
	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 1 de 22		

GUIA N° 6 - APLICACIÓN WEB TRES CAPAS ASP.NET

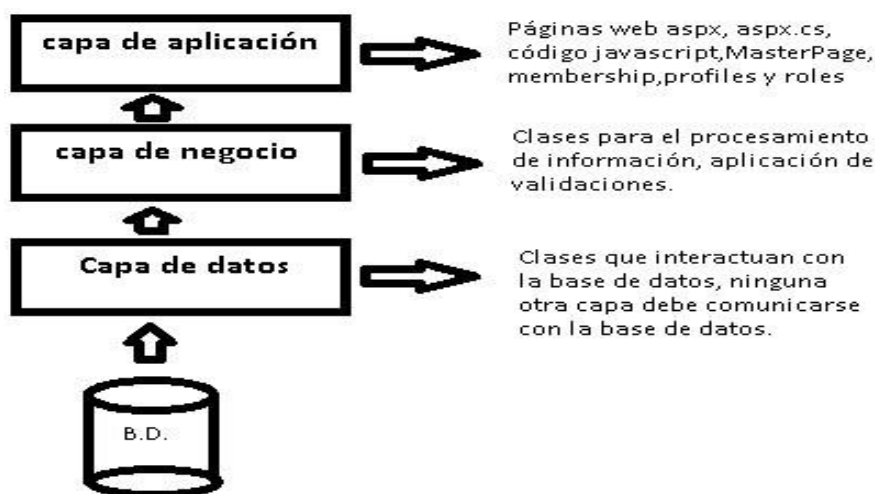
OBJETIVO:

Conocer los principios y fundamentos para desarrollar una aplicación web tres capas usado ASP.NET

INTRODUCCIÓN

El modelo en capas, es una arquitectura de software muy utilizada en el desarrollo de aplicaciones web, especialmente en el ámbito empresarial. La arquitectura más simple que se suele utilizar en estos contextos es el denominado modelo de “Tres Capas”. En el desarrollo de aplicaciones empresariales y corporativas, el número de capas puede aumentar, sin embargo, la configuración del proyecto para el desarrollo en visual studio, es prácticamente la misma sin importar el número de capas.

El modelo de desarrollo web a tres capas en términos generales se describe cómo se puede observar a continuación:

