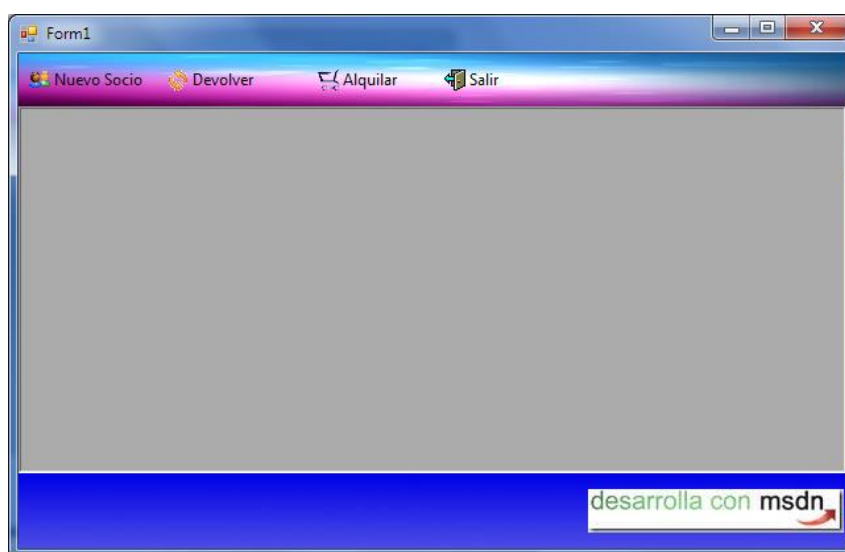




UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA INDUSTRIAL

GUIA DE LABORATORIO:

PROYECTO GESTION DE VIDEOS



CURSO: INGENIERIA DE SOFTWARE
PROFESOR: PONCE BENITES Wiler Arturo
Correo: ingenieria.software@industrial.unmsm.pe
wiler.ponce@industrial.unmsm.pe
Página Web: <http://campus.industrial.unmsm.edu.pe>

Este material de apoyo académico a sido elaborado por el profesor para uso exclusivo de los alumnos de la Facultad de Ingeniería Industrial de la Universidad Nacional Mayor de San Marcos y en concordancia con lo dispuesto por la legislación sobre derechos de autor: Decreto Legislativo 822.

Junio – 2017

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA INDUSTRIAL

Curso: Ingeniería de Software

Profesor: Ing. Wiler Arturo Ponce Benites

Correo: wiler.ponce@industrial.unmsm.pe

Página Web: <http://campusvirtualfii.unmsm.edu.pe>

Laboratorio: Diseño de un Proyecto de Software (Parte 1)

Se requiere diseñar y desarrollar un software para gestionar un Video Club, que tiene por finalidad alquilar películas a sus socios.

- 1.- El proyecto será desarrollado en plataforma C#
- 2.- El proyecto se gestionará usando una base de datos SQL Server 2008 Professional
- 3.- La base de datos está compuesta por tres tablas: Socios, Películas y Alquileres, con la siguiente estructura.

Tabla: Socios				
Nombre de Campo	Tipo	Tamaño	Obligatorio	Notas
Dni	nchar	10	Si	llave principal o primary key
Nombre	nvarchar	50	Si	
Apellidos	nvarchar	50	Si	
Telefono	nvarchar	13	No	
Email	nvarchar	50	Si	

Tabla: Películas				
Nombre de Campo	Tipo	Tamaño	Obligatorio	Notas
CodBarras	nchar	10	Si	llave principal o primary key
Titulo	nvarchar	200	Si	
Stock	int		Si	Valor Inicial = 0

Tabla: Alquileres				
Nombre de Campo	Tipo	Tamaño	Obligatorio	Notas
AlquilerID	int		Si	Primary key, Autonumerico
SocioDNI	nchar	10	Si	
CodBarras	nchar	10	Si	
FechaAlquiler	datetime		Si	
Devoluta	bit		Si	Valor inicial = 0

Requisitos:

Se deberá tener instalado en la computadora:

- Visual Studio 2010 – Ultimate
- SQL Server 2008 - Professional

Crear una carpeta para alojar el proyecto y todos sus componentes. Ejemplo: **GestionVideos**

Procedimiento:

1.- Abrir Visual Studio 2010 – Ultimate:

Inicio/Microsoft Visual Studio 2010

2.- Seleccionar New Project

En la ventana Recent Template, seleccionar Visual C#, y al desplegarse la ventana hacer clic sobre Windows Forms Application

En el textbox **Name**, digitar: **MsdnVideo**

Hacer clic en el botón **Browse** y navegar por el explorador hasta hallar la carpeta **Lab01**, seleccionarla.

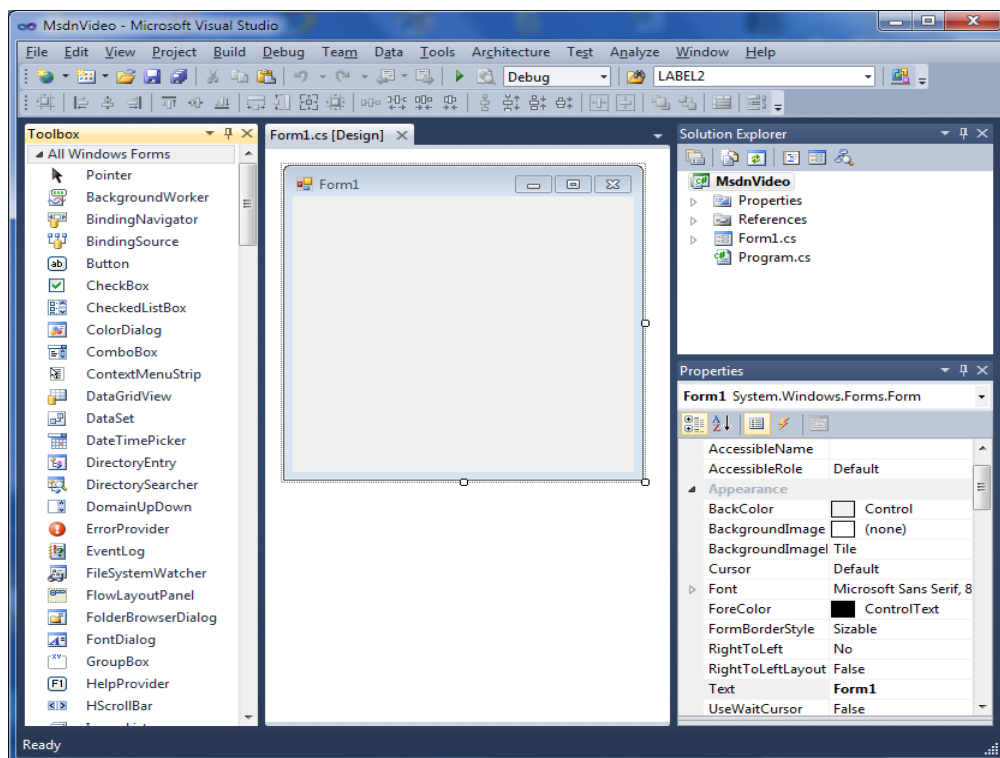
Las casillas o checkbox:

- ☐ Create directory for solution y
- ☐ Add to Source Control

No debe ser seleccionadas, si tuviera check hacer clic para desaparecerla.

Luego hacer clic en el botón OK

Como resultado nos encontramos en la ventana de desarrollo de Visual Studio 2010



Se observa, una ventana compuesta por:

- ☐ La Barra de título
- ☐ Tres barras de comandos
- ☐ Ventana de Herramientas: Toolbox
- ☐ Ventana de explorador de la solución: Solution Explorer
- ☐ Ventana de propiedades: Properties
- ☐ Ventana de formulario, don el formulario recién creado Form1.cs (Design)

Creación de la Base de Datos:

Haciendo uso de las siguientes instrucciones SQL-Transact crear la base de datos BD:

Abrir SQL Server

Abrir nueva consulta

```
/* SISTEMA: GESTION DE VIDEOS
   FUNCION: SCRIPT PARA LA CREACION DE LA BASE DE DATOS
   FECHA   : 01/02/2016
   AUTOR   : ING. WILER PONCE BENITES
*/

-- Paso 1.- Poner en uso la base de datos: master
USE master
GO

-- Paso 2.- Crear la Base de Datos
create database BD
ON
primary
    (name = BD,
     filename =
'I:\CompacPresario900\BaseDatos\GestionVideos\DB.mdf',
     size = 3MB,
     maxsize = 20,
     filegrowth = 1)
Log on
    (name = ventasllog,
     filename =
'I:\CompacPresario900\BaseDatos\GestionVideos\DB_log.ldf',
     size = 1MB,
     maxsize = 10,
     filegrowth = 1)
GO

-- Paso 3.- Poner en uso la Base de Datos creada, en este caso BD
USE [BD]
GO

-- Paso 4.- Asignar propietario a la Base de Datos: BD
use BD
EXEC sp_changedbowner 'sa';

use BD
exec sp_changedbowner 'USER02'

-- Paso 5.- Creacion de la tabla Socios
CREATE TABLE [dbo].[Socios](
    [Dni] [nchar](10) NOT NULL,
    [Nombre] [nvarchar](50) NOT NULL,
    [Apellidos] [nvarchar](50) NOT NULL,
    [Telefono] [nvarchar](13) NULL,
    [Email] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Socios] PRIMARY KEY CLUSTERED
    (
        [Dni] ASC
    )
WITH
    (
        PAD_INDEX = OFF,
```

```

        STATISTICS_NORECOMPUTE = OFF,
        IGNORE_DUP_KEY = OFF,
        ALLOW_ROW_LOCKS = ON,
        ALLOW_PAGE_LOCKS = ON
    )
    ON [PRIMARY]
)
    ON [PRIMARY]
GO

-- Paso 6.- Creacion de la tabla Peliculas
CREATE TABLE [dbo].[Peliculas](
    [CodBarras] [nchar](10) NOT NULL,
    [Titulo] [nvarchar](200) NOT NULL,
    [Stock] [int] NOT NULL,
    CONSTRAINT [PK_Peliculas] PRIMARY KEY CLUSTERED
    (
        [CodBarras] ASC
    )
WITH
    (
        PAD_INDEX = OFF,
        STATISTICS_NORECOMPUTE = OFF,
        IGNORE_DUP_KEY = OFF,
        ALLOW_ROW_LOCKS = ON,
        ALLOW_PAGE_LOCKS = ON
    )
    ON [PRIMARY]
)
    ON [PRIMARY]
GO

-- Paso 7.- Creación de la tabla Alquileres
CREATE TABLE [dbo].[Alquileres](
    [AlquilerID] [int] IDENTITY(1,1) NOT NULL,
    [SociosDNI] [nchar](10) NOT NULL,
    [CodBarras] [nchar](10) NOT NULL,
    [FechaAlquiler] [datetime] NOT NULL,
    [Devuelta] [bit] NOT NULL,
    CONSTRAINT [PK_Alquileres] PRIMARY KEY CLUSTERED
    (
        [AlquilerID] ASC
    )
WITH
    (
        PAD_INDEX = OFF,
        STATISTICS_NORECOMPUTE = OFF,
        IGNORE_DUP_KEY = OFF,
        ALLOW_ROW_LOCKS = ON,
        ALLOW_PAGE_LOCKS = ON
    )
    ON [PRIMARY]
)
    ON [PRIMARY]
GO

-- Paso 8.- Creación de las Relaciones y Diagrama de Relaciones
ALTER TABLE [dbo].[Alquileres] ADD CONSTRAINT
[DF_Alquileres_Devuelta] DEFAULT ((0)) FOR [Devuelta]
ALTER TABLE [dbo].[Peliculas] ADD CONSTRAINT [DF_Peliculas_Stock]
DEFAULT ((0)) FOR [Stock]
ALTER TABLE [dbo].[Alquileres] WITH CHECK ADD CONSTRAINT
[FK_Alquileres_Peliculas] FOREIGN KEY([CodBarras])
REFERENCES [dbo].[Peliculas] ([CodBarras])

```

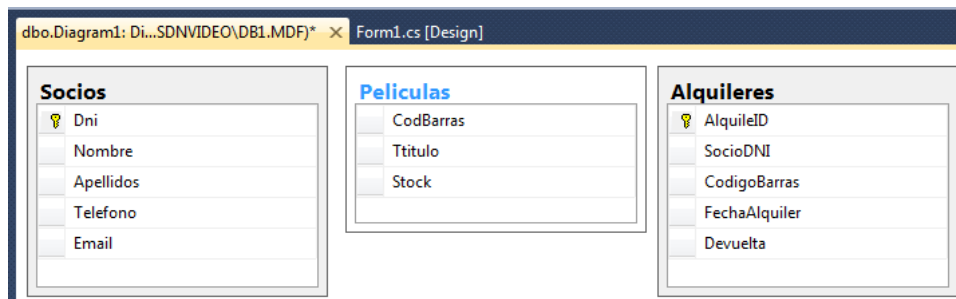
```

ALTER TABLE [dbo].[Alquileres] CHECK CONSTRAINT
[FK_Alquileres_Peliculas]
ALTER TABLE [dbo].[Alquileres] WITH CHECK ADD CONSTRAINT
[FK_Alquileres_Socios] FOREIGN KEY ([SociosDNI])
REFERENCES [dbo].[Socios] ([Dni])
ALTER TABLE [dbo].[Alquileres] CHECK CONSTRAINT [FK_Alquileres_Socios]
GO

