Unidad 06 - Laboratorio ABM

1. Creación Formulario de ABM

Objetivos

Crear el formulario de ABM y todos sus controles.

Aclaraciones

Se presentará una manera de realizar el ABM de usuario para su ejercitación.

Aviso

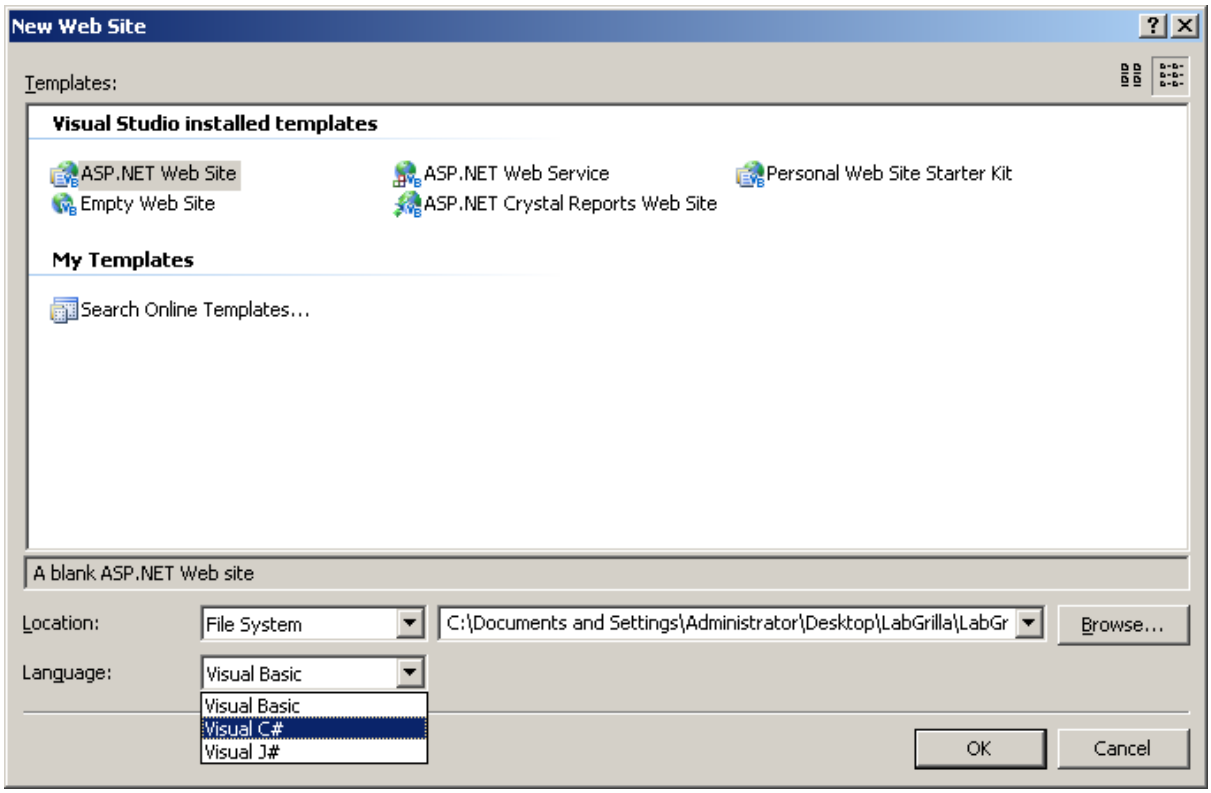
Para realizar este laboratorio necesita utilizar la base de datos academia con la tabla de usuarios. Si se encuentra en el laboratorio del Departamento de Sistemas la misma se encuentra disponible en el servidor del departamento de sistemas serverisi. Si realiza este laboratorio desde otra ubicación antes de realizar los pasos indicados a continuación realice los pasos indicados en el Anexo – Creación de la base de datos

