

Unidad 06 - Laboratorio DataGridView

1. Creación Formulario de Listado

Objetivos

Crear el formulario de listado y todos sus controles

Aviso

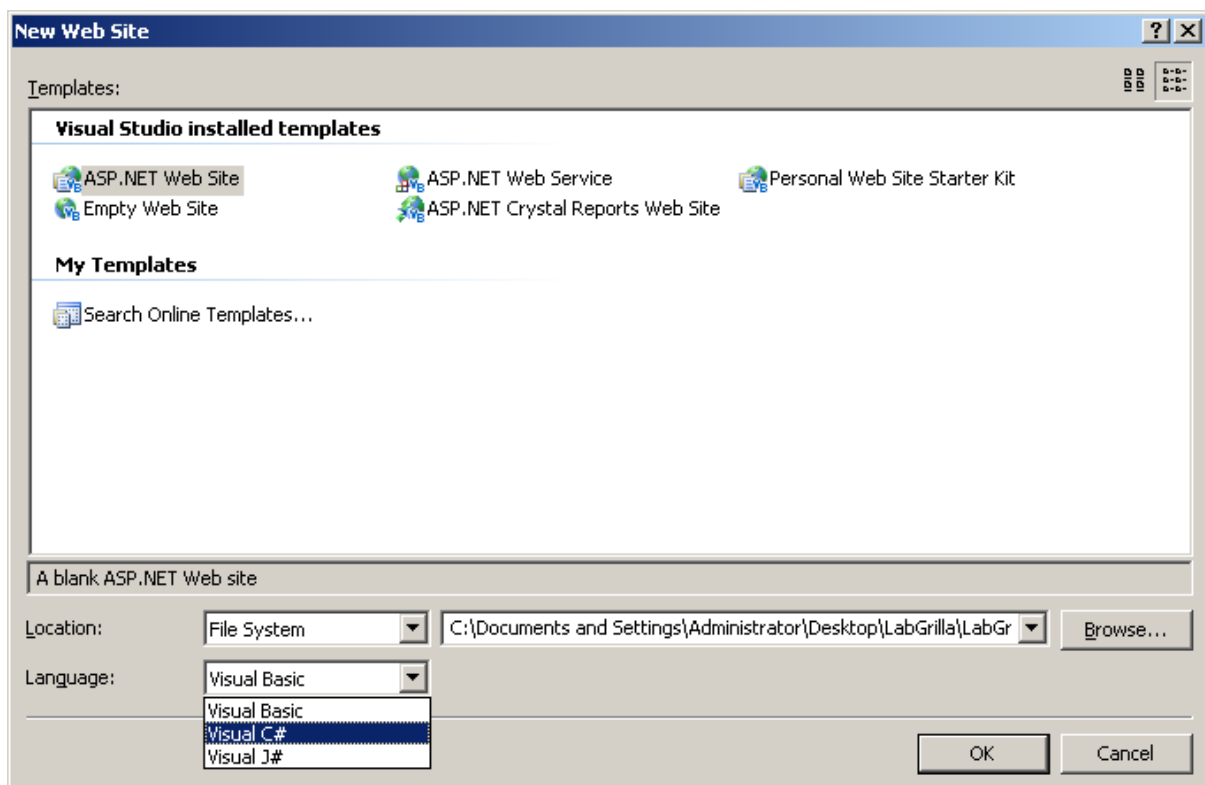
Para realizar este laboratorio necesita utilizar la base de datos academia con la tabla de usuarios. Si se encuentra en el laboratorio del Departamento de Sistemas la misma se encuentra disponible en el servidor del departamento de sistemas serverisi. Si realiza este laboratorio desde otra ubicación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