```

Diagrama de Base de Relaciones

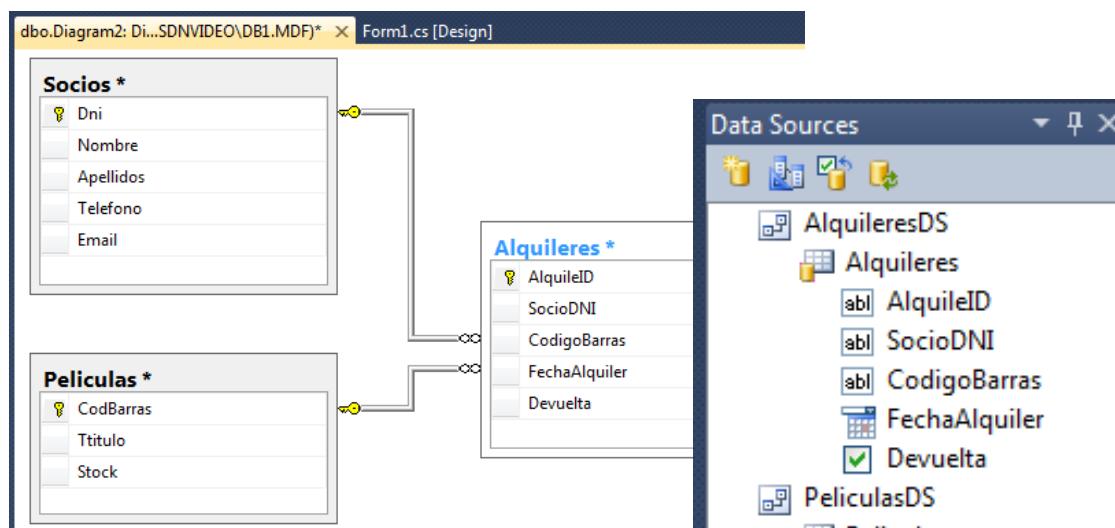
Para crear el **diagrama de relaciones** entre las tablas, se procede de la siguiente manera: Ubicamos el cursor en el icono **Database Diagrams** de la ventana **Server Explorer** y damos clic derecho, seleccionar la opción **Add New Diagram**, nos muestra una ventana con la lista de tablas definidas en nuestro proyecto, las seleccionamos y damos clic en el botón **Add**. Adicionamos en este caso las tres tablas y cerramos la ventana, en la ventana principal se muestran las tablas seleccionadas según se observa en la siguiente figura.



Las distribuimos de acuerdo a nuestras necesidades, debemos de saber que las **tablas “padre”** son **Socios** y **Películas** mientras que la tabla **Alquileres** será la tabla **“hijo”**. Para relacionar las tablas hacemos clic sobre el campo de relación, en nuestro caso son:

- DNI de la Tabla Socios con SocioDNI de la tabla Alquileres
- CodigoBarras de la Tabla Películas con Codigo de Barras de la tabla Alquileres.

Para conseguir esto, hacemos una operación de arrastre desde el campo de una de las tablas al campo relacionado de la otra tabla y obtendremos el siguiente esquema:



Grabar el diagrama, le damos el Inombre sugerido por default.

Generación de DataSet

Para crear un **Data Set**, vamos al **menú de comando** y seleccionamos:

Data / Add New Data Source

En la pantalla que nos muestra seleccionamos **Database**, clic en botón **Next** ahora elegimos **Dataset** clic en botón **Next**, nos muestra otra ventana: **Data Source Configuration Wizard**.

En esta pantalla hacemos clic en el **combobox** y nos despliega una lista de bases de datos, seleccionamos que hemos creado: **DB.mdf**, hacemos clic en el botón **Next**, otra vez **Next** y ahora nos muestra en la ventana: **Tablas, Views, Stored Procedures y Functions**, en nuestro caso vamos a desarrollar tres data set uno para cada tabla de nuestro proyecto.

Usar los siguientes nombres:

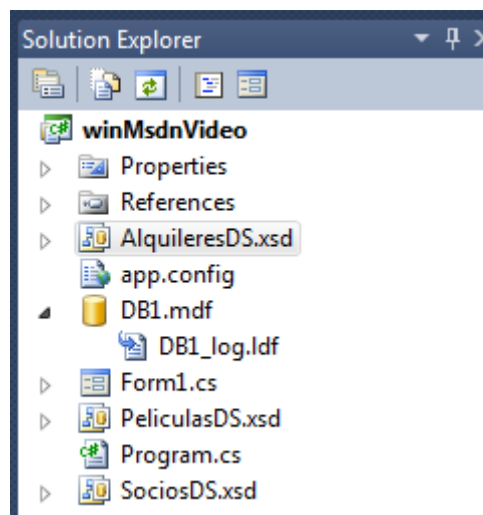
AlquileresDS
PeliculasDS
SociosDS

Nota: Para abrir la ventana **Data Sources** ejecutar la siguiente secuencia en la barra de comandos:

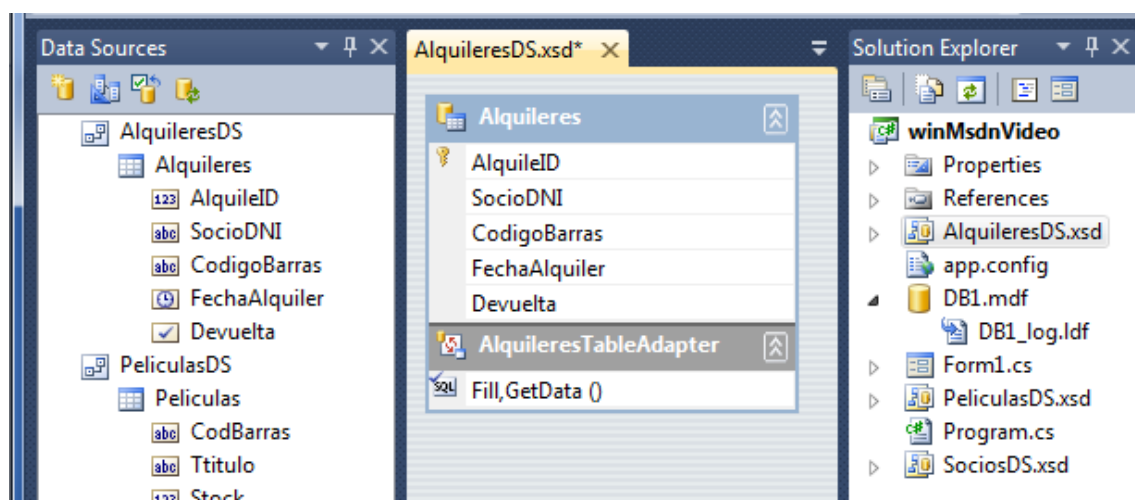
Data / Show Data Source

Crear un **DataSet** por cada una de las tablas que hemos considerado para este proyecto

Y al lado derecho se observa, también un objeto para estos **Data Set**, Guarde el proyecto previamente y luego puede explorar para fijar el concepto de la plataforma de desarrollo



Si damos doble clic en **AlquileresDS.xsd**, por ejemplo, nos muestra la siguiente pantalla



Podemos trabajar con los Data Set en vista diseño, por ejemplo me permitiría modificar los nombres de los campos que por defecto ha recuperado de la **BD.mdf**, estos tienen su tipo (ubicarse sobre el campo y dar clic derecho seleccionar **Properties**, podemos recuperar datos, actualizar datos, eliminar datos, entre otros procesos.

WPB/2014

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA INDUSTRIAL

Curso: Ingeniería de Software

Profesor: Ing. Wiler Arturo Ponce Benites

Correo: wiler.ponce@industrial.unmsm.pe

Página Web: <http://campusvirtualfii.unmsm.edu.pe>

Laboratorio: Diseño de un Proyecto de Software (Parte 2)

En este laboratorio, continuaremos el desarrollo del proyecto propuesto en la Parte 1, para ello abrimos el proyecto ya creado.

Objetivo:

- 1.- Diseñar un formulario plantilla para que en nuestro proyecto todos los formularios asociados tengan la misma presentación.
- 2.- Crear un formulario Padre con una Barra de Menú.

Recomendación:

Crear una carpeta, de nombre por ejemplo: **Iconos e Imágenes**, en él almacenar iconos relacionados a personas, a films o películas, operaciones tales como comprar, hacer compras, devoluciones

Procedimiento:

Ahora lo que vamos hacer es crear un **formulario “padre”** en el que estarán contenidos todos los demás formularios de nuestro proyecto

Sobre el formulario **Form1** (modo diseño) hacemos clic derecho y seleccionamos **Properties** y luego buscar la propiedad **IsMdiContainer** le colocamos **True**, tal que los nuevos formularios van a aparecer dentro de este.

Vamos a la ventana **Toolbox** y seleccionamos **barra de herramientas**, seleccionar:

Menus & Toolbars / MenuStrip y dar doble clic

En el formulario se carga una herramienta con la que crearemos la barra de herramientas que contendrá las siguientes opciones: **Nuevo Socio / Devolver / Alquilar / Salir**

Luego asociaremos imágenes y actualizar las propiedades de cada opción según el siguiente cuadro:

Propiedad	Valor u opcion elegida
Image	Explorar e importar la imagen seleccionada
ImageAlign	MiddleLeft
ImageScaling	None
Text	Elegir de acuerdo a nuestros requerimiento
TextAlign	MiddleLeft
AutoSize	True
Size	105, 20 (o por default)

Como adicionar una barra de estado al formulario:

Ir a la ventana **Toolbox** y seleccionar: **Containers / Panel**

Seleccionar la propiedad **Dock** y seleccionar **Bottom**

Incluir una imagen dentro del Panel seleccionando en **Toolbox: Common Controls / PictureBox**

Seleccionar la imagen, en nuestro ejemplo: **desarrolla con msdn**, cambiar la propiedad:

Behavior / SizeMode a **AutoSize**

Vamos a colocar un fondo al menú para eso hacer clic en el objeto menú y seleccionar la propiedad: **BackgroundImage**, seleccionar una opción y listo.

Grabar y ejecutar el proyecto, presionado F5.

¿Qué pasa si agrandamos la ventana?, ¿que se observa?

La imagen de la barra de estado, parte inferior de nuestro formulario se observa descentrado, ¿cómo mejorar este problema?, muy sencillo regresamos a modo desarrollo, seleccionar el objeto y cambiar la propiedad: **Layout / Anchor** a **Bottom, Right**.

Ahora, desarrollar los formularios hijos que se van a mostrar en el formulario padre, para lograr esto vamos primero a construir una plantilla para que estos formulario sean de idéntico diseño.

Crear un nuevo formulario al que le ponemos por nombre: PlantillaForm.cs

Cambiar las siguientes propiedades:

- BackColor: GradientInactiveCaption por ejemplo
- ControlBox: False

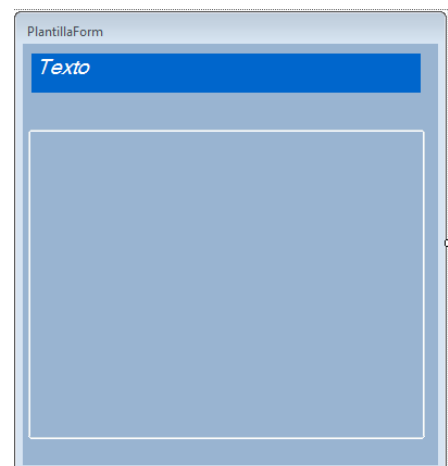
Adicionar un objeto **Label** al formulario, cambiar las propiedades:

- AutoSize: False
- BackColor: HotTrack
- ForeColor: ButtonHighlight
- Font / Size: 14
- Font / Italic: True
- Text: Texto
- Layout / Anchor: Top, Left, Right
- Design / Modifiers: Protected

Ir a Toolbox, seleccionar: **Containers / GroupBox**, adicionar al formulario, cambiar las propiedades:

- Appearance / Texto: Dejar en blanco, eliminar el texto que existía por default
- Layout / Anchor: Top, Bottom, Left, Right

Hemos terminado con el formulario plantilla.



Ahora vamos a hacer que los formularios de Socios, Películas y Alquileres tengan esta misma apariencia, ¿Cómo logramos esto?