Duración Aproximada

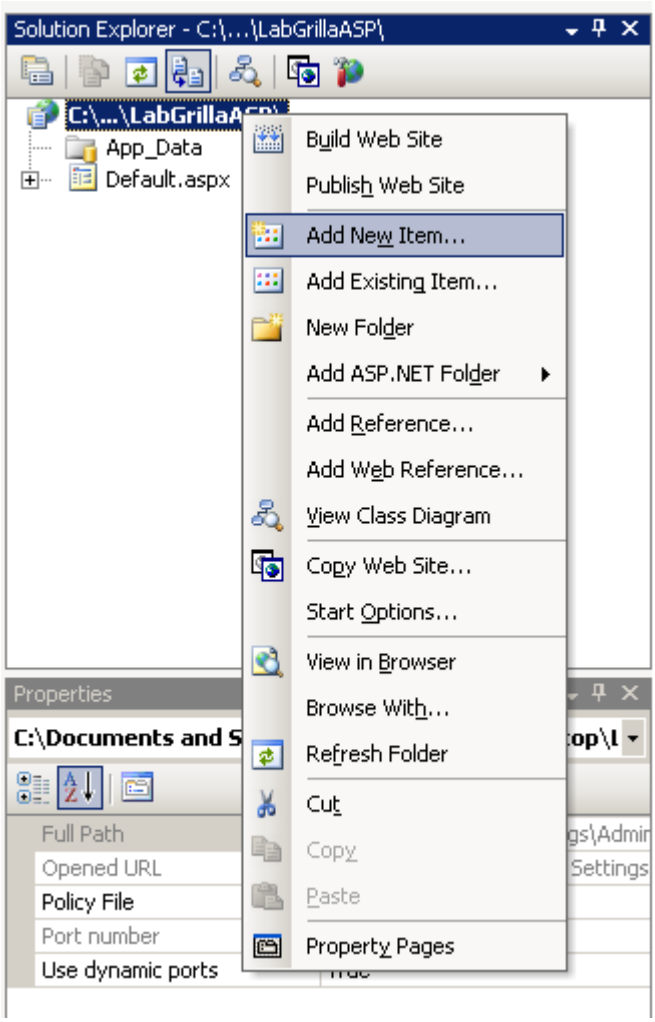
5 minutos

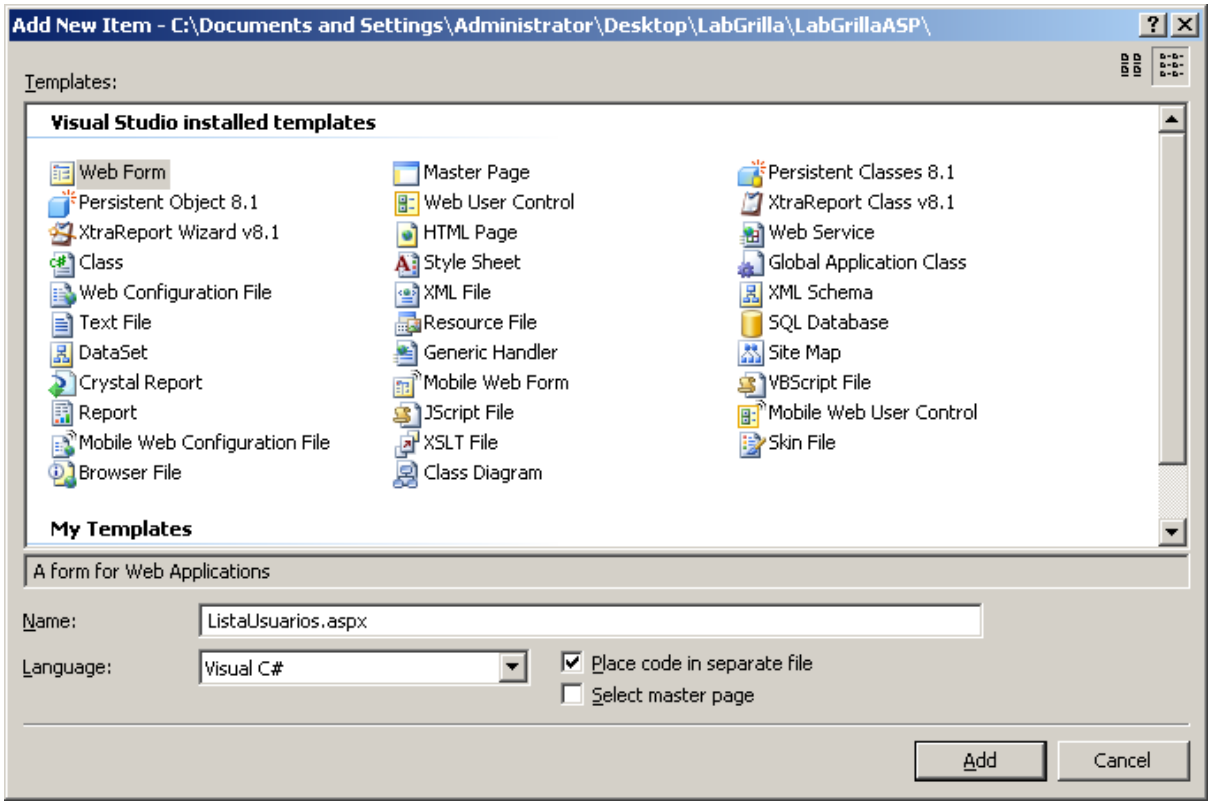
Pasos

1) Crear un nuevo proyecto de tipo Web Site (ASP.NET Web Site) llamado LabABM. Antes de crearlo, debemos elegir el lenguaje de programación que vamos a utilizar.