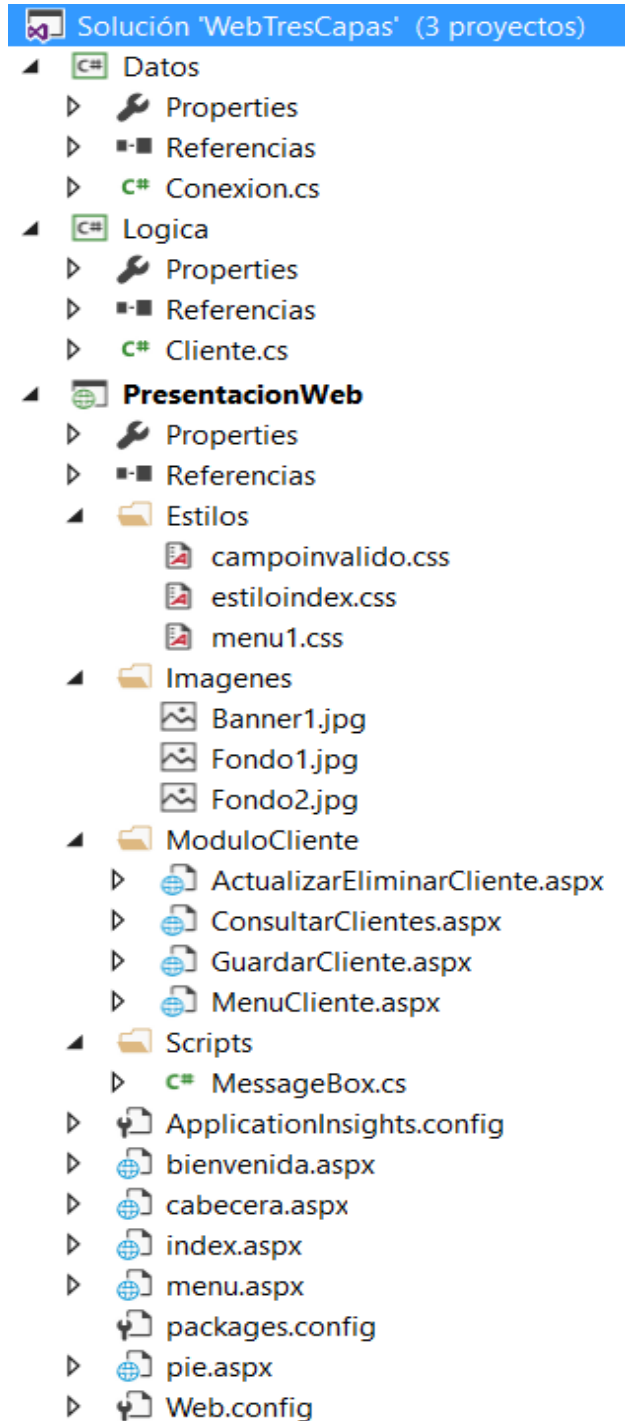




I. PROYECTO WEB TRES CAPAS EN VISUAL STUDIO .NET

1. En la opción Nuevo proyecto de Visual Studio seleccione la opción: *Otros tipos de proyecto – Solución En Blanco*, establezca como nombre de solución “WebTresCapas”.
2. En el explorador de soluciones haga click derecho sobre el proyecto “WebTresCapas”, escoja la opción agregar nuevo proyecto, web, y agregue una nueva aplicación web ASP.Net (C#) y nómbrala como “PresentacionWeb”
3. En el explorador de soluciones haga click derecho sobre el proyecto “WebTresCapas”, escoja la opción agregar nuevo proyecto, agregue una nueva Biblioteca de Clases (C#) y nómbrala como “Logica”.
4. En el explorador de soluciones haga click derecho sobre el proyecto “WebTresCapas”, escoja la opción agregar nuevo proyecto, agregue una nueva Biblioteca de Clases (C#) y nómbrala como “Datos”.

	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 2 de 22		

5. Al final una vez siga todos los pasos de esta guía el proyecto debe verse similar a como se muestra a continuación:



	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 3 de 22		

II. CAPA DE DATOS

En el SGBD "SQL Server" implemente una base de datos de nombre "Directorio" usando el script que se muestra a continuación:

```

CREATE TABLE Clientes (
  Identificacion Bigint PRIMARY KEY,
  Nombre VARCHAR(150),
  Apellido VARCHAR(150),
  Fijo VARCHAR(50),
  Celular VARCHAR(50));

CREATE PROCEDURE [dbo].[InsertarCliente]
  @Identificacion Bigint
  ,@Nombre varchar(150)
  ,@Apellido varchar(150)
  ,@Fijo varchar(50)
  ,@Celular varchar(50)
AS BEGIN

INSERT INTO Clientes
  (Identificacion
  ,Nombre
  ,Apellido
  ,Fijo
  ,Celular)
VALUES
  (@Identificacion
  ,@Nombre
  ,@Apellido
  ,@Fijo
  ,@Celular)

END



CREATE PROCEDURE ActualizarCliente
  @Identificacion Bigint
  ,@Nombre varchar(150)
  ,@Apellido varchar(150)
  ,@Fijo varchar(50)
  ,@Celular varchar(50)
AS BEGIN

UPDATE Clientes SET
  Nombre = @Nombre
  ,Apellido = @Apellido
  ,Fijo = @Fijo
  ,Celular = @Celular

WHERE Identificacion = @Identificacion

END