Crear nuevos formularios

Adicionar nuevos formularios y poner como nombres:

- AlquilarForm.cs
- DevolverForm.cs
- NuevoSocioForm.cs

Abrir en modo programación, esto se logra dando doble clic sobre el marco en el que se muestra el nombre del formulario, nos muestra las siguientes líneas de programación:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MsdnVideo
{
    public partial class NuevoSocioForm : Form
    {
        public NuevoSocioForm()
        {
            InitializeComponent();
        }
    }
}
```

En la línea de programa que se indica, cambiar **Form** por **PlantillaForm**

```
public partial class NuevoSocioForm : PlantillaForm
```

Hacer lo mismo para los dos formularios siguientes, hacer los ajustes de tamaño de los objetos y grabar.

Modificar la caja de texto para cada formulario, colocando

Label1 / Text: Nuevo Socio / Alquilar Película / Devolver Película

Ahora regresamos al primer formulario: Form1

Para efectuar la conexión o enlace con los otros formularios, agregar en modo programación las siguientes líneas de comando.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MsdnVideo
```

```

{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void nuevoSocioToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            NuevoSocioForm formNSocio = new NuevoSocioForm();
            formNSocio.MdiParent = this;
            formNSocio.Show();
        }

        private void devolverToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DevolverForm formDevolver = new DevolverForm();
            formDevolver.MdiParent = this;
            formDevolver.Show();
        }

        private void alquilarToolStripMenuItem_Click(object sender, EventArgs e)
        {
            AlquilarForm formAlquilar = new AlquilarForm();
            formAlquilar.MdiParent = this;
            formAlquilar.Show();
        }

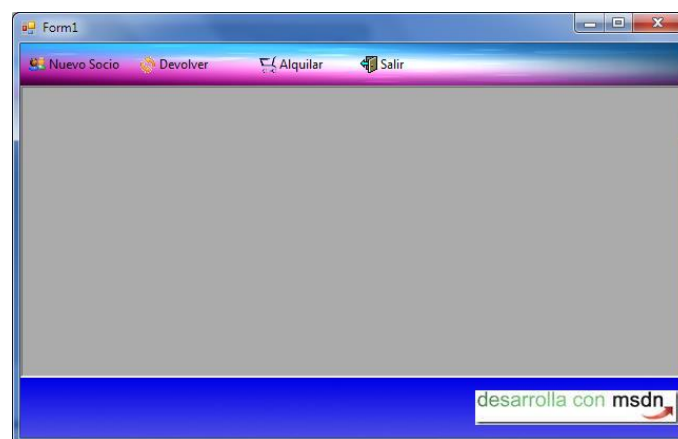
        private void salirToolStripMenuItem_Click(object sender, EventArgs e)
        {
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}

```

Grabar y hemos terminado esta segunda parte del laboratorio.

Para probar, presionar F5 y obtenemos el siguiente resultado:



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA INDUSTRIAL

Curso: Ingeniería de Software

Profesor: Ing. Wiler Arturo Ponce Benites

Correo: wiler.ponce@industrial.unmsm.pe

Página Web: <http://campusvirtualfii.unmsm.edu.pe>

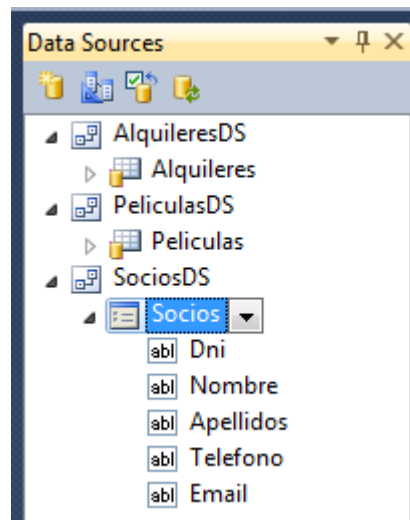
Laboratorio: Diseño de un Proyecto de Software (Parte 3)

En este laboratorio, realizaremos el desarrollo del formulario: **Nuevo Socio**

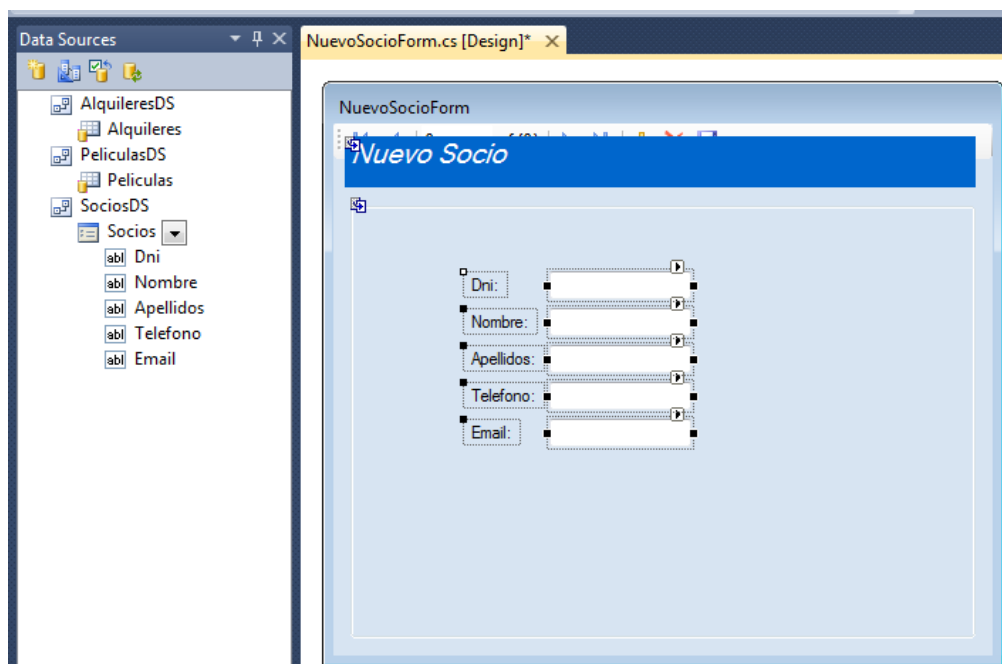
Recuperar el formulario NuevoSocioForm.cs

Este formulario tiene por objetivo efectuar el ingreso, modificación de datos de la tabla socios,

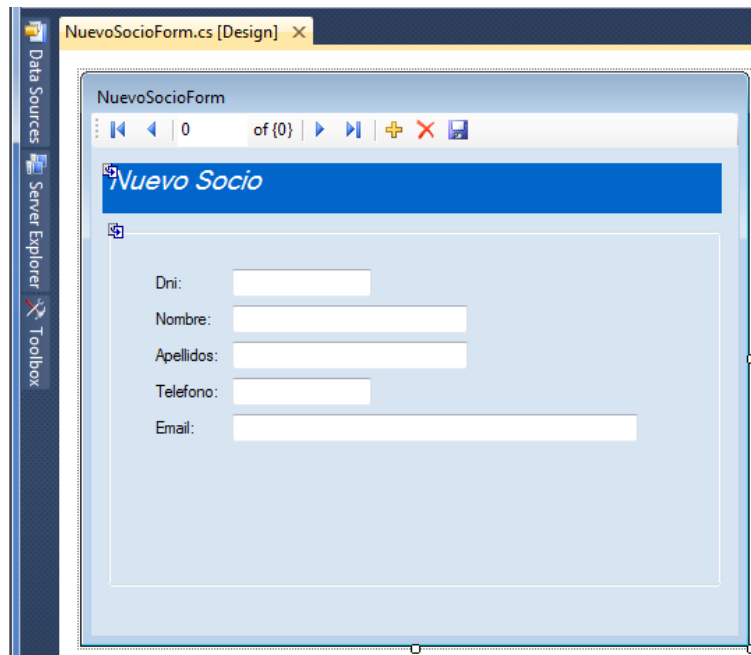
Al invocar la pestaña: Data Sources, y elegir el icono SociosDS / Socios, se observa:



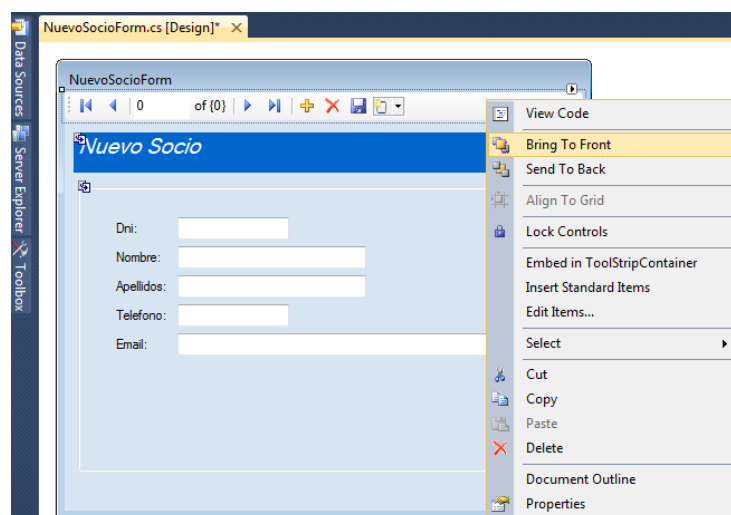
Hacemos una operación de arrastre al formulario y se obtiene:



Se puede modificar el tamaño de los TextBox a voluntad, Mediante una operación de arrastre ubicando el curso en el punto negro, luego reubicando los objetos dentro de formulario, se tiene el siguiente:



Ubicar el cursor sobre el objeto **sociosBindingNavigator** y hacer clic sobre la opción **Bring To Front**.



Hasta aquí ya nuestro formulario está operativo hacer pruebas con datos. Todo lo realizado hasta el momento ha agregado a nuestro formulario las siguientes líneas de comando:

Esto solo mantiene los cambios en memoria, aún no han sido almacenados en la Base de Datos o que simplemente está navegando por el Dataset.

Para que los cambios se guarden en la base de datos hay que efectuar los siguientes cambios en la programación:

Hacer cambios en la programación de tal manera que se muestre de la siguiente manera

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MsdnVideo
{
    public partial class NuevoSocioForm : PlantillaForm
    {
        public NuevoSocioForm()
        {
            InitializeComponent();
        }

        private void NuevoSocioForm_Load(object sender, EventArgs e)
        {
            this.sociosBindingSource.AddNew();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Close();
        }

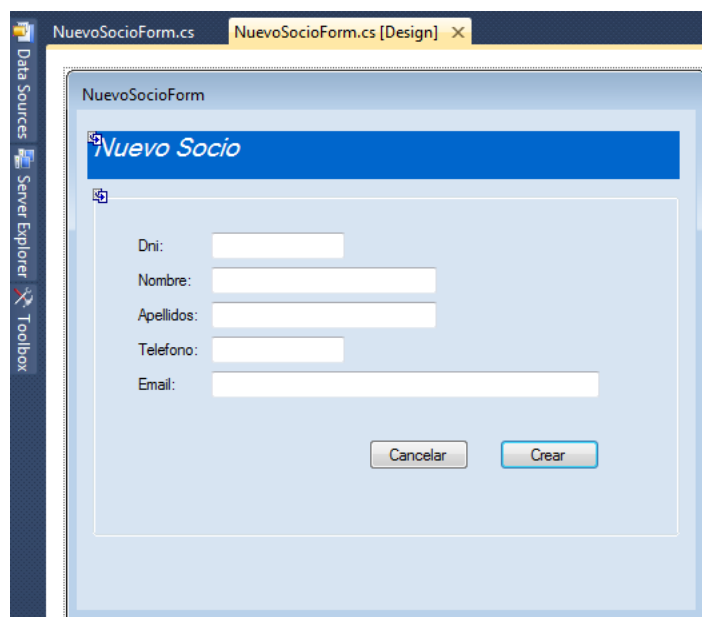
        private void button1_Click(object sender, EventArgs e)
        {
            if (this.Validate())
            {
                this.sociosBindingSource.EndEdit();
                this.sociosTableAdapter.Update(this.sociosDS.Socios);
            }
            else
            {
                System.Windows.Forms.MessageBox.Show(this, "Validation error occurred.", "Save", System.Windows.Forms.MessageBoxButtons.OK, System.Windows.Forms.MessageBoxIcon.Exclamation);
            }
        }
    }
}
```

Finalmente este sería nuestro formulario deseado.

A continuación veremos otras consideraciones importantes:

1.- El campo DNI, requiere de una máscara de entrada, esto se logra de la siguiente manera.

Validación del DNI, para ello vamos al icono de eventos de la ventana de exploración y damos clic, nos muestra una lista de evento, elegimos Validating



Estos eventos requieren mayor investigación

Programa finalizado:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MsdnVideo
{
    public partial class NuevoSocioForm : PlantillaForm
    {
        public NuevoSocioForm()
        {
            InitializeComponent();
        }

        private void NuevoSocioForm_Load(object sender, EventArgs e)
        {
            this.sociosBindingSource.AddNew();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (this.Validate())
            {
                this.sociosBindingSource.EndEdit();
                this.sociosTableAdapter.Update(this.sociosDS.Socios);
                MessageBox.Show("Socio insertado correctamente");
                this.sociosBindingSource.AddNew();
            }
            else
            {
                System.Windows.Forms.MessageBox.Show(this, "Validation error occurred.", "Save", System.Windows.Forms.MessageBoxButtons.OK, System.Windows.Forms.MessageBoxIcon.Exclamation);
            }
        }
    }
}
```



```

        private void dniTextBox_Validating(object sender, CancelEventArgs e)
        {
            /* Esta validación del DNI verifica que la Letra de chequeo corresponda al
            conjunto de numeros ingresados; para nuestro caso para poder pasar y no nos
            detenga así haya error en la letra de chequeo, en modo diseño, seleccionar el
            formulario y ubicar la propiedad Autovalidate, seleccionar la opción:
            EnableAllowFocusChange.
            */

            string Letras = "TRWAGMYFPDXBNJZSQVHLCKE";
            int numNif;
            string nif = dniTextBox.Text;
            bool Valido = true;

            if (!System.Text.RegularExpressions.Regex.IsMatch(nif, @"^\d{8}[" +
Letras + "]+$"))
                Valido = false;
            else
            {
                numNif = Int32.Parse(nif.Substring(0, nif.Length - 1));
                if (!(nif[8] == Letras[numNif % 23]))
                    Valido = false;
            }

            if (!Valido)
            {
                e.Cancel = true;
                errorProvider1.SetError(dniTextBox, "La letra del DNI no es
correcta");
            }
            else
            {
                e.Cancel = false;
                errorProvider1.SetError(dniTextBox, string.Empty);
            }
        }

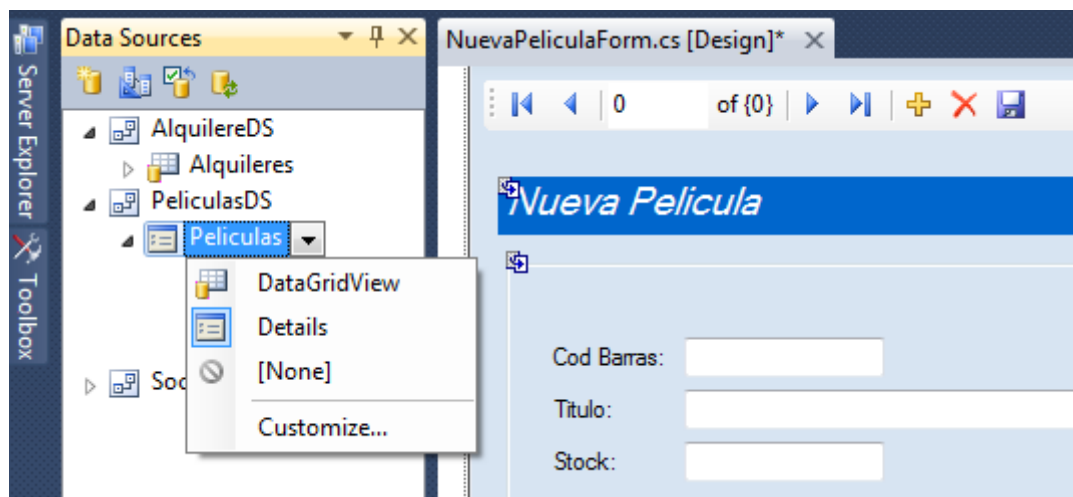
        private void nombreTextBox_Validating(object sender, CancelEventArgs e)
        {
            if (nombreTextBox.Text.Length == 0)
            {
                e.Cancel = true;
                errorProvider1.SetError(nombreTextBox, "introduzca un nombre");
            }
            else
            {
                e.Cancel = false;
                errorProvider1.SetError(nombreTextBox, string.Empty);
            }
        }
    }