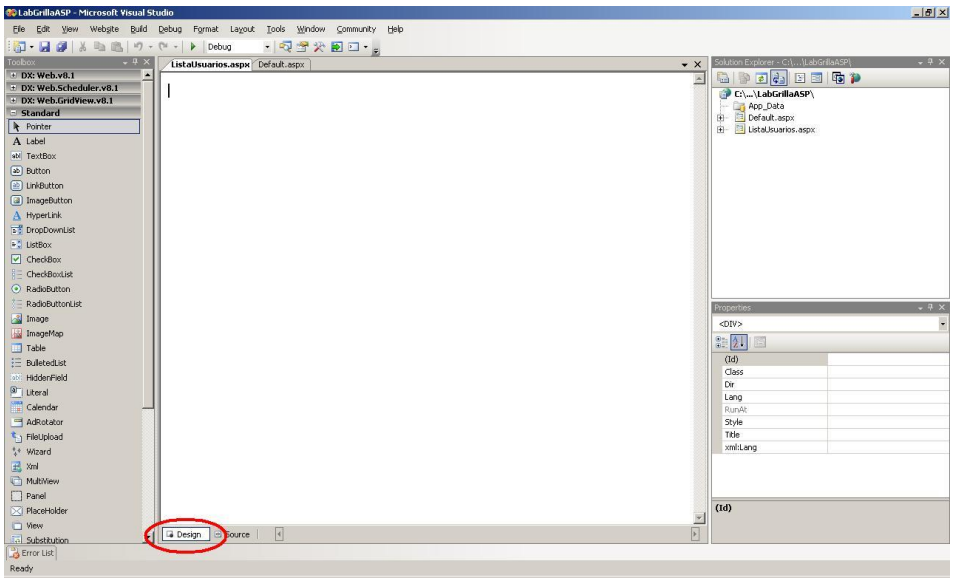


2) Creamos un nuevo Formulario Web (Web Form) desde el explorador de soluciones y lo llamamos ListaUsuarios.aspx

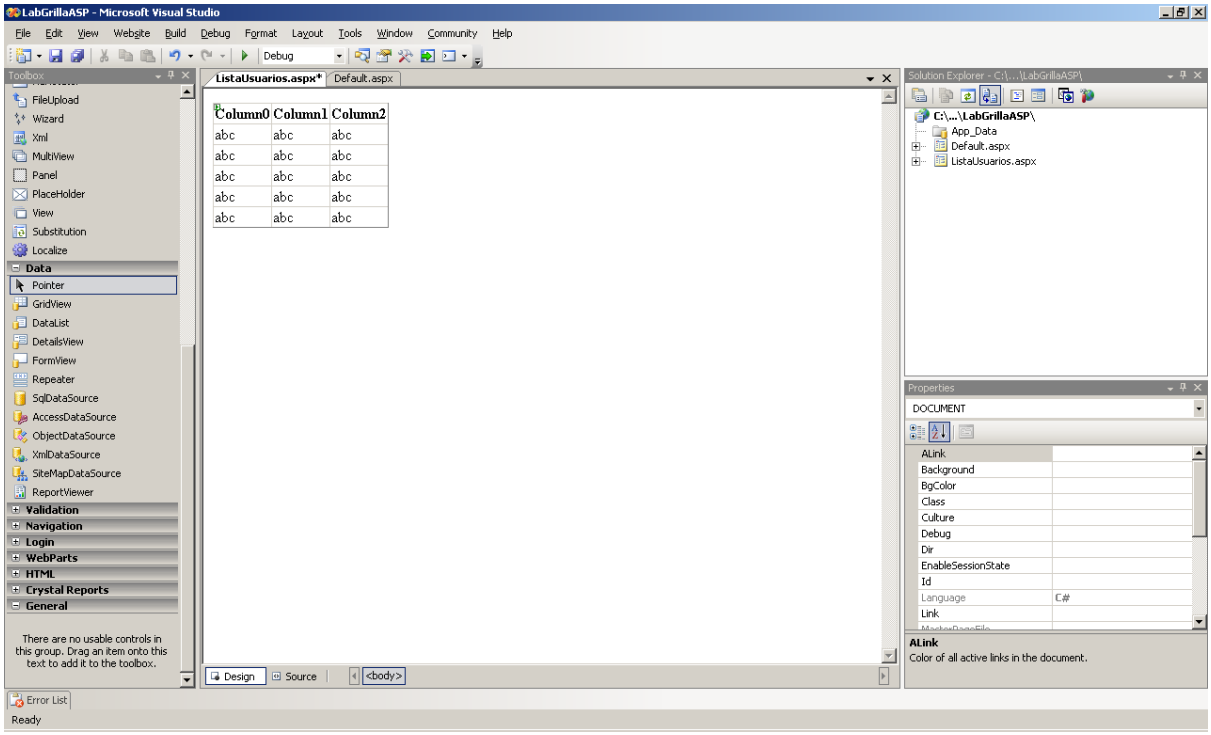




3) Una vez creado el formulario, nos posicionamos en la vista de Diseño.

