuration.ConfigurationManager.ConnectionStrings[
"cadenaconexion1"].ConnectionString;
        SqlConnection conexion = new SqlConnection(s);
        conexion.Open();
        SqlCommand comando = new SqlCommand("select
nombre,clave,mail from usuarios " +
            " where nombre='" + this.TextBox1.Text + "'",
conexion);
        SqlDataReader registro = comando.ExecuteReader();
        if (registro.Read())
        {
```

## Clases SqlConnection y SqlCommand (ABM - Altas, Bajas y Modificaciones)

```
        this.TextBox2.Text =
registro["clave"].ToString();
        this.TextBox3.Text =
registro["mail"].ToString();
    }
    else
        this.Label1.Text = "No existe un usuario con
dicho nombre";
        conexion.Close();
    }

protected void Button2 Click(object sender, EventArgs e)
{
    string s =
System.Configuration.ConfigurationManager.ConnectionStrings[
"cadenaconexion1"].ConnectionString;
    SqlConnection conexion = new SqlConnection(s);
    conexion.Open();
    SqlCommand comando = new SqlCommand("update usuarios
set " +
        "clave='" + this.TextBox2.Text +
        "',mail='" + this.TextBox3.Text +
        "' where nombre='" + this.TextBox1.Text +
        "'", conexion);
    int cantidad = comando.ExecuteNonQuery();
    if (cantidad == 1)
        this.Label2.Text = "Datos Modificados";
    else
        this.Label2.Text = "No existe el usuario";
    conexion.Close();
}
}
```

El botón ?Buscar? hace lo mismo que vimos en la consulta. Luego cuando se presiona el botón ?Modificar? procedemos a hacer un update de la tabla usuarios con los datos cargados en los TextBox.

### Ejercicio propuesto

1 ? Crear una tabla:

alumnos (dni varchar(8), apellidonom varchar(50), provincia varchar(30))

Confeccionar una serie de páginas que permitan efectuar altas, bajas, modificaciones y consultas.