```

```
e)
private void apellidosTextBox_Validating(object sender, CancelEventArgs
{
    if (apellidosTextBox.Text.Length == 0)
    {
        e.Cancel = true;
        errorProvider1.SetError(apellidosTextBox, "introduzca los
apellidos");
    }
    else
    {
        e.Cancel = false;
        errorProvider1.SetError(apellidosTextBox, string.Empty);
    }
}
}
```

Formulario: NuevaPeliculaForm

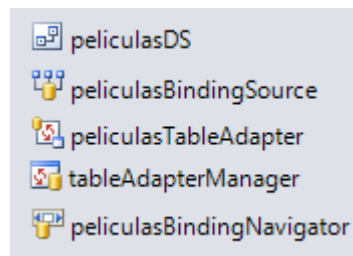
1.- En la ventana Data Source, seleccionar el DataSet Película, hacer clic en Películas y desplegar el combobox, seleccionar: **Details**



2.- Con el cursor seleccionar Películas y arrastrar hacia el formulario, soltar el botón del mouse.

3.- Esta acción ha adicionado a nuestro formulario los objetos:

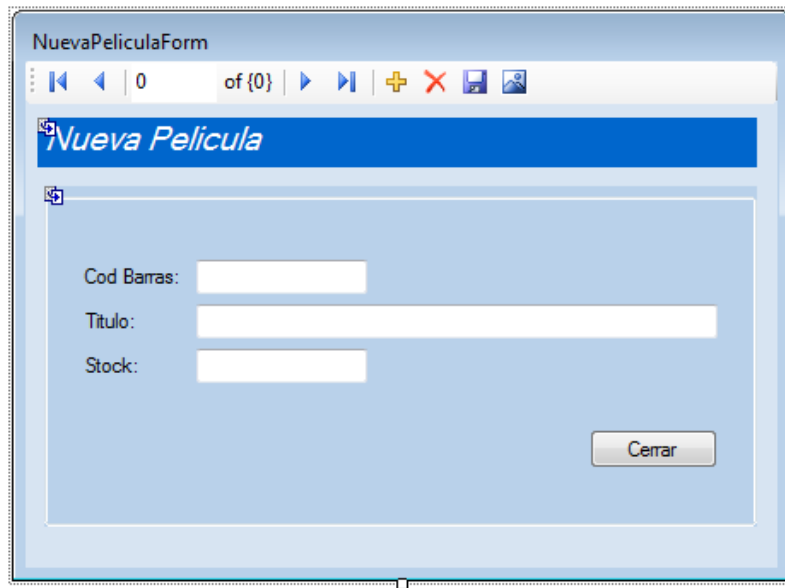
- películasDS
- películasBindingSource
- películasTableAdapter
- tableAdapterManager
- películasBindingNavigator



4.- Adicionar un botón de comando modificar las siguientes propiedades:

- | | |
|----------|-----------|
| ■ (Name) | btnCerrar |
| ■ Text | Cerrar |

5.- Este es nuestro formulario, finalmente diseñado:



5.-Finalmente el programa deberá quedar de la siguiente así:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace GestionVideos
{
    public partial class NuevaPeliculaForm : PlantillaForm
    {
        public NuevaPeliculaForm()
        {
            InitializeComponent();
        }

        private void NuevaPeliculaForm_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
            'peliculasDS.Peliculas' table. You can move, or remove it, as needed.
            this.peliculasTableAdapter.Fill(this.peliculasDS.Peliculas);
        }

        private void peliculasBindingNavigatorSaveItem_Click_1(object sender,
            EventArgs e)
        {
            this.Validate();
            this.peliculasBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.peliculasDS);
            System.Windows.Forms.MessageBox.Show(this, "Grabación exitosa.",
            "Grabar", System.Windows.Forms.MessageBoxButtons.OK,
            System.Windows.Forms.MessageBoxIcon.Information);
            this.peliculasBindingSource.AddNew();
        }
    }
}
```