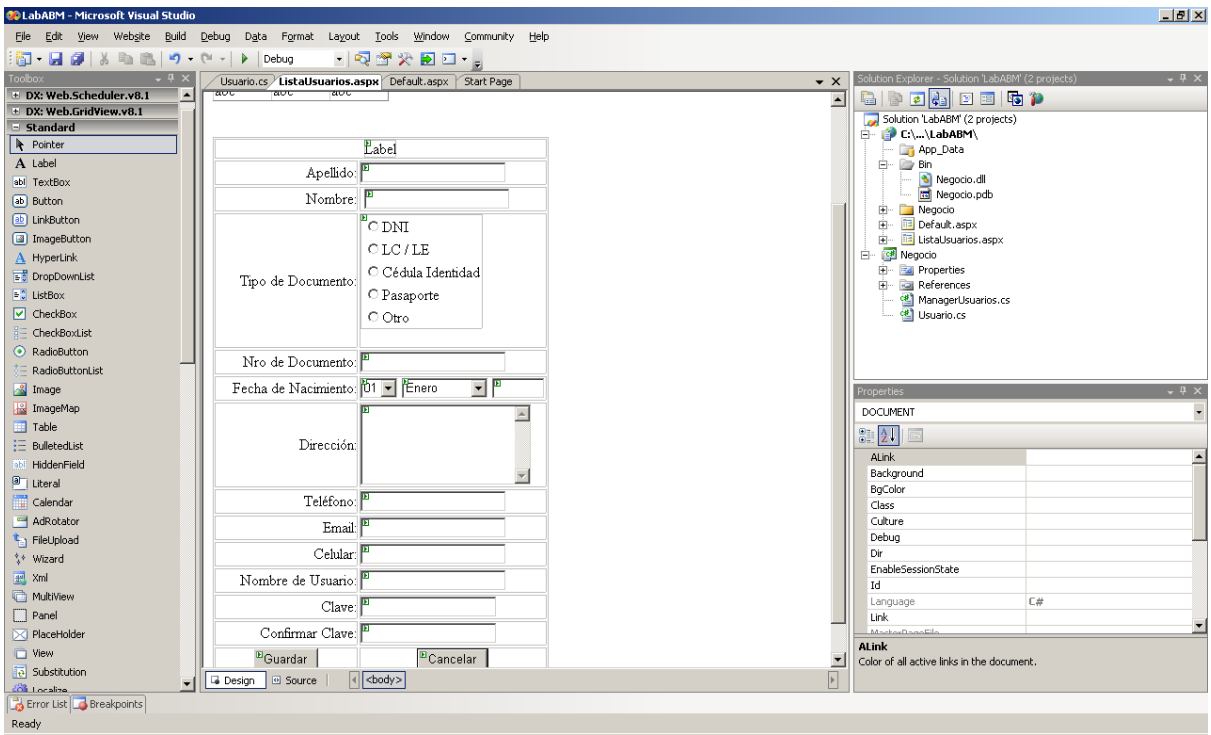


4) Desde el Cuadro de herramientas arrastramos el control GridView (por lo general se encuentra dentro de la Categoría **Data** del cuadro de herramientas). El resultado será algo similar a:



5) Cambiar el nombre de la grilla por grdUsuarios.

6) Crear los campos para el alta y edición del Usuario, para ello crear una tabla HTML (Categoría **HTML** del cuadro de herramientas) de 2 columnas y 14 filas, como se muestra a continuación:



Para los controles utilizar las propios de .NET que se encuentran en la Categoría **Standard** del cuadro de Herramientas. La tabla debería quedar algo similar a:

```
<table border="1">
            <tr>
                <td align="center" colspan="2">
                    <asp:Label ID="lblAccion" runat="server" Text="Label"></asp:Label></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Apellido:</td>
                <td>
                    <asp:TextBox ID="txtApellido" runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Nombre:</td>

                <td>
                     <asp:TextBox ID="txtNombre" runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Tipo de Documento:</td>
                <td>
                    <asp:RadioButtonList ID="rblTipoDocumento" runat="server">
                        <asp:ListItem Value="1">DNI</asp:ListItem>
                        <asp:ListItem Value="2">LC / LE</asp:ListItem>
                        <asp:ListItem Value="3">C&#233;dula Identidad</asp:ListItem>
                        <asp:ListItem Value="4">Pasaporte</asp:ListItem>
                        <asp:ListItem Value="5">Otro</asp:ListItem>
                    </asp:RadioButtonList></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Nro de Documento:</td>
                <td>
                    <asp:TextBox ID="txtNroDocumento" runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Fecha de Nacimiento:</td>
                <td>
                    <asp:DropDownList ID="ddlDiaFechaNacimiento" runat="server">
                    </asp:DropDownList>
                    <asp:DropDownList ID="ddlMesFechaNacimiento" runat="server">
                        <asp:ListItem>Enero</asp:ListItem>
                        <asp:ListItem>Febrero</asp:ListItem>
                        <asp:ListItem>Marzo</asp:ListItem>
                        <asp:ListItem>Abril</asp:ListItem>
                        <asp:ListItem>Mayo</asp:ListItem>
                        <asp:ListItem>Junio</asp:ListItem>
                        <asp:ListItem>Julio</asp:ListItem>
                        <asp:ListItem>Agosto</asp:ListItem>
                        <asp:ListItem>Septiembre</asp:ListItem>
                        <asp:ListItem>Octubre</asp:ListItem>
                        <asp:ListItem>Noviembre</asp:ListItem>
                        <asp:ListItem>Diciembre</asp:ListItem>
                    </asp:DropDownList>
                    <asp:TextBox ID="txtAnioFechaNacimiento" runat="server" MaxLength="4"
Width="50px"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Dirección:</td>
                <td>
                    <asp:TextBox ID="txtDirección" runat="server" Rows="5"
TextMode="MultiLine"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Teléfono:</td>
                <td>
                    <asp:TextBox ID="txtTelefono" runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Email:</td>
                <td>
                    <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Celular:</td>
                <td>
                    <asp:TextBox ID="txtCelular" runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Nombre de Usuario:</td>
```

```
                <td>
                    <asp:TextBox ID="txtNombreUsuario" runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Clave:</td>
                <td>
                    <asp:TextBox ID="txtClave" runat="server" TextMode="Password"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="right">
                    Confirmar Clave:</td>
                <td>
                    <asp:TextBox ID="txtConfirmarClave" runat="server"
TextMode="Password"></asp:TextBox></td>
            </tr>
            <tr>
                <td style="width: 150px" align="center">
                    <asp:Button ID="btnGuardar" runat="server" Text="Guardar" /></td>
                <td align="center">
                    <asp:Button ID="btnCancelar" runat="server" Text="Cancelar" /></td>
            </tr>
        </table>
```