Duración Aproximada

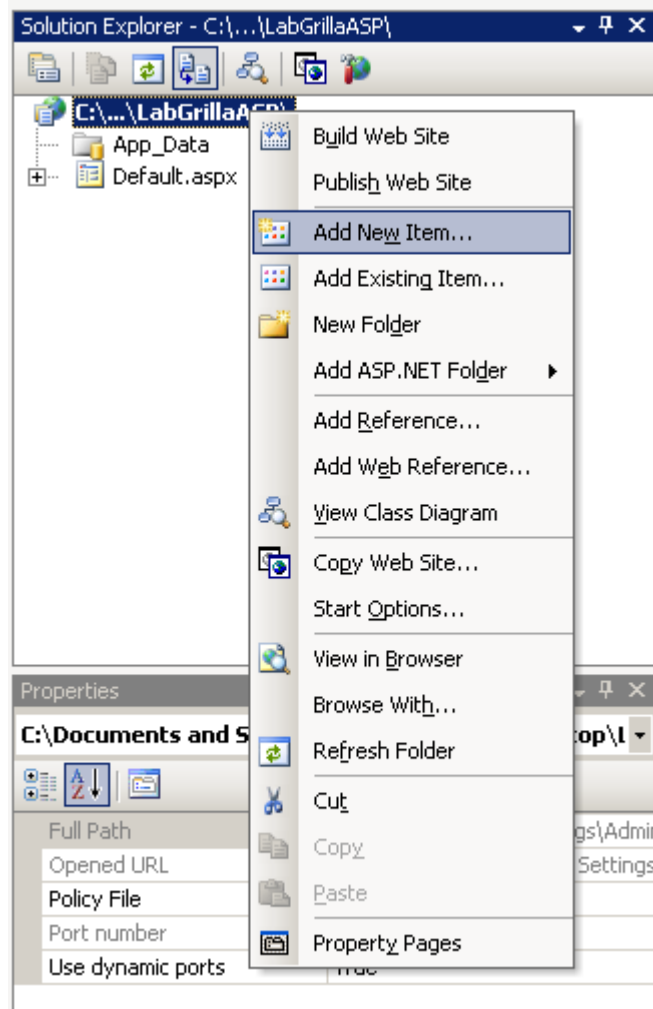
5 minutos

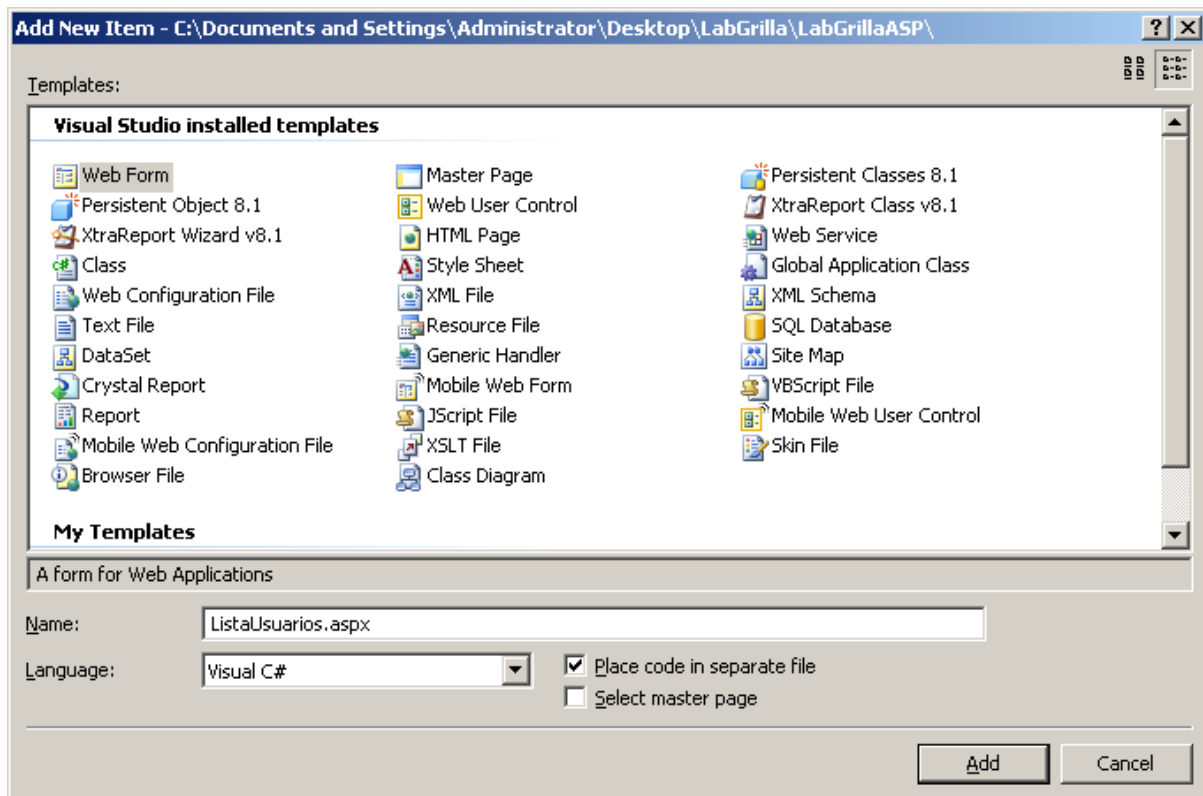
Pasos

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