```
private void btnCerrar_Click(object sender, EventArgs e)
{
    Close();
}

private void bindingNavigatorAddNewItem_Click(object sender, EventArgs e)
{
}
}
```

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA INDUSTRIAL**

Curso: Ingeniería de Software

Profesor: Ing. Wiler Arturo Ponce Benites

Correo: wiler.ponce@industrial.unmsm.pe

Página Web: <http://campusvirtualfii.unmsm.edu.pe>

Laboratorio: Diseño de un Proyecto de Software (Parte 4)

Formulario: Alquiler de Películas

En este laboratorio, realizaremos el desarrollo del formulario: **Alquilar Película**

Recuperar el formulario **AlquilarForm.cs**

Agregar los siguientes objetos y actualizar las propiedades que se indican a continuación:

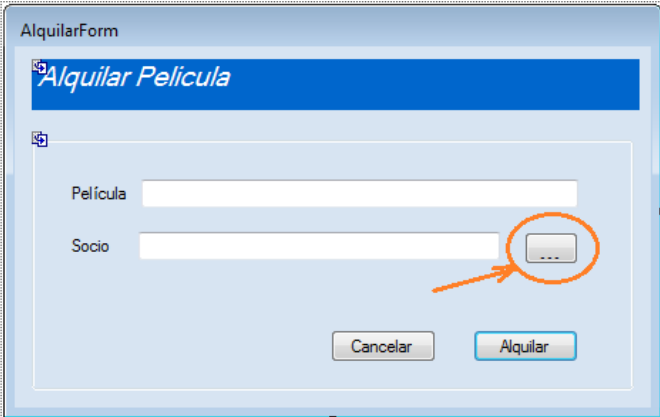
Label2	Text Anchor	Película TL (<u>T</u> op, <u>L</u> eft)
Label3	Text Anchor	SocioDNI TL (<u>T</u> op, <u>L</u> eft)
Textbox1	Text Anchor	en blanco TLR (<u>T</u> op, <u>L</u> eft, <u>R</u> ight)
Textbox2	Text Anchor	en blanco TR (<u>T</u> op, <u>R</u> ight)
Button1	Text Anchor	... TR (<u>T</u> op, <u>R</u> ight)
Button2	Text Anchor	Alquilar TR (<u>T</u> op, <u>R</u> ight)
Button3	Text Anchor	Cancelar TR (<u>T</u> op, <u>R</u> ight)

En el formulario adicionar las siguientes propiedades:

AcceptButton	button2
CancelButton	button3

Con el botón de búsqueda, llamar al formulario **búsqueda de socios**, al que daremos por nombre:

SeleccionSociosForm.cs



Formulario para la Búsqueda de Socios

Adicionar un nuevo formulario, darle el nombre de **SeleccionSociosForm.cs**

Dar doble clic en el formulario y efectuar el siguiente cambio:

Dice: `public partial class AlquilarForm : Form`
Debe decir: `public partial class AlquilarForm : PlantillaForm`

Este cambio se hace para asociar el formato Plantilla y sea el diseño igual para todos los formularios, tal como lo indicamos en la Guía de Laboratorio (Parte 2)

Agregar los objetos:

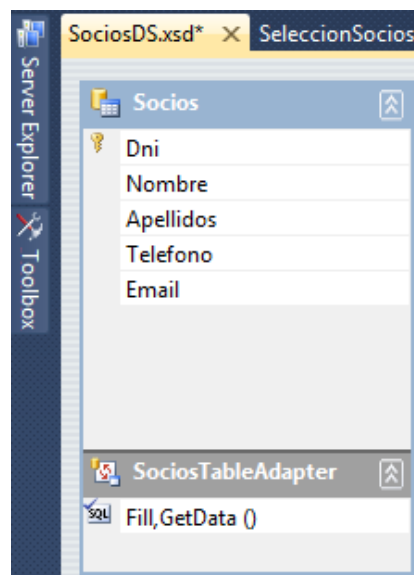
Label2	Text Anchor	Nombre TL
TextBox1	Text Anchor	en blanco
Button1	Text Anchor	Buscar TR (Top, Right)
DataGridView	Anchor	TBLR (Top, Bottom, Left, Right)

Al presionar el botón de búsqueda del formulario Alquilar Películas debe mostrarse el formulario Búsqueda de Socios en donde al ingresar, cualquiera de los datos: nombre o apellido debe mostrar una lista de socios que cumplan con la llave de búsqueda.

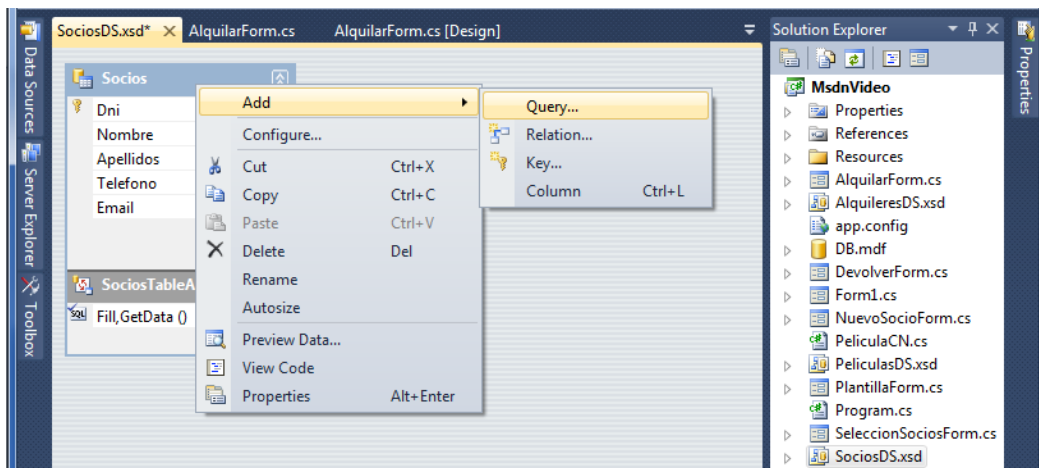
Para lograr esto realizar el siguiente procedimiento

1.- Construir un filtro en la tabla Socios, para lograr esto hacemos lo siguiente

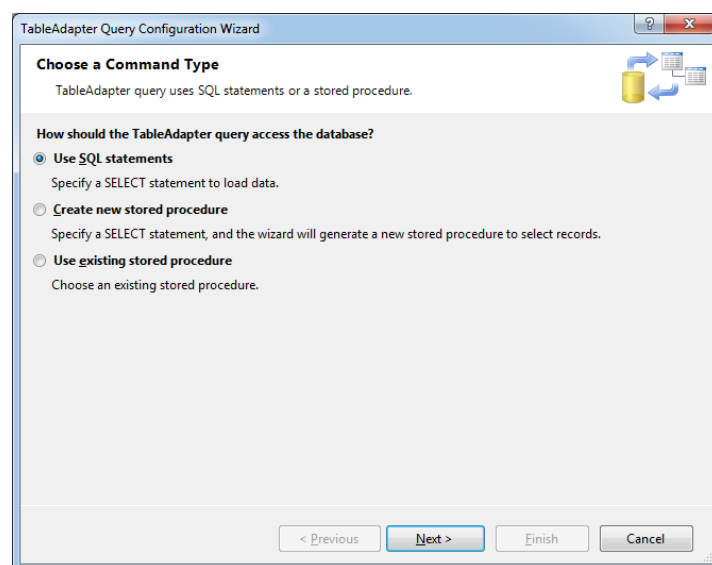
Ir a la ventana **Solution Explorer**, seleccionar **SociosDS.xsd** y dar doble clic para abrirlo, nos muestra la siguiente ventana, se muestra un solo método que sirve para rellenar el dataset con todos los registros de la tabla y que es Fill,GetData ()



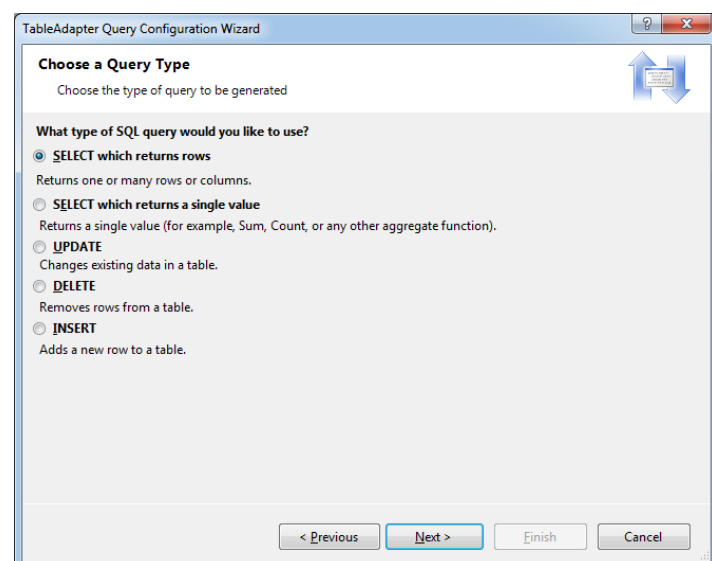
Hacer clic derecho en el título del objeto: **Socios** y seleccionar **Add / Query**



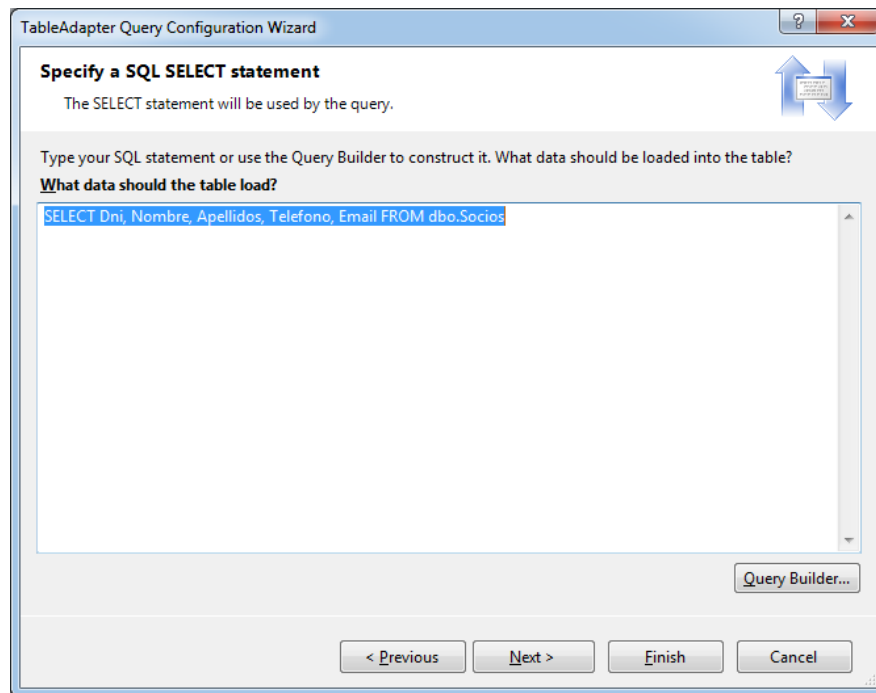
y nos muestra la siguiente ventana:



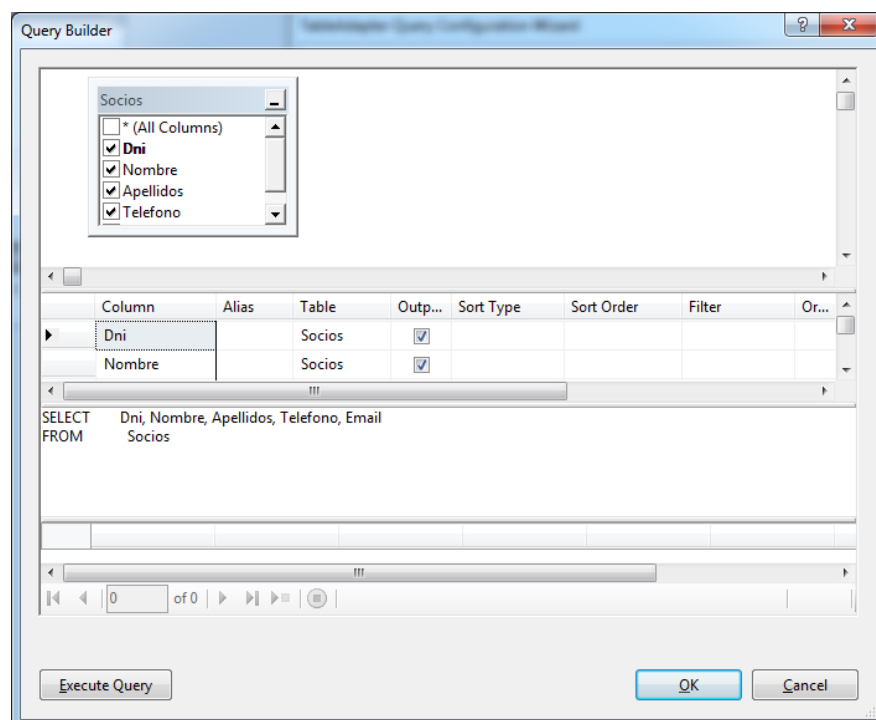
Seleccionamos la opción Use SQL statements y hacemos clic en **Next** y obtenemos la siguiente ventana:



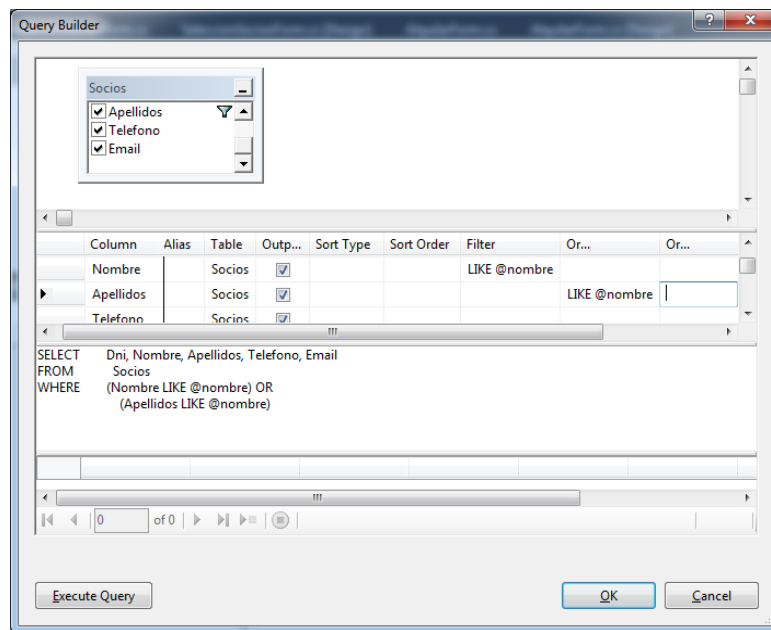
Hacemos clic en **Next** y obtenemos:



Abrir **Generador de Consultas**, esto se logra, haciendo clic en el botón: **Query Builder** y obtenemos la siguiente ventana:



Se puede ajustar el tamaño de las ventanas a voluntad, así procedemos y obtenemos la siguiente ventana, en la que digitamos en la columna **Filter** para el campo **Nombre**, **LIKE @nombre** y para el campo **Apellidos**, **LIKE @nombre** en la columna **Or** y tenemos lo siguiente:

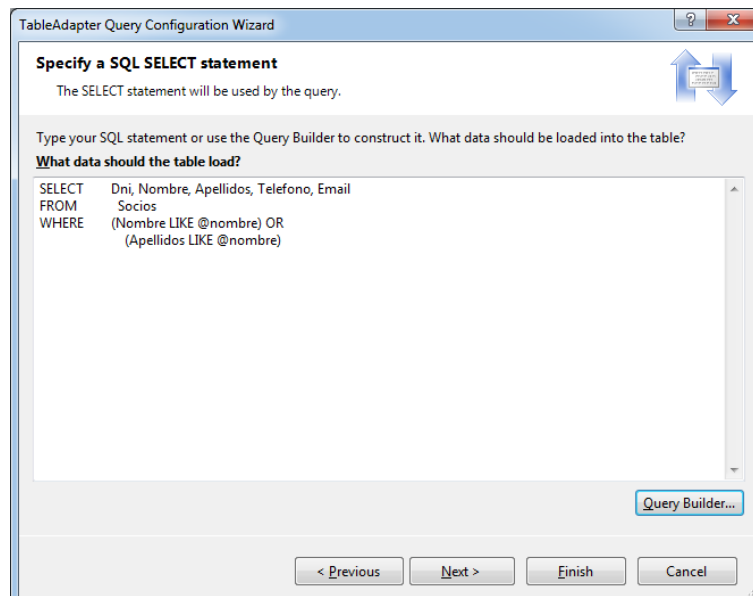


Lo que se ha logrado es la instrucción SQL siguiente:

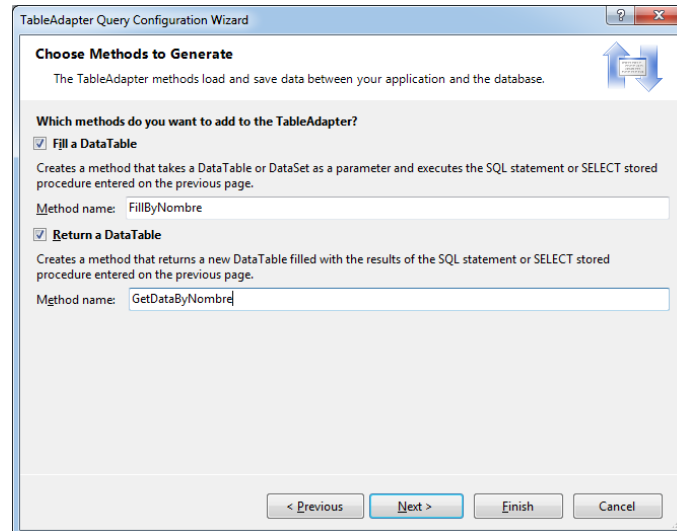
```