```

	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 4 de 22		

```
CREATE PROCEDURE ConsultarClientes
```

```
@Valor varchar(15),
@Columna varchar(15)
```

```
AS BEGIN
```

```
IF @Columna = 'Todas'
    SELECT * FROM Clientes
```

```
Else IF @Columna = 'Identificacion'
    SELECT * FROM Clientes
    WHERE identificacion=@Valor
```

```
Else IF @Columna = 'Nombre'
    SELECT * FROM Clientes
    WHERE Nombre like @Valor + '%'
```

```
Else IF @Columna = 'Apellido'
    SELECT * FROM Clientes
    WHERE Apellido like @Valor + '%'
```

```
End
```

```
CREATE PROCEDURE EliminarCliente
    @Identificacion Bigint
```

```
AS BEGIN
```

```
DELETE FROM Clientes WHERE Identificacion = @Identificacion;
END
```

Una vez implementada la base de datos dirijase a Visual Studio .Net y en el explorador de soluciones del proyecto en ubique la capa de "Datos" y agregue la clase "Conexion" y en cada bloque de código escriba las instrucciones según se muestra a continuación

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

namespace Datos
{
    public class Conexion
    {
        private string mensaje;
        SqlConnection conn;

        public string Mensaje
        {
            get { return mensaje; }
            set { mensaje = value; }
        }
    }
}
```



GUÍA DE TRABAJO N° 6 - DESARROLLO WEB

Educación Media Fortalecida
SED/SENA

Programación de Software
GRADO 11

Ing. Néstor Raúl Suarez Perpiñan

Página 5 de 22





```
public Conexion()
{
    //cadena de Conexion Para SQL Server:
    String cadenaconexion = @"Data Source= (localdb)\MSSQLLocalDB; Initial
    Catalog = Directorio; Integrated Security=True";
    conn = new SqlConnection(cadenaconexion);
}

public DataSet ConsultarSQL(String SentenciaSQL)
{
    try
    {
        conn.Open();
        SqlDataAdapter objRes = new SqlDataAdapter(SentenciaSQL, conn);
        DataSet datos = new DataSet();
        objRes.Fill(datos, "DatosConsultados");
        mensaje = "La consulta de datos fue Exitosa";
        return datos;
    }
    catch (Exception MiExc)
    {
        DataSet datos2=new DataSet();
        mensaje = "ERROR: " + MiExc.Message;
        return datos2;
    }
    finally {conn.Close(); }
}

public bool EjecutarSQL(String SentenciaSQL)
{
    try
    {
        conn.Open();
        SqlCommand miComando = new SqlCommand();
        miComando.Connection = conn;
        miComando.CommandText = SentenciaSQL;
        miComando.ExecuteNonQuery();

        mensaje = "Proceso Ejecutado con Exito";
        return true ;
    }
    catch (Exception e)
    {
        mensaje = "Se presento el siguiente error " + e.Message;
        return false;
    }
    finally
    { conn.Close(); }
}
}
```

	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 6 de 22		

III. CAPA DE LÓGICA DE APLICACIÓN (LÓGICA DEL NEGOCIO)

1. **IMPORTANTE:** Desde el explorador de soluciones en la parte de la solución llamada “Logica” y dentro del submenú llamado “Referencias” escoja la opción agregar referencia. En la ventana Agregar Referencia escoja la pestaña “Proyectos”, seleccione “Datos” en nombre de proyecto y finalmente presione click en el botón aceptar.

2. Desde el explorador de soluciones en la parte de la solución llamada “Logica” agregue una nueva Clase y llámela “Cliente”.

3. Ubique la clase cliente para escribir dentro las instrucciones según se muestran a continuación

```
using System.Data;
using Datos;

namespace Logica
{
    public class Cliente:Conexion
    {
        private long identificacion;
        private string nombre;
        private string apellido;
        private string fijo;
        private string celular;

        public long Identificacion
        {
            get { return identificacion; }
            set { identificacion = value; }
        }

        public string Nombre
        {
            get { return nombre; }
            set { nombre = value; }
        }

        public string Apellido
        {
            get { return apellido; }
            set { apellido = value; }
        }

        public string Fijo
        {
            get { return fijo; }
            set { fijo = value; }
        }

        public string Celular
        {
            get { return celular; }
            set { celular = value; }
        }
    }
}
```