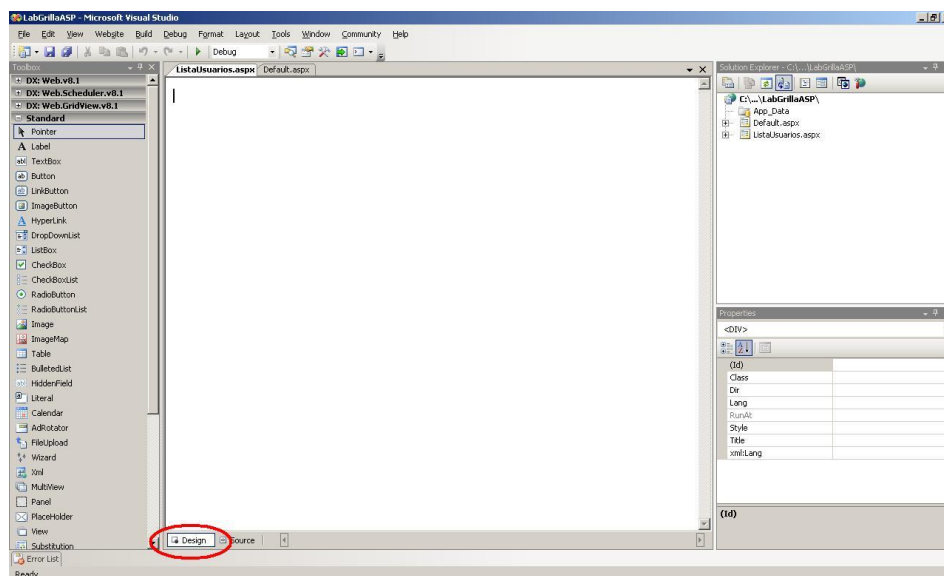


- 2) Creamos un nuevo Formulario Web (Web Form) desde el explorador de soluciones y lo llamamos ListaUsuarios.aspx