7) Para cargar el combo de los días de la Fecha de Nacimiento, utilizar una función que contenga un bucle y cargue los números del 01 al 31.
8) La etiqueta lblAccion contendrá el texto "Editar Usuario *<idUsuario>*" o "Agregar Nuevo Usuario" dependiendo del estado de la página.

2. Crear Origen de Datos de la Grilla

Objetivos

Definir las clases de origen de datos de la Grilla. Para este Laboratorio utilizaremos el modelo conectado de ADO.NET.

Duración Aproximada

30 minutos

Pasos

1) Crear un nuevo proyecto en la solución de tipo Class Library llamado Negocio.
2) En el explorador de soluciones cambiamos el nombre de Class1 a ManagerUsuarios.
3) En el proyecto LabABM Agregamos una referencia al proyecto Negocio.
   Para ello hacemos clic con el botón derecho sobre el nodo References del Proyecto LabABM y hacemos clic en Agregar Referencia…
   Allí vamos a la pestaña Proyectos, seleccionamos el proyecto Negocio y hacemos clic en Aceptar.

4) Creamos una clase Usuario que contiene las siguientes variables y propiedades:

```
namespace Negocio
{
    public class Usuario
    {
        private int _id;
        private Nullable<int> _tipoDoc;
        private Nullable<int >_nroDoc;
        private string _fechaNac;
        private string _apellido;
        private string _nombre;
        private string _direccion;
        private string _telefono;
        private string _email;
        private string _celular;
```

```
        private string _usuario;
        private string _clave;


        public int Id
        {
            get { return _id; }
            set { _id = value; }
        }
        public Nullable<int> TipoDoc
        {
            get { return _tipoDoc; }
            set { _tipoDoc = value; }
        }
        public Nullable<int> NroDoc
        {
            get { return _nroDoc; }
            set { _nroDoc = value; }
        }
        public string FechaNac
        {
            get { return _fechaNac; }
            set { _fechaNac = value; }
        }
        public string Apellido
        {
            get { return _apellido; }
            set { _apellido = value; }
        }
        public string Nombre
        {
            get { return _nombre; }
            set { _nombre = value; }
        }
        public string Direccion
        {
            get { return _direccion; }
            set { _direccion = value; }
        }
        public string Telefono
        {
            get { return _telefono; }
            set { _telefono = value; }
        }
        public string Email
        {
            get { return _email; }
            set { _email = value; }
        }
        public string Celular
        {
            get { return _celular; }
            set { _celular = value; }
        }
        public string NombreUsuario
        {
            get { return _usuario; }
            set { _usuario = value; }
        }
        public string Clave
        {
            get { return _clave; }
            set { _clave = value; }
        }

    }
}
```

5) En la Clase ManagerUsuarios agregamos un using System.Data; y using System.Data.SqlClient;
6) Luego agregamos la propiedad Conn como sigue:

```
   private SqlConnection _conn;

   protected SqlConnection Conn
   {
        get { return _conn; }
     set { _conn = value; }
   }
```

Esta propiedad sólo es necesaria porque no implementamos la capa de acceso a datos por separado.

7) Creamos entonces el constructor de la clase ManagerUsuarios:

```
public ManagerUsuarios()
{
    this.Conn = new SqlConnection(
    "Data Source=serverisi;Initial Catalog=academia;Integrated Security=false;user=net;password=net;");
            /*
             * Este connection string es para conectarse con la base de datos academia en el servidor
             * del departamento sistemas desde una PC de los laboratorios de sistemas,
             * si realiza este Laboratorio desde su PC puede probar el siguiente connection string
             *
             * "Data Source=localhost;Initial Catalog=academia;Integrated Security=true;"
             *
             * Si realiza esta práctica sobre el MS SQL SERVER 2005 Express Edition entonce debe
             * utilizar el siguiente connection string
             *
             * "Data Source=localhost\SQLEXPRESS;Initial Catalog=academia;Integrated Security=true;"
             */

}
```