```
public bool InsertarCliente()
{
    string ProcedimientoInsertar =
        "EXEC InsertarCliente @Identificacion = " + this.identificacion +
        ",@Nombre = '" + this.nombre + "', @Apellido = '" + this.apellido + "',
        @Fijo = '" + this.fijo + "', @Celular = '" + this.celular + "'";

    bool respuestaSQL = EjecutarSQL(ProcedimientoInsertar);
    return respuestaSQL;
}

public bool ActualizarCliente()
{
    string ProcedimientoInsertar =
        "EXEC ActualizarCliente @Identificacion = " + this.identificacion +
        ",@Nombre = '" + this.nombre + "', @Apellido = '" + this.apellido + "',
        @Fijo = '" + this.fijo + "', @Celular = '" + this.celular + "'";



    bool respuestaSQL = EjecutarSQL(ProcedimientoInsertar);
    return respuestaSQL;
}

public bool EliminarCliente(string identificacion_buscar)
{
    string ProcedimientoInsertar =
        "EXEC EliminarCliente @Identificacion = " + identificacion_buscar;

    bool respuestaSQL = EjecutarSQL(ProcedimientoInsertar);
    return respuestaSQL;
}

public DataSet ConsultarCliente(string Valor,string Columna)
{
    string ProcedimientoDeConsulta =
        "EXEC ConsultarClientes @Valor = '" + Valor + "', @Columna = '" +
        Columna + "'";

    DataSet ConsultaResultante = ConsultarSQL(ProcedimientoDeConsulta);
    return ConsultaResultante;
}
}
```

	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 8 de 22		

I. CAPA DE PRESENTACIÓN WEB

IMPORTANTE: Desde el explorador de soluciones en la parte de la solución llamada “Presentación” y dentro del submenú llamado “Referencias” escoja la opción agregar referencia. En la ventana “Agregar Referencia” escoja la pestaña “Proyectos”, seleccione “Lógica” y “Datos” en nombre de proyecto y finalmente presione click en el botón aceptar.

A continuación encontrará el paso a paso correspondiente a la capa de presentación del proyecto. Los primeros pasos son prácticamente iguales a los que se aplicaron en la guía anterior “Fundamentos de ASP.NET” a excepción del archivo `campoinvalido.css` y el `menú.aspx`. Siga las instrucciones y verifique el funcionamiento de cada uno de los pasos propuestos.

Paso 1. En la sección de “PresentaciónWeb”, cree cuatro nuevas carpetas y llámelas *Estilos*, *Imágenes*, *scripts* y *ModuloCliente* respectivamente

Paso 2: Copie el código de los archivos de las hojas de estilo que se muestran a continuación y guárdelos dentro de la carpeta “Estilos”

✓ Archivo “`campoinvalido.css`”:

```
input:invalid, textarea:invalid {
    background:red;
}
```

✓ Archivo “`estiloindex.css`”:

```
body {margin:0; padding:0;text-align:center;background: #DBEBF6;}
#content {text-align:left;margin:auto;
           width:100%; height:100%;position:absolute;}
#cabecera {width:100%;height:20%;background-color:#0FF;}
#menu {width:20%; height:65%; float:left;}
#principal {width:80%;height:65%;float:left;}
#pie {width:100%;height:15%;}
```

✓ Archivo “`menu1.css`”:

```
#button {
    width: 11em;
    border-right: 1px solid #000;
    padding: 0 0 1em 0;
    margin-bottom: 1em;
    font-family: 'Trebuchet MS', 'Lucida Grande', Verdana, Arial,
    sans-serif;
    background-color: #90bade;
    color: #333;
}
```