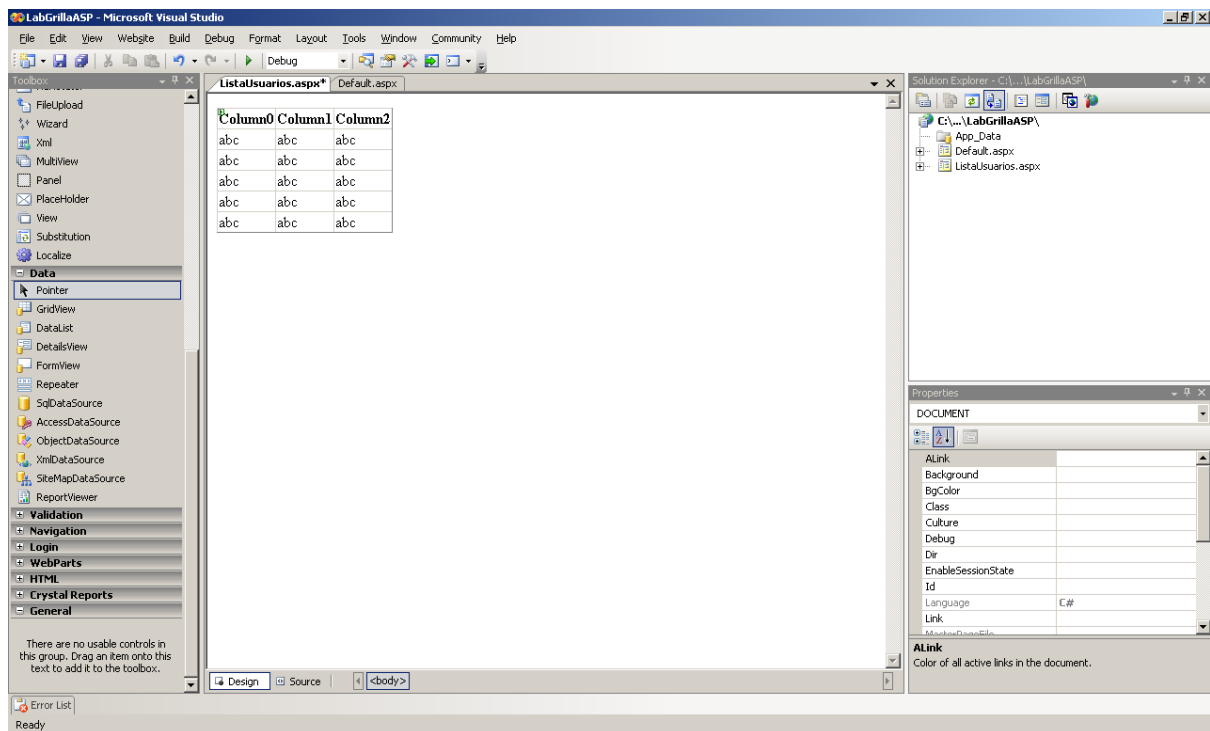




3) Una vez creado el formulario, nos posicionamos en la vista de Diseño.



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.

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 LabGrillaASP Agregamos una referencia al proyecto Negocio.
Para ello hacemos clic con el botón derecho sobre el nodo References del Proyecto LabGrillaASP 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 entonces 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 y guardar cambios

Objetivos

Mostrar los datos en la grilla.

Duración Aproximada

45 minutos

Aclaraciones

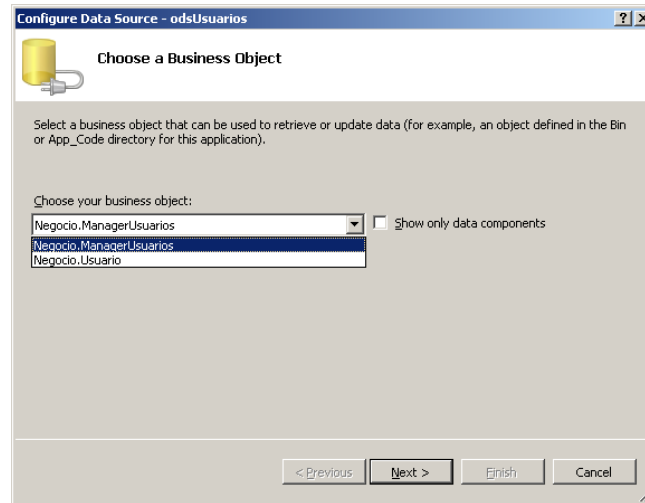
Se utilizará un objeto `ObjectDataSource` para recuperar la información proveniente desde el Proyecto Negocio y se utilizarán las funcionalidades nativas que provee la grilla para editar y borrar registros existentes. La grilla no tiene funcionalidad nativa para la inserción de registros por lo que se optó por customizarla para que incluya dicha funcionalidad. Existen diferentes formas de realizarlo, aquí se muestra solamente una.

Pasos

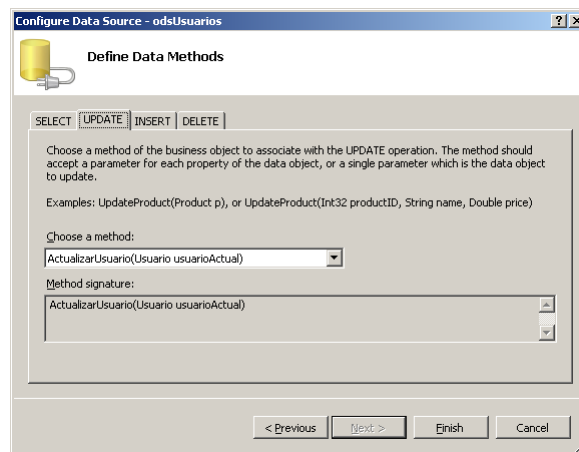
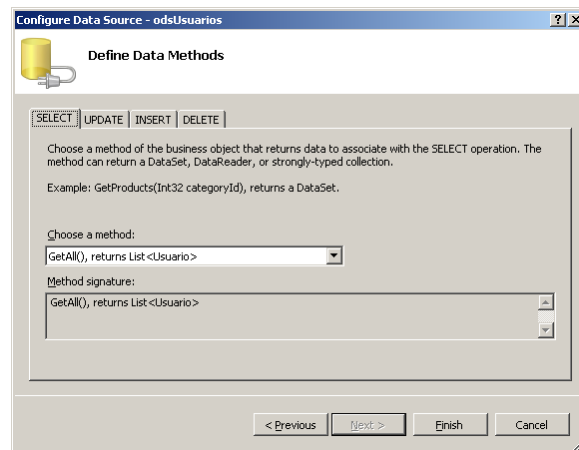
- 1) Desde el Cuadro de herramientas arrastramos el control `ObjectDataSource` hacia el Formulario (por lo general se encuentra dentro de la Categoría **Data** del cuadro de herramientas).
- 2) Cambiar el nombre del `ObjectDataSource` a `odsUsuarios`.
- 3) Abrir el SmartTag del `ObjectDataSource` para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