SELECT      Dni, Nombre, Apellidos, Telefono, Email
FROM        Socios
WHERE       (Nombre LIKE @nombre) OR (Apellidos LIKE @nombre)
  
```

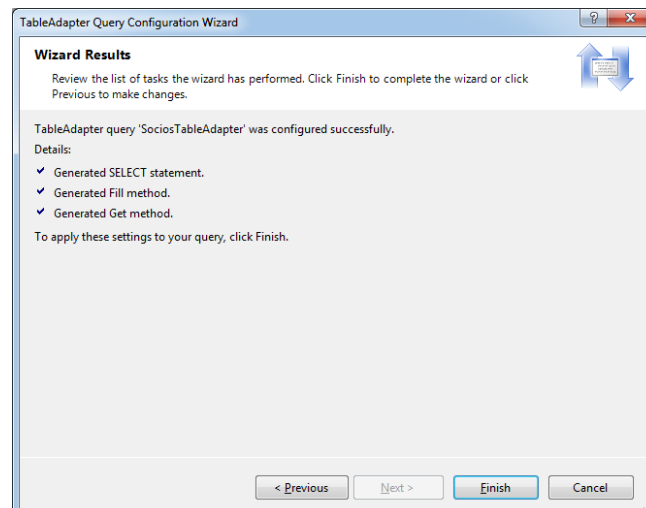
Hacemos clic en el botón **OK** y obtenemos la siguiente pantalla:



Hacer clic en el botón **Next** y llegamos a la siguiente ventana, en donde en el textBox Method name, se completa con FillByNombre y hace lo mismo con el segundo textBox Method name, completando con GetDataByNombre,



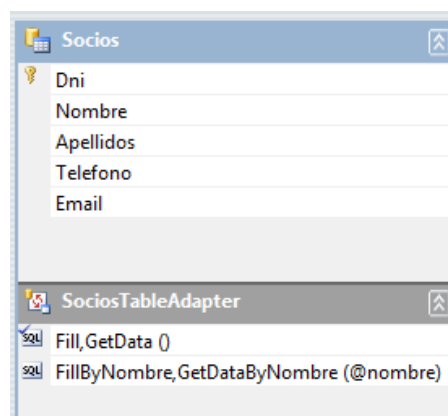
hacemos clic en el botón **Next** y obtenemos la siguiente ventana:



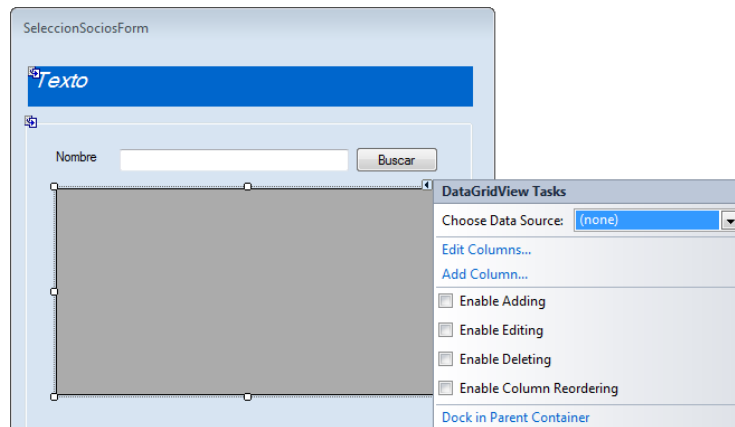
Hacemos clic en el botón **Finish** y observamos que en el DataSet se ha adicionado un nuevo metodo que nos permite efectuar el filtro deseado:

FillByNombre,GetDataByNombre(@nombre)

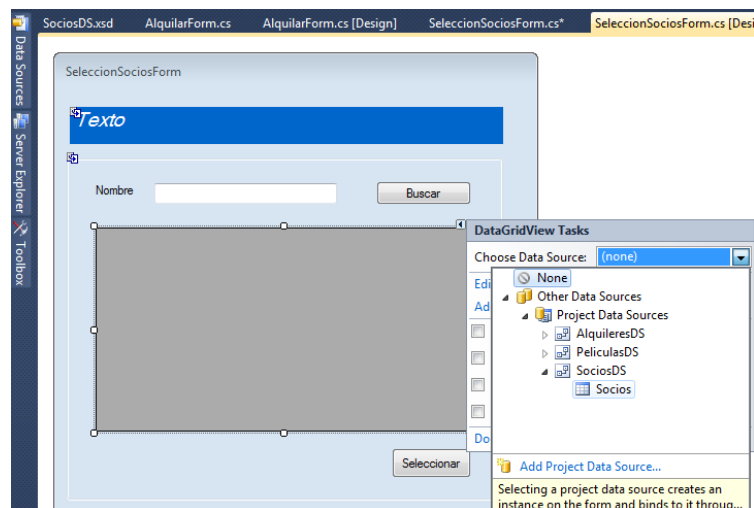
,



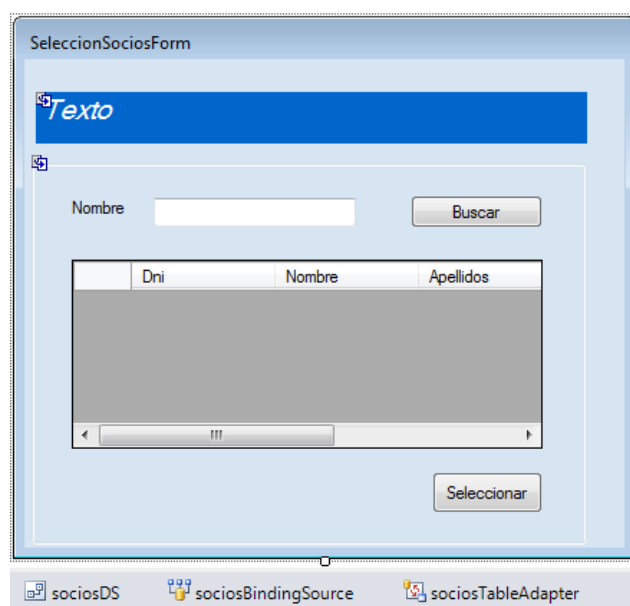
Retornamos al formulario, ahora trabajamos con el DataGridView, hacemos clic en la punta de flecha para desplegar la ventana de opciones de DGV y desactivamos todos los CheckBox,



Ubicando el cursor en el ComboBox de esta caja desplegamos el explorador hasta encontrar el objeto socios, lo seleccionamos y hacemos clic.



Como resultado tenemos el DGV debidamente configurado.



Probar el programa en esta parte hay errores

Codigo de: SeleccionSociosForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MsdnVideo
{
    public partial class SeleccionSociosForm : PlantillaForm
    {
        public string DNI = string.Empty;
        public SeleccionSociosForm()
        {
            InitializeComponent();

            private void SeleccionSociosForm_Load(object sender, EventArgs e)
            {
                // TODO: This line of code loads data into the 'sociosDS.Socios'
                table. You can move, or remove it, as needed.
                // this.sociosTableAdapter.Fill(this.sociosDS.Socios);

            }

            private void button1_Click(object sender, EventArgs e)
            {
                SociosDSTableAdapters.SociosTableAdapter adaptadorSocios= new
                MsdnVideo.SociosDSTableAdapters.SociosTableAdapter();
                // SociosDS sociosDS = new SociosDS();
                adaptadorSocios.FillByNombre(sociosDS.Socios, string.Format("%(0)%",
                textBox1.Text));
                // adaptadorSocios.FillByNombre(sociosDS.Socios,
                string.Format("%(0)%", textBox1.Text));

                MessageBox.Show("Hemos llegado hasta aqui");
            }

            private void button2_Click(object sender, EventArgs e)
            {
                DNI = sociosDS.Socios[sociosBindingSource.Position].Dni;
                Close();
            }
        }
    }
}
```

Codigo del AlquilerForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MsdnVideo
{
    public partial class AlquilerForm : PlantillaForm
    {
        public AlquilerForm()
        {
            InitializeComponent();
        }

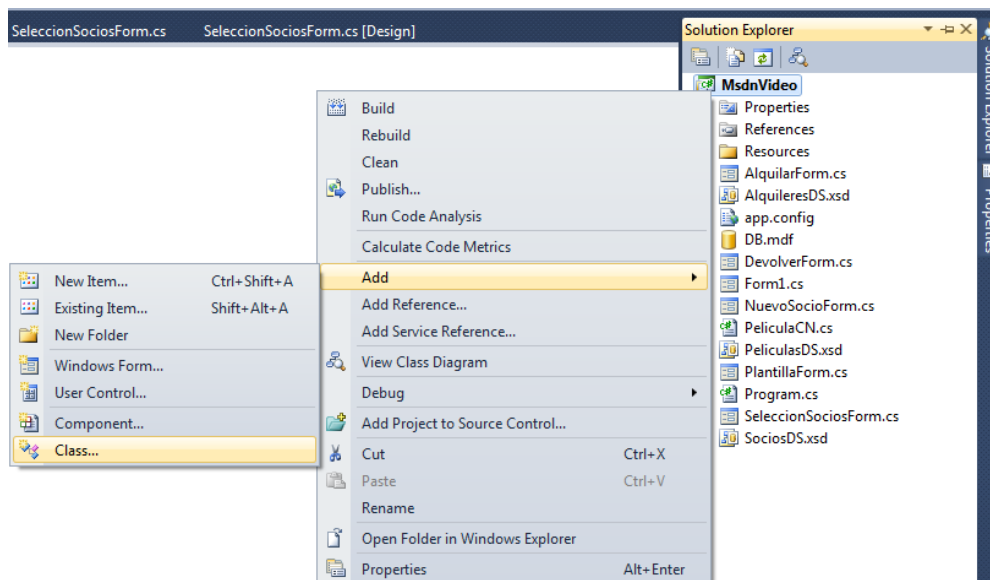
        private void AlquilerForm_Load(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SeleccionSociosForm formSeleccionSocios = new SeleccionSociosForm();
            formSeleccionSocios.ShowDialog();
            textBox2.Text = formSeleccionSocios.DNI;
        }