```
#button ul {
    list-style: none;
    margin: 0;
    padding: 0;
    border: none;
}
#button li {
    border-bottom: 1px solid #90bade;
    margin: 0;
    list-style: none;
    list-style-image: none;
}
#button li a {
    display: block;
    padding: 5px 5px 5px 0.5em;
    border-left: 10px solid #1958b7;
    border-right: 10px solid #508fc4;
    background-color: #2175bc;
    color: #fff;
    text-decoration: none;
    width: 100%;
}
html>body #button li a {
    width: auto;
}
#button li a:hover {
    border-left: 10px solid #1c64d1;
    border-right: 10px solid #5ba3e0;
    background-color: #2586d7;
    color: #fff;
}
```

Paso 3. Agregue a la raíz del proyecto un web form aspx con el nombre “*bienvenida.aspx*”, copie el código que se muestra a continuación y guárdelo

```
<html>
<head>
    <title>Bienvenido</title>
    <meta charset="utf8" />
</head>
<body>
    <p> BIENVENIDO!</p>
    <p> Seleccione una opción del Menú</p>
</body>
</html>
```

Paso 4. Agregue a la raíz del proyecto un web form aspx con el nombre “*cabecera.aspx*”, copie el código que se muestra a continuación y guárdelo

	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		 Servicio Nacional de Aprendizaje
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 10 de 22		

```

<html>
<head>
    <title>Encabezado del Sitio WEB</title>
    <meta charset="utf8" />
</head>
<body bgcolor="#3366ff">
<center>
    
    <h1> Bienvenido a Mi Sitio WEB!!!</h1>
</center>
</body>
</html>

```

***Nota:** Ubique una imagen de su preferencia en la carpeta Imágenes para que la utilice en la página de encabezado del Sitio WEB

Paso 5. Agregue a la raíz del proyecto un web form aspx con el nombre "pie.aspx", copie el código que se muestra a continuación y guárdelo:

```

<html>
<head>
    <title>Pie Del Sitio WEB</title>
    <meta charset="utf8" />
</head>
<body background="Imágenes/Fondo2.jpg">
<center>
    COLEGIO ENRIQUE OLAYA HERRERA<br />
    PROGRAMACIÓN DE SOFTWARE<br />
    2017
</center>
</body>
</html>

```

***Nota:** Ubique una imagen de su preferencia en la carpeta Imágenes para que la utilice en la página de pie del Sitio WEB

Paso 6. agregue a la raíz del proyecto un web form aspx con el nombre "menu.aspx", copie el código que se muestra a continuación y guárdelo

```

<html>
<head>
    <title>Menu Principal</title>
    <link href="Estilos/menu1.css" rel="stylesheet" type="text/css" />
    <meta charset="utf8" />
</head>
<body background="Imágenes/Fondo1.jpg">
<center>

```



GUÍA DE TRABAJO N° 6 - DESARROLLO WEB

Educación Media Fortalecida
SED/SENA

Programación de Software
GRADO 11

Ing. Néstor Raúl Suarez Perpiñan

Página 11 de 22



```
<h3> Menu de Opciones</h3>
<div id="button">
<ul>
<li><a href="bienvenida.aspx" target="principal">Inicio</a></li>
<li><a href="ModuloCliente/MenuCliente.aspx" target="menu">Clientes </a></li>
</ul>
</div>
</center>
</body>
</html>
```



***Nota:** Ubique una imagen de su preferencia en la carpeta *Imágenes* para que la utilice de fondo del menú

Paso 7: Copie cada una de las siguientes líneas en un editor y guárdelo con el nombre "index.aspx" dentro la carpeta principal del proyecto.