8) Luego en la clase ManagerUsuarios del proyecto Negocio agregamos el Método GetAll() como sigue para obtener la lista de Usuarios con sus datos

```
        public List<Usuario> GetAll()
        {
            //Creo la lista en la que le voy a ir agregando los usuarios
            List<Usuario> listaUsuarios = new List<Usuario>();
            //Defino una variable de tipo Usuario
            Usuario usuarioActual;

            //Creo el comando para ejecutar la sentencia SQL, le asocio la Conexión también
            SqlCommand cmdGetUsuarios = new SqlCommand("select * from usuarios", this.Conn);

            //Abro la conexión
            this.Conn.Open();

            //Ejecuto la consulta, me devuelve un objeto SqlDataReader
            SqlDataReader rdrUsuarios = cmdGetUsuarios.ExecuteReader();

            //Recorro el SqlDataReader, transformo el registro a objeto y
            //agrego ese objeto a la lista de usuarios
            while (rdrUsuarios.Read())
            {
                usuarioActual = new Usuario();
                usuarioActual.Id = (int)rdrUsuarios["id"];
                usuarioActual.TipoDoc = (Nullable<int>)rdrUsuarios["tipo_doc"];
                usuarioActual.NroDoc = (Nullable<int>)rdrUsuarios["nro_doc"];
                usuarioActual.FechaNac = rdrUsuarios["fecha_nac"].ToString();
                usuarioActual.Apellido = rdrUsuarios["apellido"].ToString();
                usuarioActual.Nombre = rdrUsuarios["nombre"].ToString();
                usuarioActual.Direccion = rdrUsuarios["direccion"].ToString();
                usuarioActual.Telefono = rdrUsuarios["telefono"].ToString();
                usuarioActual.Email = rdrUsuarios["email"].ToString();
                usuarioActual.Celular = rdrUsuarios["celular"].ToString();
                usuarioActual.NombreUsuario = rdrUsuarios["usuario"].ToString();
                usuarioActual.Clave = rdrUsuarios["clave"].ToString();

                //Agrego el objeto a la lista de usuarios
                listaUsuarios.Add(usuarioActual);

            }

            //Cierro la conexión
            this.Conn.Close();

            //Devuelvo la Lista de Usuarios
            return listaUsuarios;
        }
```

Este método devuelve una lista de objetos Usuario según el resultado de la consulta a la tabla de Usuarios.

9) Creamos también los métodos para actualizar, insertar y eliminar usuarios:

```
        public void BorrarUsuario(Usuario usuarioActual)
        {
            //Creo el comando para ejecutar la sentencia SQL de DELETE,
            //le asocio la Conexión también
            SqlCommand cmdDeleteUsuario = new SqlCommand(" DELETE FROM usuarios WHERE id=@id ", this.Conn);

            //Le agrego los parámetros necesarios
            cmdDeleteUsuario.Parameters.Add(new SqlParameter("@id", usuarioActual.Id.ToString()));

            //Abro la Conexión
            this.Conn.Open();
            //Ejecuto la instrucción SQL de DELETE
            cmdDeleteUsuario.ExecuteNonQuery();

            //Cierro la Conexión
            this.Conn.Close();
        }

        public void ActualizarUsuario(Usuario usuarioActual)
        {
            //Creo el comando para ejecutar la sentencia SQL de DELETE,
            //le asocio la Conexión también
            SqlCommand cmdActualizarUsuario = new SqlCommand(" UPDATE usuarios " +
                                                " SET tipo_doc = @tipo_doc, nro_doc = @nro_doc, fecha_nac =
@fecha_nac, " +
                                                " apellido = @apellido, nombre = @nombre, direccion =
@direccion, " +
                                                " telefono = @telefono, email = @email, celular = @celular,
usuario = @usuario, " +
                                                " clave = @clave WHERE id=@id ", this.Conn);

            //Le agrego los parámetros necesarios
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@tipo_doc",
usuarioActual.TipoDoc.ToString()));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@nro_doc",
usuarioActual.NroDoc.ToString()));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@fecha_nac", usuarioActual.FechaNac));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@apellido", usuarioActual.Apellido));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@nombre", usuarioActual.Nombre));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@direccion", usuarioActual.Direccion));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@telefono", usuarioActual.Telefono));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@email", usuarioActual.Email));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@celular", usuarioActual.Celular));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@usuario", usuarioActual.NombreUsuario));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@clave", usuarioActual.Clave));
            cmdActualizarUsuario.Parameters.Add(new SqlParameter("@id", usuarioActual.Id.ToString()));

            //Abro la Conexión
            this.Conn.Open();
            //Ejecuto la instrucción SQL de UPDATE
            cmdActualizarUsuario.ExecuteNonQuery();
            //Cierro la Conexión
            this.Conn.Close();
        }

        public void AgregarUsuario(Usuario usuarioActual)
        {
            //Creo el comando para ejecutar la sentencia SQL de DELETE,
            //le asocio la Conexión también
            SqlCommand cmdInsertarUsuario = new SqlCommand(" INSERT INTO
usuarios(tipo_doc,nro_doc,fecha_nac,apellido, " +
                                                " nombre,direccion,telefono,email,celular,usuario,clave) " +
                                                " VALUES
(@tipo_doc,@nro_doc,@fecha_nac,@apellido,@nombre,@direccion, " +
                                                " @telefono,@email,@celular, @usuario, @clave  )",
this.Conn);

            //Le agrego los parámetros necesarios
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@tipo_doc",
usuarioActual.TipoDoc.ToString()));
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@nro_doc",
usuarioActual.NroDoc.ToString()));
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@fecha_nac", usuarioActual.FechaNac));
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@apellido", usuarioActual.Apellido));
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@nombre", usuarioActual.Nombre));
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@direccion", usuarioActual.Direccion));
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@telefono", usuarioActual.Telefono));
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@email", usuarioActual.Email));
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@celular", usuarioActual.Celular));
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@usuario", usuarioActual.NombreUsuario));
            cmdInsertarUsuario.Parameters.Add(new SqlParameter("@clave", usuarioActual.Clave));

            //Abro la Conexión
```

```
            this.Conn.Open();
            //Ejecuto la instrucción SQL de INSERT
            cmdInsertarUsuario.ExecuteNonQuery();
            //Cierro la Conexión
            this.Conn.Open();
        }
```