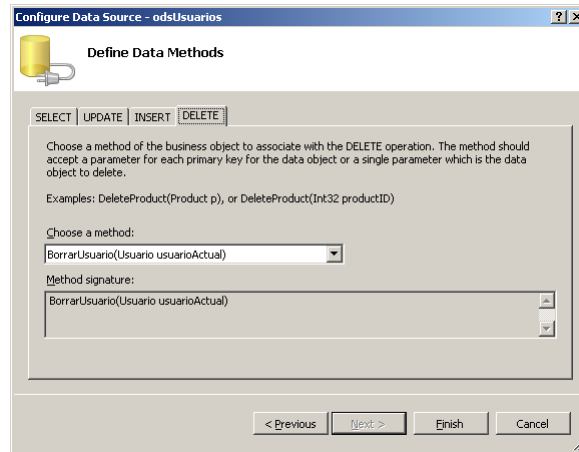


- 4) Nos abrirá una Wizard con varias pantallas. En la primera hay que seleccionar el objeto de negocio, en este caso sería `Negocio.ManagerUsuarios`:

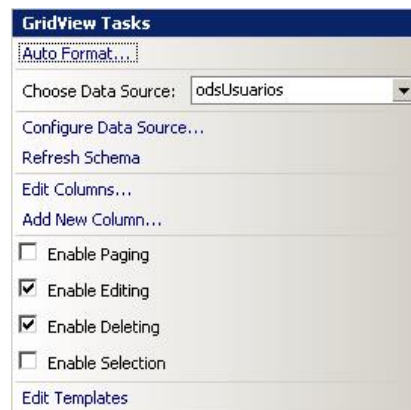


5) Una vez seleccionado, clicar en el botón Siguiente. Nos aparecerá otra pantalla, en la cual tenemos que configurar los métodos de la clase ManagerUsuarios para acceder a los datos. Los métodos SELECT (GetAll), UPDATE (ActualizarUsuario) y DELETE (BorrarUsuario) son los que vamos a configurar:

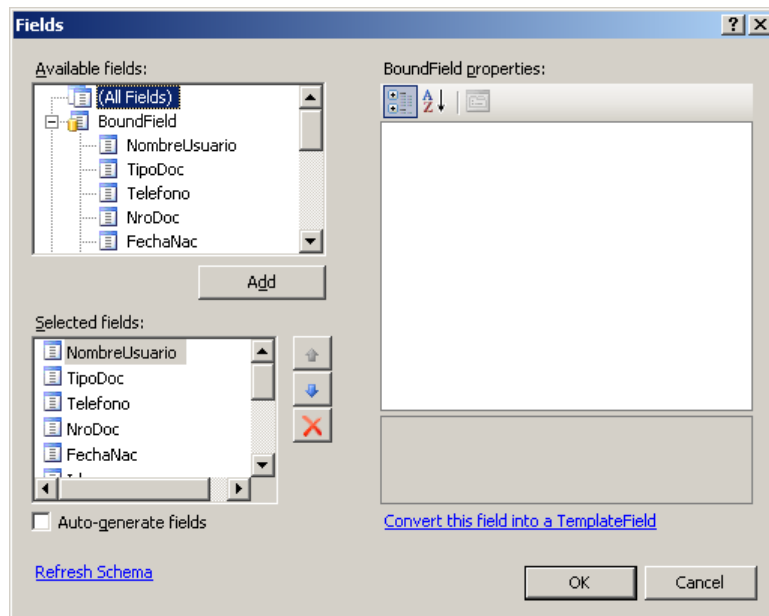




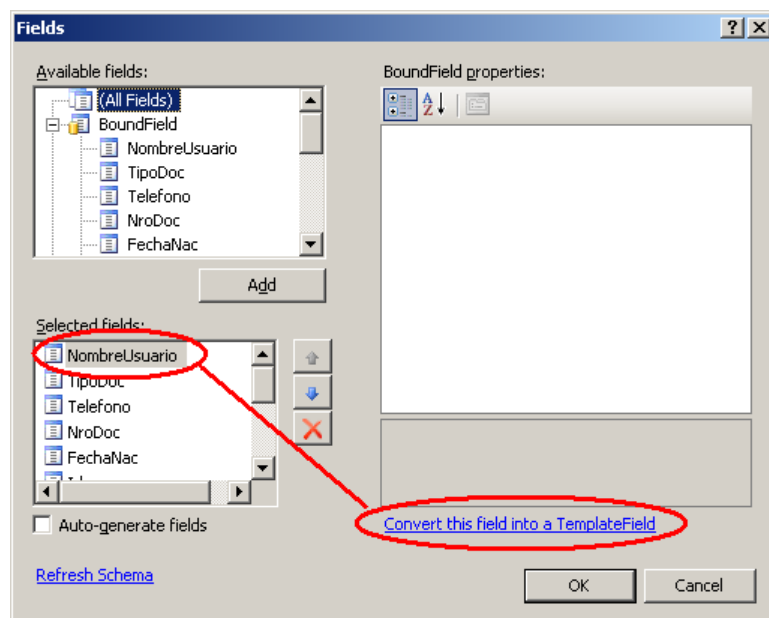
- 6) Clickeamos en el botón Finalizar para concluir la configuración y proceder a enlazar este ObjectDataSource a la grilla. Para ello clickeamos en el SmartTag de la grilla y en el combo que dice correspondiente al DataSource, seleccionamos odsUsuarios. Habilitamos también que permita Edición y Borrado:



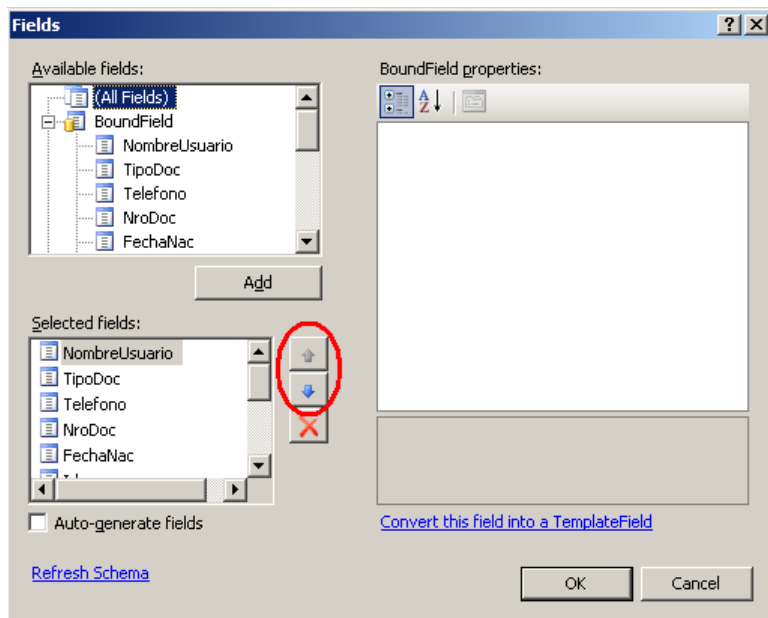
- 7) Posteriormente clickearemos en el link Editar Columnas y aparecerá una pantalla como la siguiente:



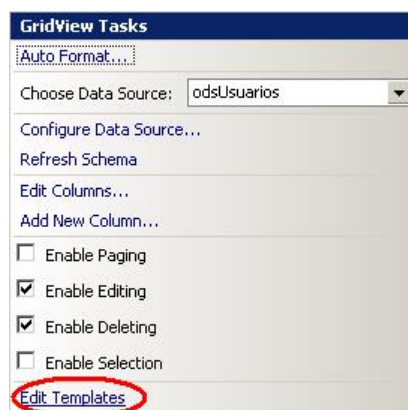
- 8) Por cada uno de los campos seleccionados, los convertiremos a campos de tipo TemplateFields:



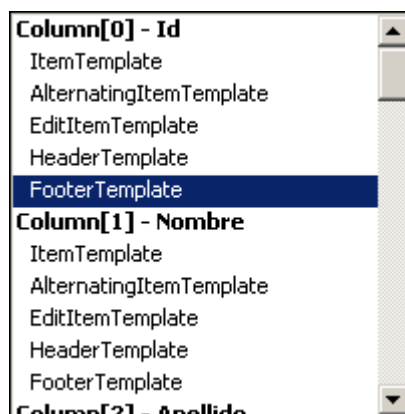
- 9) Luego acomodaremos los campos para que aparezcan en el siguiente orden: Id, Nombre, Apellido, TipoDoc, NroDoc, FechaNac, Direccion, Telefono, Email, Celular, nombreUsuario y Clave. Para ello utilizaremos las flechas que se muestran en la siguiente imagen:



- 10) En las propiedades de la grilla, setear el campo Id en la propiedad DataKeyNames y setear a True la propiedad ShowFooter.
- 11) Ir al SmartTag de la grilla y clickear en Editar Templates:

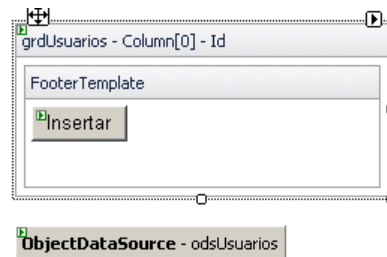


- 12) Desplegar el combo que aparece y seleccionar FooterTemplate de la Columna Id (debería ser la columna con índice 0):



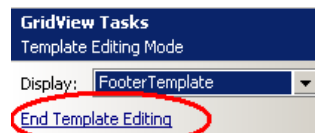
13) En la grilla ahora estamos editando el Footer de la columna Id. Vamos a arrastrar un botón desde la barra de Herramientas y le setearemos las siguientes propiedades:

- ID = btnInsertar
- CommandName = Insertar
- Text = Insertar



14) Para el resto de las columnas, haremos el mismo procedimiento de los 2 pasos anteriores pero en vez de arrastrar un botón, arrastraremos una caja de texto. A dicha caja le setearemos la propiedad ID y dependiendo de la columna, tendrán los siguientes nombre: txtNombre, txtApellido, txtTipoDoc, txtNroDoc, txtFechaNac, txtDireccion, txtTelefono, txtEmail, txtCelular, txtNombreUsuario, txtClave.

15) Una vez finalizado, terminar la edición del template, para ello, clickear en el SmartTag de la grilla en la opción correspondiente:

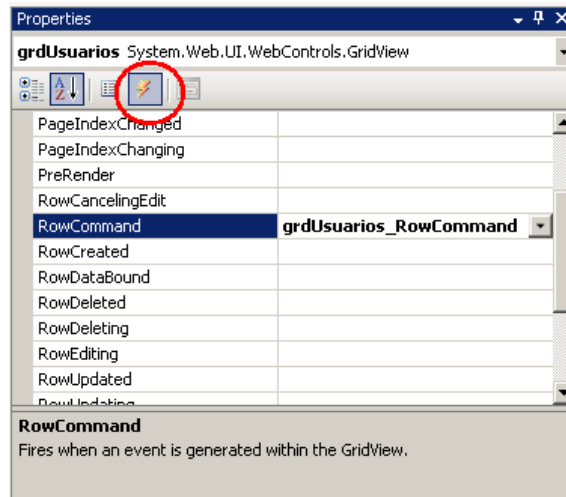


16) El formulario deberá quedar similar a la siguiente imagen:

	Id	Nombre	Apellido	TipoDoc	NroDoc	FechaNac	Direcc
Edit Delete	0	abc	abc	0	0	abc	abc
Edit Delete	1	abc	abc	1	1	abc	abc
Edit Delete	2	abc	abc	2	2	abc	abc
Edit Delete	3	abc	abc	3	3	abc	abc
Edit Delete	4	abc	abc	4	4	abc	abc
	Insertar						

ObjectDataSource - odsUsuarios

17) Lo que resta es programar el evento correspondiente para insertar un registro. Para ellos nos vamos a las propiedades de la grilla, en la parte de eventos y hacemos doble clic sobre la fila correspondiente al evento RowCommand:



18) Programo lo siguiente en dicho evento:

```
protected void grdUsuarios_RowCommand(object sender, GridViewCommandEventArgs e)
{
    //Verificamos que se ejecute lo siguiente solamente cuando lo produce
    //a este evento el botón Insertar
    if (e.CommandName.ToLower() == "insertar")
    {
        TextBox cajaTexto;
        string textoActual;
        Negocio.Usuario usuarioNuevo = new Negocio.Usuario();

        //Busco el control y se lo asigno a la propiedad correspondiente
        //del objeto Usuario
        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtNombre");
        textoActual = cajaTexto.Text;
        usuarioNuevo.Nombre = textoActual;

        //Así hago sucesivamente con el resto de los parámetros
        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtApellido");
        textoActual = cajaTexto.Text;
        usuarioNuevo.Apellido = textoActual;

        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtTipoDoc");
        textoActual = cajaTexto.Text;
        usuarioNuevo.TipoDoc = Convert.ToInt32(textoActual);

        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtNroDoc");
        textoActual = cajaTexto.Text;
        usuarioNuevo.NroDoc = Convert.ToInt32(textoActual);

        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtFechaNac");
        textoActual = cajaTexto.Text;
        usuarioNuevo.FechaNac = textoActual;

        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtDireccion");
        textoActual = cajaTexto.Text;
        usuarioNuevo.Direccion = textoActual;

        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtTelefono");
        textoActual = cajaTexto.Text;
        usuarioNuevo.Telefono = textoActual;

        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtEmail");
        textoActual = cajaTexto.Text;
        usuarioNuevo.Email = textoActual;

        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtCelular");
        textoActual = cajaTexto.Text;
        usuarioNuevo.Celular = textoActual;

        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtNombreUsuario");
        textoActual = cajaTexto.Text;
        usuarioNuevo.NombreUsuario = textoActual;

        cajaTexto = (TextBox)grdUsuarios.FooterRow.FindControl("txtClave");
        textoActual = cajaTexto.Text;
        usuarioNuevo.Clave = textoActual;
    }
}
```

```

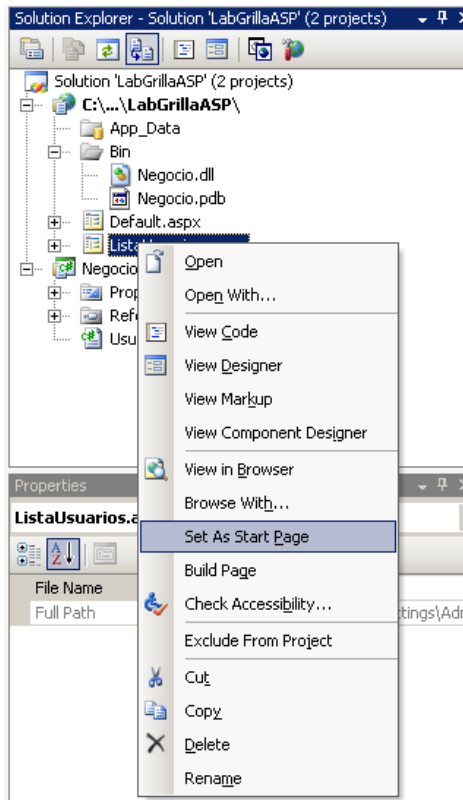
        //Defino una variable del Manager para ejecutar el evento de Insertar
        Negocio.ManagerUsuarios manager = new Negocio.ManagerUsuarios();

        //Agrego el Nuevo Usuario
        manager.AgregarUsuario(usuarioNuevo);

        //Actualizo la Grilla
        grdUsuarios.DataBind();
    }
}

```

- 19) 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".



- 20) Ejecutamos la aplicación y probamos. Entonces veremos una página web como la siguiente:

Untitled Page - Windows Internet Explorer

http://localhost:1037/LabGrillaASP/ListaUsuarios.aspx

File Edit View Favorites Tools Help

Untitled Page

Page Tools

	Id	Nombre	Apellido	TipoDoc	NroDoc	FechaNac	Direccion	Telefono
Edit Delete	1	Perez	Juana	1	11111111	11/11/1911 12:00:00 AM	A la vuelta de la esquina	111-1111
Edit Delete	2	Doe	Jhon	2	22222222	2/2/1902 12:00:00 AM	Lejos	222-2222
	<input type="button" value="Insertar"/>							

Local intranet 100%