```
<html>
<head>
<title>Mi Pagina Index</title>
<link href="Estilos/estiloindex.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="content">
  <div id="cabecera">
    <iframe width='100%' height='100%' name='cabecera' scrolling="No"
src='cabecera.aspx'></iframe></div>
    <div id="menu" >
      <iframe width='100%' height='100%' name='menu' scrolling="Yes"
src='menu.aspx'></iframe></div>
      <div id="principal">
        <iframe width='100%' height='100%' name='principal' scrolling="Yes"
src='bienvenida.aspx'></iframe></div>
        <div id="pie">
          <iframe width='100%' height='100%' name='pie' scrolling="No"
src='pie.aspx'></iframe>
        </div>
      </div>
</body>
</html>
```

Nota: Observe que en la propiedad "src" de cada iframe se debe colocar el nombre del archivo o página que se quiere cargar en el iframe.

Paso 8. En la carpeta llamada *Scripts* agregue una nueva clase y coloque como nombre "MessageBox.cs" y escriba el código que se muestra a continuación:

	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 12 de 22		

```

using System.Web;
using System.Text;

namespace PresentacionWeb.Scripts
{
    public class MessageBox
    {
        public static void alert(string msg)
        {
            string sMsg = msg.Replace("\n", "\\n"); sMsg = msg.Replace("\"", "");

            StringBuilder sb = new StringBuilder();

            sb.Append(@"<script language='javascript'>"); sb.Append(@"alert( "" +
            sMsg + @"" "" );"); sb.Append(@"</script>");

            HttpContext.Current.Response.Write(sb.ToString());
        }
    }
}

```

Paso 9. En la carpeta llamada “ModuloCliente” agregue un nuevo WebForm, colóquelo como nombre “MenuCliente.aspx” y escriba el código que se muestra a continuación:

```

<html>
<head>
    <title>Menu Principal</title>
    <link href=" ../Estilos/menu1.css" rel="stylesheet" type="text/css" />
    <meta charset="utf-8" />
</head>
<body background=" ../Imagenes/Fondo1.jpg">
<center>
<h3> Menu de Opciones</h3>
<div id="button">
<ul>

<li><a href="GuardarCliente.aspx" target="principal">Guardar Cliente </a></li>
<li><a href="ActualizarEliminarCliente.aspx" target="principal">Actualizar/Eliminar
Cliente</a></li>
    <li><a href="ConsultarClientes.aspx" target="principal">Consultar Clientes/Ver
Todos</a></li>
    <li><a href=" ../menu.aspx" target="menu">Salir</a></li>
</ul>
</div>
</center>
</body>
</html>

```

Paso 10. En la carpeta llamada “ModuloCliente” agregue un nuevo WebForm, colóquelo como nombre “GuardarCliente.aspx”, realice el diseño y escriba el código que se muestra a continuación:



GUARDAR CLIENTE

Identificación:	<input type="text"/>
Nombre:	<input type="text"/>
Apellido:	<input type="text"/>
Fijo(Opcional):	<input type="text"/>
Celular(Opcional):	<input type="text"/>

Label

```
using PresentacionWeb.Scripts;
using Logica;

namespace PresentacionWeb.ModuloCliente
{
    public partial class GuardarCliente : System.Web.UI.Page
    {
        protected void Button_Guardar_Click(object sender, EventArgs e)
        {
            Cliente ObjCliente = new Cliente();

            try
            {
                ObjCliente.Identificacion = long.Parse(TextBox1.Text);
                ObjCliente.Nombre = TextBox2.Text;
                ObjCliente.Apellido = TextBox3.Text;
                ObjCliente.Fijo = TextBox4.Text;
                ObjCliente.Celular = TextBox5.Text;

                bool respuestaSQL = ObjCliente.InsertarCliente();

                if (respuestaSQL == true)
                {
                    MessageBox.alert("Los datos del nuevo cliente fueron insertados correctamente");
                    Label1.Text = "Los datos del nuevo cliente fueron insertados correctamente";

                    TextBox1.Text = ""; TextBox2.Text = ""; TextBox3.Text = "";
                    TextBox4.Text = ""; TextBox5.Text = "";
                }
            }
            else
            {
                MessageBox.alert(ObjCliente.Mensaje);
            }
        }
    }
}
```