3. Mostrar datos en Grilla

Objetivos

Mostrar los datos en la grilla.

Duración Aproximada

45 minutos

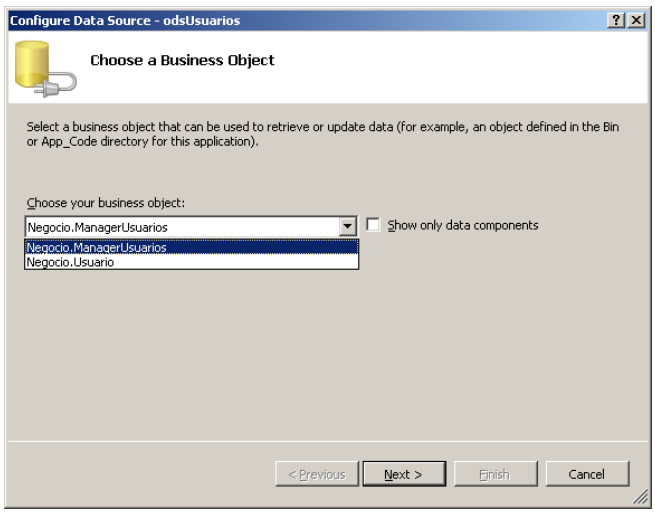
Aclaraciones

Se utilizará un objeto ObjectDataSource para recuperar la información proveniente desde el Proyecto Negocio.

Pasos

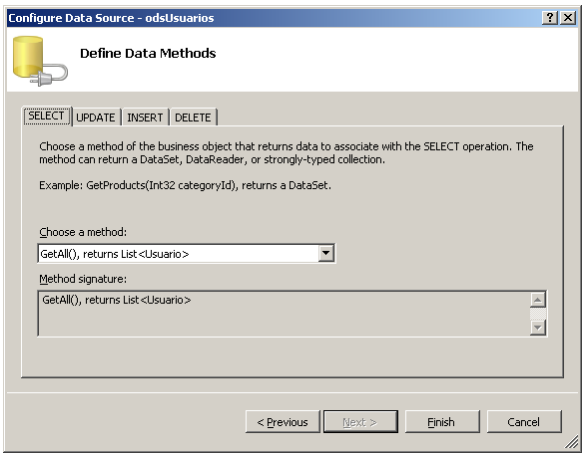
1) Desde el Cuadro de herramientas arrastramos el control ObjectDataSource hacia el Formulario (por lo general se encuentra dentro de la Categoría **Data** del cuadro de herramientas).
2) Cambiar el nombre del ObjectDataSource a odsUsuarios.
3) Abrir el SmartTag del ObjectDataSource para configurarlo:

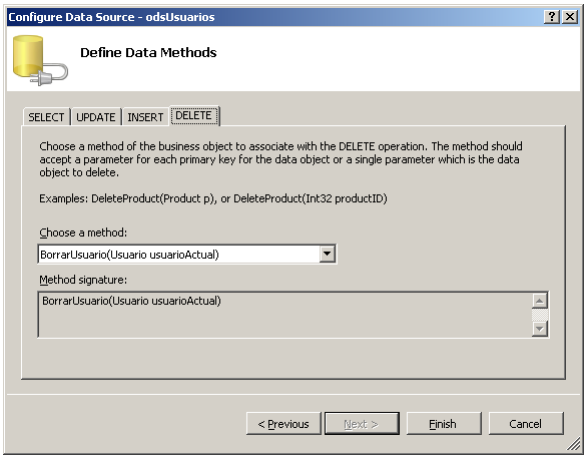
4) Nos abrirá una Wizard con varias pantallas. En la primera hay que seleccionar el objeto de negocio, en este caso sería Negocio.ManagerUsuarios:



5) Una vez seleccionado, clickear en el botón Siguiente. Nos aparecerá otra pantalla, en la cual tenemos que configurar los métodos de la clase

ManagerUsuarios para acceder a los datos. Los métodos SELECT (GetAll) y DELETE (BorrarUsuario) son los que vamos a configurar::





6) Clickeamos en el botón Finalizar para concluir la configuración y proceder a enlazar este ObjectDataSource a la grilla. Para ello clickeamos en el SmartTag de la grilla y en el combo que dice correspondiente al DataSource, seleccionamos odsUsuarios.