        private void button2_Click(object sender, EventArgs e)
        {
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}
```

Crear una nueva clase:



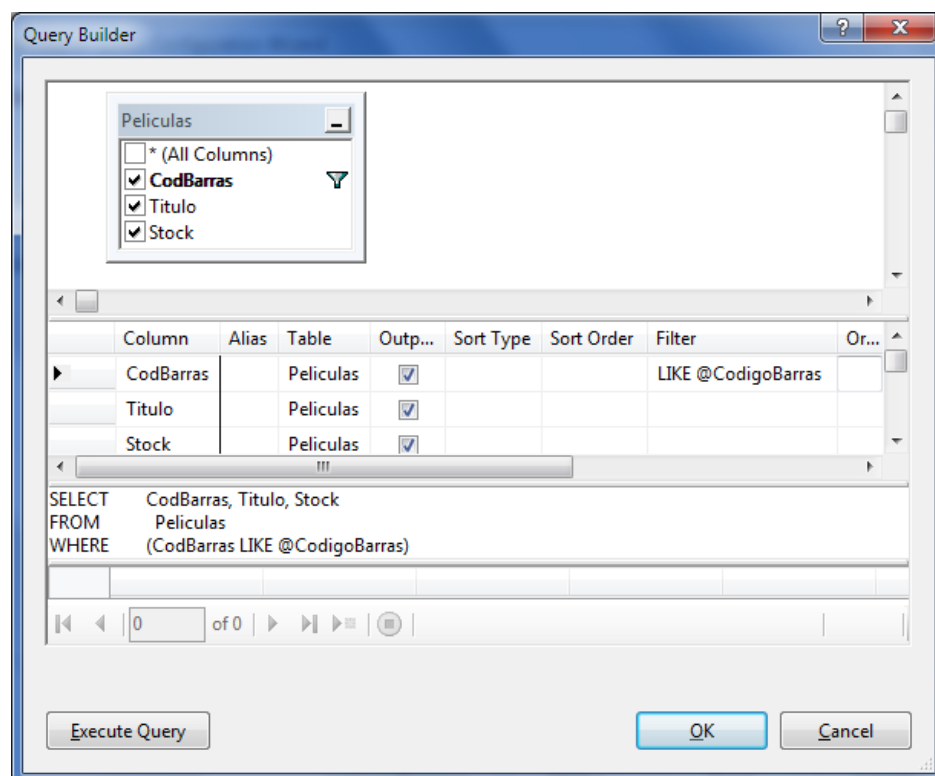
Darle por nombre: **PeliculaCN.cs**, hacer clic en el boton **Add**.

Luego generamos en el DataSet de Peliculas un metodo de selecci3n similar al realizado para busqueda de socios:

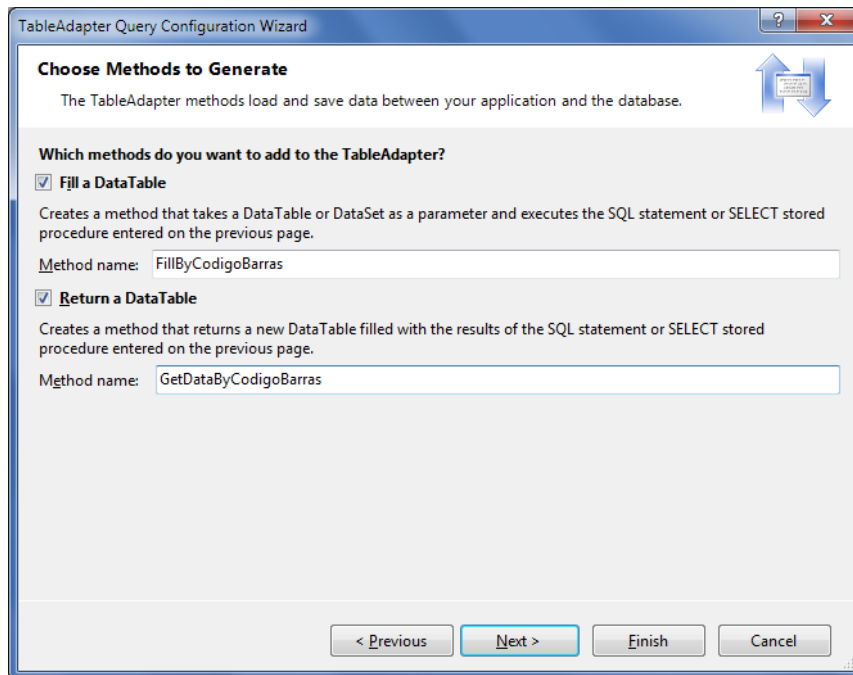
Abrir el DataSet Peliculas

Clic derecho en la barra de titulo del dataset, se despliega una ventana de menu, seleccionar **Add / Query**

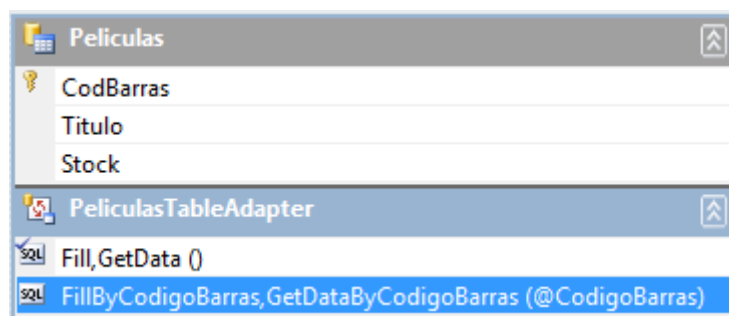
Y generamos el siguiente query:



Click en botón **OK** luego en botón **Next**



Clic en botón **Next** y luego en la siguiente ventana en botón **Finish**, regresamos a la ventana del dataset y observamos que se ha adicionado el nuevo método de filtrado de datos, en este caso para la tabla películas.



Hasta aquí debemos de contar con el siguiente código

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace MsdnVideo
{
    class PeliculasCN
    {
        public void AlquilarPelicula(string CodBarrasPelicula, string DniSocio)
        {
            // Comprobar el stock
            PeliculasDSTableAdapters.PeliculasTableAdapter adaptadorPeliculas =
            new MsdnVideo.PeliculasDSTableAdapters.PeliculasTableAdapter();
            PeliculasDS peliculasDS = new PeliculasDS();
            adaptadorPeliculas.FillByCodigoBarras(peliculasDS.Peliculas,
            CodBarrasPelicula);

            if (peliculasDS.Peliculas.Count == 0)
            {
                throw new ApplicationException("No existe la pelicula");
            }
        }
    }
}
```

```

    }

    if (peliculasDS.Peliculas[0].Stock == 0)
    {
        throw new ApplicationException("No hay stock de peliculas");
    }

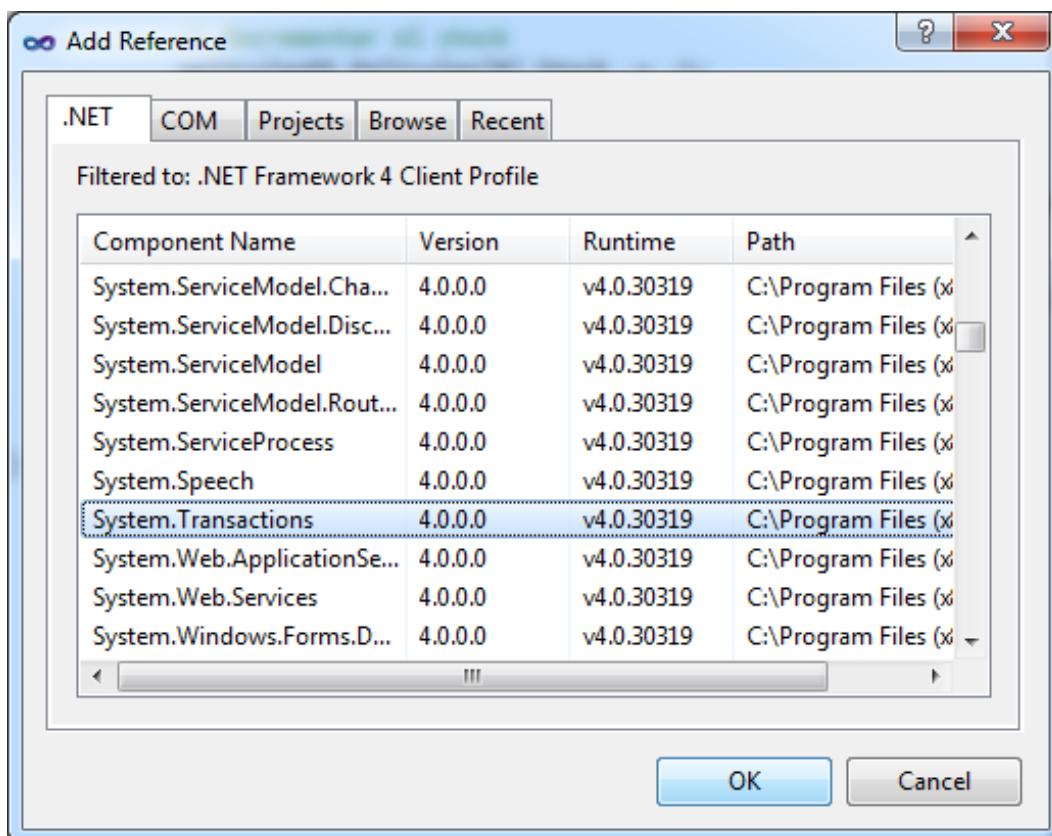
    // Decrementar el stock
    peliculasDS.Peliculas[0].Stock -= 1;
    adaptadorPeliculas.Update(peliculasDS.Peliculas);

    // Alquilar la peliculas
    AlquileresDSTableAdapters.AlquileresTableAdapter adaptadorAlquileres
= new MsdnVideo.AlquileresDSTableAdapters.AlquileresTableAdapter();
    AlquileresDS alquilerDS = new AlquileresDS();

    alquilerDS.Alquileres.AddAlquileresRow(DniSocio, CodBarrasPelicula,
DateTime.Now, false);
    adaptadorAlquileres.Update(alquilerDS.Alquileres);
}
}
}

```

Ahora adicionamos una **Referencia**, para lograrlo, en la ventana **Solution Explorer** sobre el icono **References** hacemos clic derecho y desplegamos la ventana **Add Reference** en la pestaña **.Net** localizamos el componente **System.Transactions** la seleccionamos y damos clic en **OK**



Agregamos nuevas líneas de código y obtenemos el siguiente programa:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using System.Transactions;

namespace MsdnVideo
{
    class PeliculasCN
    {
        public void AlquilarPelicula(string CodBarrasPelicula, string DniSocio)
        {
            TransactionScope tx = new TransactionScope();
            using(tx)
            {
                // Comprobar el stock
                PeliculasDSTableAdapters.PeliculasTableAdapter adaptadorPeliculas
= new MsdnVideo.PeliculasDSTableAdapters.PeliculasTableAdapter();
                PeliculasDS peliculasDS = new PeliculasDS();
                adaptadorPeliculas.FillByCodigoBarras(peliculasDS.Peliculas,
CodBarrasPelicula);

                if (peliculasDS.Peliculas.Count == 0)
                {
                    throw new ApplicationException("No existe la pelicula");
                }

                if (peliculasDS.Peliculas[0].Stock == 0)
                {
                    throw new ApplicationException("No hay stock de peliculas");
                }

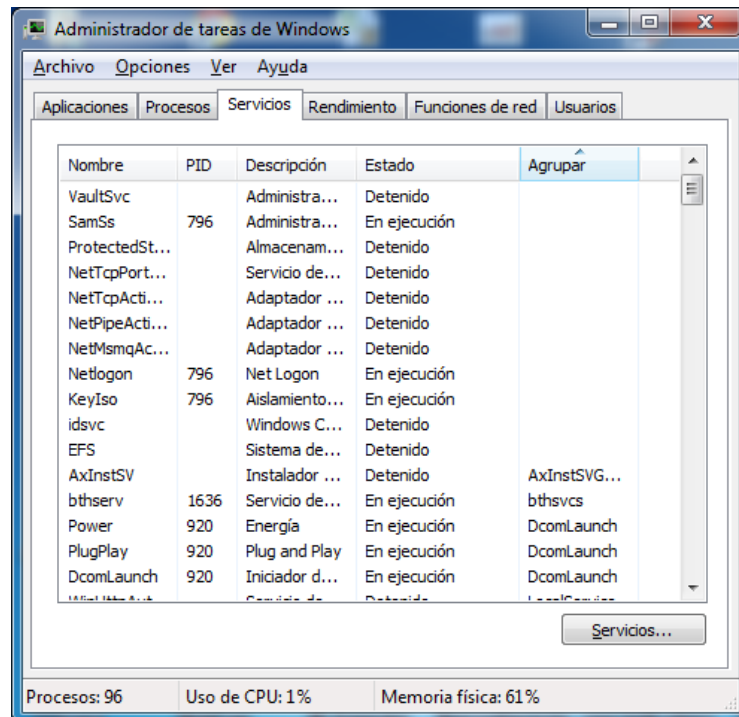
                // Decrementar el stock
                peliculasDS.Peliculas[0].Stock -= 1;
                adaptadorPeliculas.Update(peliculasDS.Peliculas);

                // Alquilar la peliculas
                AlquileresDSTableAdapters.AlquileresTableAdapter
adaptadorAlquileres = new
MsdnVideo.AlquileresDSTableAdapters.AlquileresTableAdapter();
                AlquileresDS alquilerDS = new AlquileresDS();

                alquilerDS.Alquileres.AddAlquileresRow(DniSocio,
CodBarrasPelicula, DateTime.Now, false);
                adaptadorAlquileres.Update(alquilerDS.Alquileres);

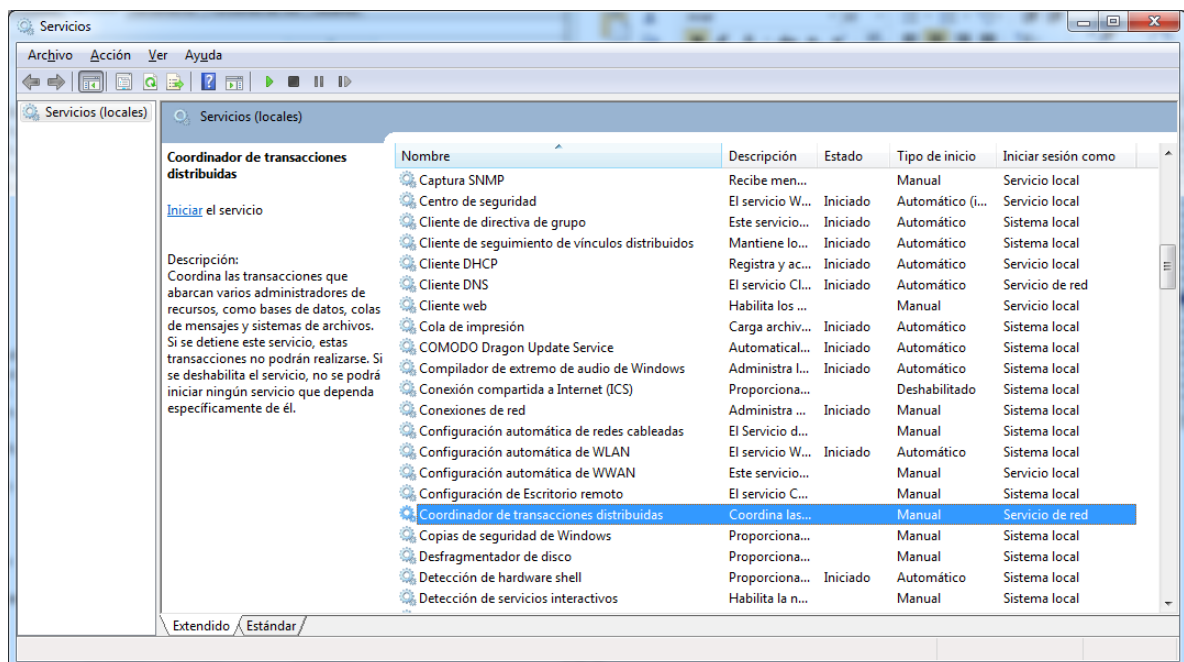
                // Completamos la transaccion si todo es correcto
                tx.Complete();
            }
        }
    }
}
```

Abrir el Administrador de tareas de Windows, presionando las teclas: **Alt + Ctrl + Supr**, seleccionar la pestaña **Servicios** y hacemos clic en el botón **Servicios**.



Obtenemos esta ventana en donde elegimos la opción:

Coordinador de transacciones distribuidas de Microsoft



Hacer clic en [Iniciar el servicio](#)

Finalmente, el código para nuestros formularios,es:

Formulario AlquilerForm.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using System.Transactions;

namespace MsdnVideo
{
    class PeliculasCN
    {
        public void AlquilarPelicula(string CodBarrasPelicula, string DniSocio)
        {
            TransactionScope tx = new TransactionScope();
            using(tx)
            {
                // Comprobar el stock
                PeliculasDSTableAdapters.PeliculasTableAdapter adaptadorPeliculas
= new MsdnVideo.PeliculasDSTableAdapters.PeliculasTableAdapter();
                PeliculasDS peliculasDS = new PeliculasDS();
                adaptadorPeliculas.FillByCodigoBarras(peliculasDS.Peliculas,
CodBarrasPelicula);

                if (peliculasDS.Peliculas.Count == 0)
                {
                    throw new ApplicationException("No existe la pelicula");
                }

                if (peliculasDS.Peliculas[0].Stock == 0)
                {
                    throw new ApplicationException("No hay stock de peliculas");
                }

                // Decrementar el stock
                peliculasDS.Peliculas[0].Stock -= 1;
                adaptadorPeliculas.Update(peliculasDS.Peliculas);

                // Alquilar la peliculas
                AlquileresDSTableAdapters.AlquileresTableAdapter
adaptadorAlquileres = new
MsdnVideo.AlquileresDSTableAdapters.AlquileresTableAdapter();
                AlquileresDS alquilerDS = new AlquileresDS();

                alquilerDS.Alquileres.AddAlquileresRow(DniSocio,
CodBarrasPelicula, DateTime.Now, false);
                adaptadorAlquileres.Update(alquilerDS.Alquileres);

                // Completamos la transaccion si todo es correcto
                tx.Complete();
            }
        }
    }
}
```