```
        Label1.Text = ObjCliente.Mensaje;
    }
}
catch (Exception Ex)
{
    MessageBox.alert("Error!: " + Ex.Message + " " + ObjCliente.Mensaje);
    Label1.Text = "Error!: " + Ex.Message + " " + ObjCliente.Mensaje;
}
}
}
}
```

IMPORTANTE: Para validar campos en ASP.NET se pueden hacer uso de las propiedades que ofrece html5, en este caso en la sección de código de etiquetas del WebForm agregue la referencia a la hoja de estilos llamada "campoinvalido.css" y las propiedades a cada uno de los TextBox para que el código quede tal y como se muestra a continuación:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Guardar</title>
    <link href=" ../Estilos/campoinvalido.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <form id="form1" runat="server">
        <center>
            <div> GUARDAR CLIENTE<br /> <br />
        </div>
        <table border="1" style="width:40%;">
            <tr>
                <td>Identificación:</td>
                <td>
                    <asp:TextBox ID="TextBox1" runat="server" type="number"
                    required="required" placeholder="Identificación del cliente"
                    maxlength = "10" title="Ingrese solo hasta 10
                    números"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td>Nombre:</td>
                <td>
                    <asp:TextBox ID="TextBox2" runat="server" required="required"
                    maxlength="50" placeholder="Nombre del cliente" pattern="[A-Za-z
                    ]+" title="Ingrese solo letras"></asp:TextBox>
                </td>
            </tr>
            <tr>
                <td>Apellido:</td>
                <td>
                    <asp:TextBox ID="TextBox3" runat="server" required="required"
                    maxlength="50" placeholder="Apellido del cliente" pattern="[A-
                    Za-z ]+" title="Ingrese solo letras"></asp:TextBox>
                </td>
            </tr>
        </table>
    </form>
</body>
</html>
```

```

</tr>
<tr>
  <td>Fijo(Opcional):</td>
  <td>
    <asp:TextBox ID="TextBox4" runat="server" maxlength="12"
      placeholder="Telefono Fijo del cliente" pattern="[0-9]+"
      title="Ingrese solo numeros"></asp:TextBox>
  </td>
</tr>
<tr>
  <td>Celular(Opcional):</td>
  <td>
    <asp:TextBox ID="TextBox5" runat="server" maxlength="10"
      placeholder="Telefono Movil del cliente" pattern="[0-9]+"
      title="Ingrese solo numeros"></asp:TextBox>
  </td>
</tr>
</table>
<br />
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
<br />
<br />
<asp:Button ID="Button_Guardar" runat="server" Text="Guardar"
  OnClick="Button_Guardar_Click" />
</center>
</form>
</body>
</html>

```



Paso 11. En la carpeta llamada “ModuloCliente” agregue un nuevo WebForm, colóquele como nombre “ActualizarEliminarCliente.aspx”, realice el diseño y escriba el código que se muestra a continuación:

ACTUALIZAR/ELIMINAR CLIENTE

Identificación:

Nombre:	<input type="text"/>
Apellido:	<input type="text"/>
Fijo(Opcional):	<input type="text"/>
Celular(Opcional):	<input type="text"/>

Label

	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 16 de 22		

```

using System.Data;
using PresentacionWeb.Scripts;
using Logica;

namespace PresentacionWeb.ModuloCliente
{
    public partial class ActualizarEliminarCliente : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Page.IsPostBack == false)
            {
                Panel1.Visible = false;
            }
        }

        protected void Button_Consultar_Click(object sender, EventArgs e)
        {
            Cliente ObjCliente = new Cliente();

            try
            {
                DataSet DatosCliente
                ObjCliente.ConsultarCliente(TextBox1.Text, "Identificacion");

                DataTable DatosConsultados =
                DatosCliente.Tables["DatosConsultados"];

                int numregistros = DatosConsultados.Rows.Count;

                if (numregistros == 0)
                {
                    MessageBox.alert("No existe en la Base de Datos Cliente con esta
                    identificación");



                    Label1.Text = "No existe en la Base de Datos Cliente con esta
                    identificación";

                    Panel1.Visible = false;
                }
                else
                {
                    Panel1.Visible = true;

                    TextBox2.Text = DatosConsultados.Rows[0]["Nombre"].ToString();
                    TextBox3.Text = DatosConsultados.Rows[0]["Apellido"].ToString();
                    TextBox4.Text = DatosConsultados.Rows[0]["Fijo"].ToString();
                    TextBox5.Text = DatosConsultados.Rows[0]["Celular"].ToString();

                    Label1.Text = "Datos Consultados Con Exito";
                }
            }
        }
    }
}