4. Programar funcionalidad para ABM

Objetivos

Configurar la grilla y programar el ABM.

Duración Aproximada

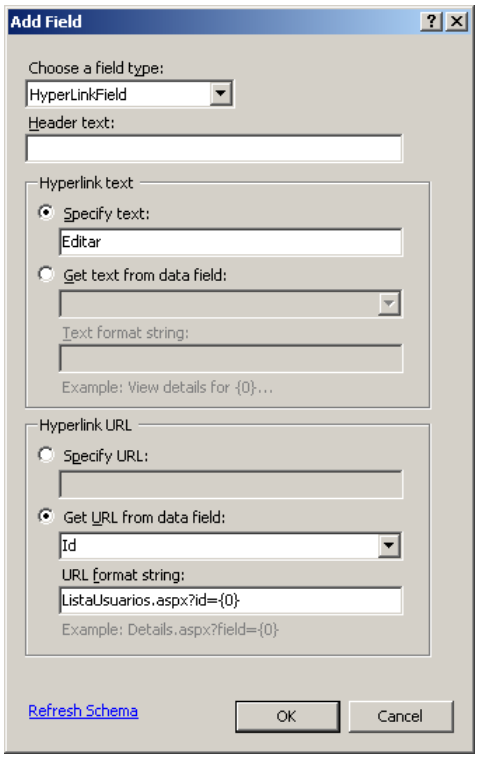
90 minutos

Pasos

1) Definir las siguientes columnas de la grilla y su orden dentro de la misma:

	Id	Apellido	Nombre	NombreUsuario	FechaNac	Email	
Delete	0	abc	abc	abc	abc	abc	Editar
Delete	1	abc	abc	abc	abc	abc	Editar
Delete	2	abc	abc	abc	abc	abc	Editar
Delete	3	abc	abc	abc	abc	abc	Editar
Delete	4	abc	abc	abc	abc	abc	Editar

Para ello, habilitar en la grilla el ordenamiento de las columnas y el borrado. Posteriormente agregar una columna de tipo HyperLink, según se muestra a continuación:



2) Programar un método que cargue los días en el combo ddlDiaFechaNacimiento y llamarla desde el Load cuando la página se carga por primera vez, es decir cuando la propiedad Page.IsPostBack es False. La firma del método debe ser como sigue:

```
private void cargarDiasCalendario()
{
    //Cargar en el combo los items correspondientes a los días
    //(del 1 al 31)
}
```

3) La página constará de 2 estados (en Edición y en Alta) y, dependiendo de dichos estados, la tabla con los controles tendrá diferentes comportamientos. El estado por defecto (cuando se carga la página) será Alta. En el evento Load de la página validar el estado de la misma. Utilizar el siguiente método:

```
private bool PaginaEnEstadoEdicion()
{
    if (Request.QueryString["id"] != null)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

- Edición (cuando la variable de QueryString *id* tiene valor): cargará los valores del objeto Usuario en cuestión (según el valor de la variable *id*) en los controles pertinentes. Para obtener el usuario, agregar un método en la clase ManagerUsuarios del proyecto Negocio con la siguiente firma:

```
public Usuario GetUsuario(int idUsuario)
{
    //Basarse en el método GetAll pero obtener únicamente el usuario que viene
    //como párametro
}
```

Para cargar los datos del Usuario, utilizar el siguiente método:

```
private void CargarDatosUsuario(int idUsuario)
{
    // 1 - Obtener los datos del usuario en cuestión
    // 2 - Cargar en los controles de la tabla los datos del usuario obtenido
}
```

Modificar la etiqueta lblAccion con el texto "Editar Usuario <*idUsuario*>", donde el *idUsuario* es el valor de la variable de QueryString *id*

- Alta (cuando la variable de QueryString *id* NO tiene valor): modificar la etiqueta lblAccion con el texto "Agregar Nuevo Usuario"

4) Programar el botón "Guardar": en el evento *btnGuardar_Click* crear un objeto Usuario y asignarle los valores de los controles del formulario de la tabla de edición. Si el estado de la página es Edición, llamar al método ActualizarUsuario de ManagerUsuario. Si es Alta, llamar a AgregarUsuario.
5) Programar el botón "Cancelar": en el evento btnCancelar_Click redireccionar a la misma página para borrar el estado en que y cargar el estado de la página por default.
6) Seteamos el Formulario ListaUsuarios.aspx como página de inicio. Para esto en el Explorador de Soluciones, clickeamos el botón derecho del mouse sobre el formulario y seleccionamos la opción "Establecer como página de inicio".
7) Ejecutamos la aplicación y probamos.