Formulario SeleccionSociosForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MsdnVideo
{
    public partial class SeleccionSociosForm : PlantillaForm
    {
        public string DNI = string.Empty;
        public SeleccionSociosForm()
        {
            InitializeComponent();

            private void SeleccionSociosForm_Load(object sender, EventArgs e)
            {
                // TODO: This line of code loads data into the 'sociosDS.Socios'
                table. You can move, or remove it, as needed.
                // this.sociosTableAdapter.Fill(this.sociosDS.Socios);
            }

            private void button1_Click(object sender, EventArgs e)
            {
                SociosDSTableAdapters.SociosTableAdapter adaptadorSocios= new
                MsdnVideo.SociosDSTableAdapters.SociosTableAdapter();
                // SociosDS sociosDS = new SociosDS();
                // adaptadorSocios.FillByNombre(sociosDS.Socios,
                string.Format("%(0)%", textBox1.Text));
                adaptadorSocios.FillByNombre(sociosDS.Socios, string.Format("%(0)%",
                textBox1.Text));
                MessageBox.Show("Hemos llegado hasta aqui");
            }

            private void button2_Click(object sender, EventArgs e)
            {
                DNI = sociosDS.Socios[sociosBindingSource.Position].Dni;
                Close();
            }
        }
    }
}
```

FormularioClase PeliculasCN.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using System.Transactions;

namespace MsdnVideo
{
    class PeliculasCN
    {
        public void AlquilarPelicula(string CodBarrasPelicula, string DniSocio)
        {
            TransactionScope tx = new TransactionScope();
            using(tx)
            {
                // Comprobar el stock
                PeliculasDSTableAdapters.PeliculasTableAdapter adaptadorPeliculas
= new MsdnVideo.PeliculasDSTableAdapters.PeliculasTableAdapter();
                PeliculasDS peliculasDS = new PeliculasDS();
                adaptadorPeliculas.FillByCodigoBarras(peliculasDS.Peliculas,
CodBarrasPelicula);

                if (peliculasDS.Peliculas.Count == 0)
                {
                    throw new ApplicationException("No existe la pelicula");
                }

                if (peliculasDS.Peliculas[0].Stock == 0)
                {
                    throw new ApplicationException("No hay stock de peliculas");
                }

                // Decrementar el stock
                peliculasDS.Peliculas[0].Stock -= 1;
                adaptadorPeliculas.Update(peliculasDS.Peliculas);

                // Alquilar la peliculas
                AlquileresDSTableAdapters.AlquileresTableAdapter
adaptadorAlquileres = new
MsdnVideo.AlquileresDSTableAdapters.AlquileresTableAdapter();
                AlquileresDS alquilerDS = new AlquileresDS();

                alquilerDS.Alquileres.AddAlquileresRow(DniSocio,
CodBarrasPelicula, DateTime.Now, false);
                adaptadorAlquileres.Update(alquilerDS.Alquileres);

                // Completamos la transaccion si todo es correcto
                tx.Complete();
            }
        }
    }
}
```

Ejecutar el Proyecto, hacer pruebas y anotar sus observaciones y comentarios, los mismos que agradeceremos se sirvan remitir al correo wiler.ponce@industrial.unmsm.pe

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA INDUSTRIAL

Curso: Ingeniería de Software

Profesor: Ing. Wiler Arturo Ponce Benites

Correo: wiler.ponce@industrial.unmsm.pe

Página Web: <http://campusvirtualfii.unmsm.edu.pe>

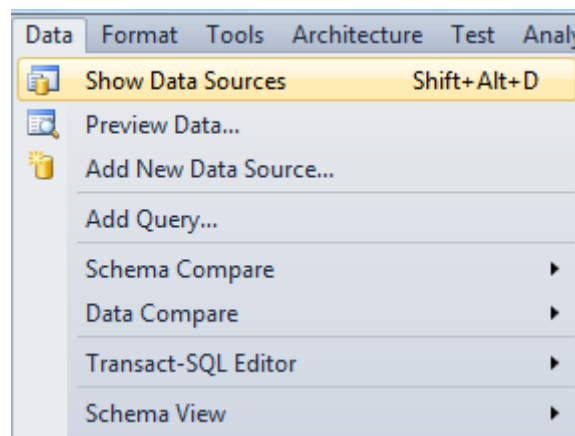
Laboratorio: Diseño de un Proyecto de Software (Parte 5)

Formulario para devolver Películas

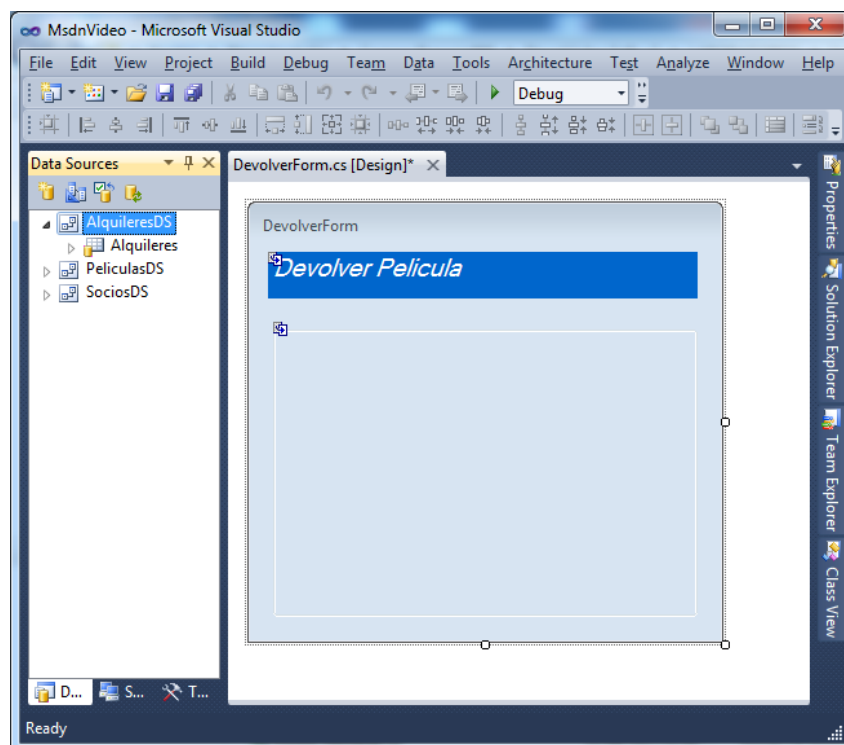
Cargar el formulario **DevolverForm**, que fue diseñado en laboratorio anterior.

Abrir la ventana **Data Sources** Si no estuviera abierta la ventana ir a la barra de comandos y seguir la siguiente secuencia:

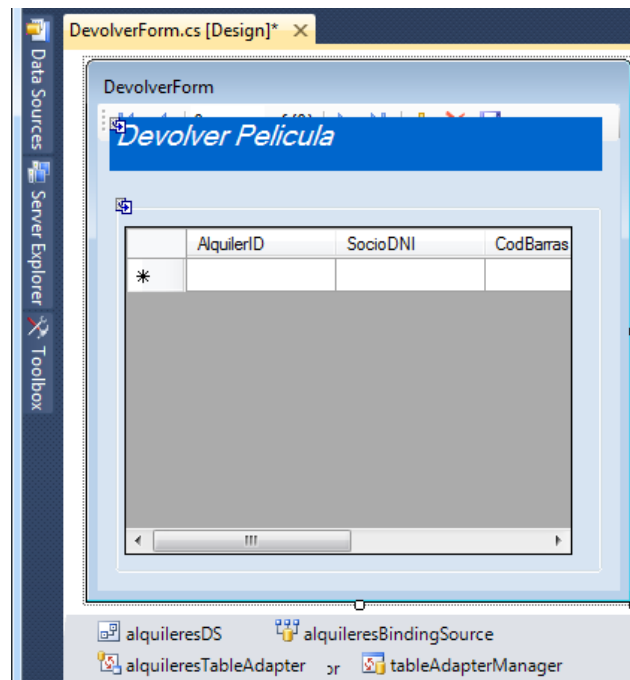
Data / Show Data Sources o presionar simultáneamente las teclas: Shift + Alt + D



Hacer clic sobre el icono Alquileres y arrastrar al formulario.



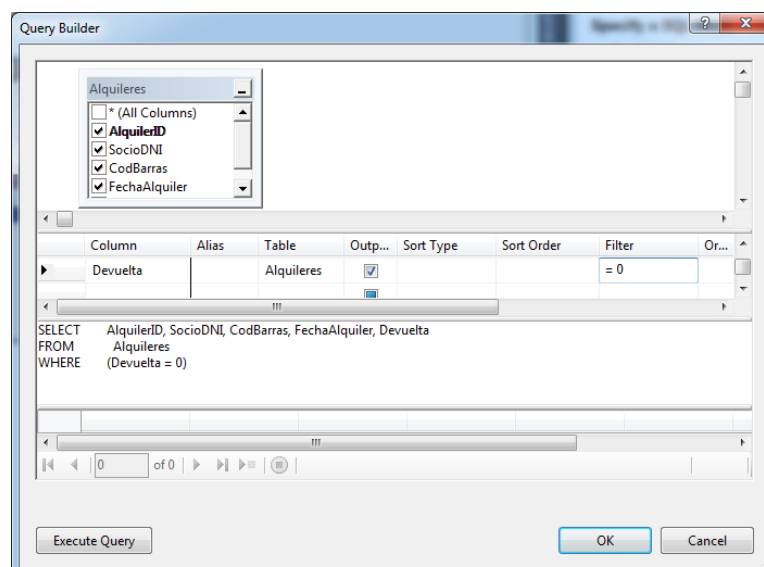
Como resultado, tenemos:



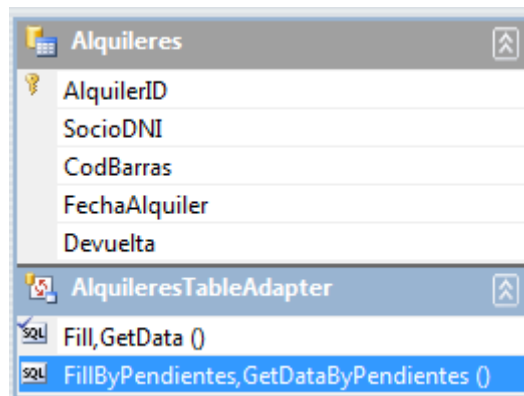
Adicionar y fijar las propiedades necesarias, para los objetos

Button1	(Name)	Devolver
	Text	Devolver
	Anchor	BR
Button2	(Name)	Cancelar
	Text	Cancelar
	Anchor	BR
Form	AcceptButton	Devolver
	CancelButton	Cancelar
DGV	SeleccionMode	FullRowSelect
	Anchor	TBLR

En el dataset de Alquileres generar la siguiente consulta:



Obtenemos el siguiente dataset:



Programación:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace MsdnVideo
{
    public partial class DevolverForm : PlantillaForm
    {
        public DevolverForm()
        {
            InitializeComponent();

            private void DevolverForm_Load(object sender, EventArgs e)
            {
                // TODO: This line of code loads data into the
                'alquileresDS.Alquileres' table. You can move, or remove it, as needed.
                this.alquileresTableAdapter.FillByPendientes(this.alquileresDS.Alquileres);
            }

            private void Devolver_Click(object sender, EventArgs e)
            {
                AlquileresDS.AlquileresRow rowAlquileres;
                foreach (DataGridViewRow fila in alquileresDataGridView.SelectedRows)
                {
                    rowAlquileres =
                    (AlquileresDS.AlquileresRow)alquileresDS.Alquileres[fila.Index];

                    PeliculasCN peliculasCN = new PeliculasCN();
                    peliculasCN.DevolverPelicula(rowAlquileres);
                }
                Close();
            }
        }
    }
}
```



```

        private void Cancelar_Click(object sender, EventArgs e)
        {
            Close();
        }

        // private void alquileresBindingNavigatorSaveItem_Click(object sender,
        // EventArgs e)
        // {
        //     // this.Validate();
        //     // this.alquileresBindingSource.EndEdit();
        //     // this.tableAdapterManager.UpdateAll(this.alquileresDS);
        // }
    }
}

```

Ejecutar el proyecto, probar el formulario, anotar sus observaciones y comentarios, agradeceremos enviarlos al correo del profesor: wiler.ponce@industrial.unmsm.pe lo que nos permitirá mejorar la guía. Gracias.

WPB/2015