```


	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 18 de 22		

```

protected void Button_Eliminar_Click(object sender, EventArgs e)
{
    Cliente ObjCliente = new Cliente();

    try
    {
        bool respuestaSQL = ObjCliente.EliminarCliente(TextBox1.Text);

        if (respuestaSQL == true)
        {
            MessageBox.alert("Los datos de este cliente fueron Eliminados
correctamente");

            Label1.Text = "Los datos de este cliente fueron Eliminados
correctamente";

            TextBox1.Text = ""; TextBox2.Text = ""; TextBox3.Text = "";
            TextBox4.Text = ""; TextBox5.Text = "";
        }
        else
        {
            MessageBox.alert(ObjCliente.Mensaje);
            Label1.Text = ObjCliente.Mensaje;
        }
    }

    catch (Exception Ex)
    {
        MessageBox.alert("Error!: " + Ex.Message + " " + ObjCliente.Mensaje);
        Label1.Text = "Error!: " + Ex.Message + " " + ObjCliente.Mensaje;
    }

    Panel1.Visible = false;
}
}
}

```

IMPORTANTE: Para validar campos en ASP.NET se pueden hacer uso de las propiedades que ofrece html5, en este caso en la sección de código de etiquetas del WebForm agregue la referencia a la hoja de estilos llamada "campoinvalido.css" y las propiedades a cada uno de los TextBox para que el código quede tal y como se muestra a continuación:

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Actualizar-Eliminar</title>
    <link href=" ../Estilos/campoinvalido.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <form id="form1" runat="server">
    <div>
    <center>

```




```
DataSet DatosCliente = new DataSet();



switch(DropDownList1.SelectedValue)
{
    case "Identificación":
    {
        DatosCliente = ObjCliente.ConsultarCliente(TextBox1.Text,
            "Identificacion");
        break;
    }
    case "Nombre":
    {
        DatosCliente = ObjCliente.ConsultarCliente(TextBox1.Text,
            "Nombre");
        break;
    }
    case "Apellido":
    {
        DatosCliente = ObjCliente.ConsultarCliente(TextBox1.Text,
            "Apellido");
        break;
    }
    case "Ver Todos":
    {
        DatosCliente = ObjCliente.ConsultarCliente("", "Todas");
        break;
    }
    default:
    {
        MessageBox.alert("Seleccione una opción valida");

        OpcionValida = false;
        break;
    }
}

if (OpcionValida == true)
{
    DataTable DatosConsultados =
    DatosCliente.Tables["DatosConsultados"];

    int numregistros = DatosConsultados.Rows.Count;

    if (numregistros == 0)
    {
        MessageBox.alert("No existe en la Base de Datos Cliente
            con el valor buscado");
    }
    else
    {
        GridView1.DataSource = DatosConsultados;
        GridView1.DataBind();
    }
}
```

	GUÍA DE TRABAJO N° 6 - DESARROLLO WEB		
	Educación Media Fortalecida SED/SENA	Programación de Software GRADO 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 22 de 22		

```

    }
  }
}

catch (Exception ex)
{
  MessageBox.alert("No se ha podido efectuar la consulta debido al
siguiente Error: " + ex.Message);
}
}
}
}

```

TALLER

Usando la misma base de datos que ya tiene desarrollada transforme cada uno de los módulos de su proyecto SENA a su versión Web ASP.NET.