**Desarrollo de Aplicaciones Empresariales Avanzadas**

**GUÍA DE LABORATORIO 1**

**“ADO.NET”**

**Docente: Hugo Torrico Márquez**

***LABORATORIO***

***Objetivos:***

* Conocer y dominar la metodología ADO.NET.

***Equipos, Materiales, Programas y Recursos:***

* PC con Sistema Operativo Windows o Linux
* Notepad++ o Sublime Text

***Introducción:***

En la presente sesión se detalla los fundamentos para la conexión entre Visual Studio.Net y SQL Server mediante la tecnología ADO.NET

***Seguridad:***

* Ubicar maletines y/o mochilas en el gabinete al final de aula de laboratorio.
* No ingresar con líquidos ni comida al aula de laboratorio.
* Al culminar la sesión de laboratorio, apagar correctamente la computadora y el monitor.

***Preparación:***

Durante el desarrollo de los temas de clase se tendrán ejercicios explicativos en cada uno de los puntos, ello le dará a la sesión una interacción de la teoría y la parte práctica, ya que en todo el momento el alumno podrá comprobar en su propia PC, todos los ítems del manual.

**Objetivo:**

**- Comprende los diferentes tipos de conexión a una base de datos con SQL Server.**

**- Implementa aplicaciones de conexión a base de datos SQL Server.**

**- Configura de manera adecuada el app.config para aplicación con conexión a base de datos de SQL Server.**

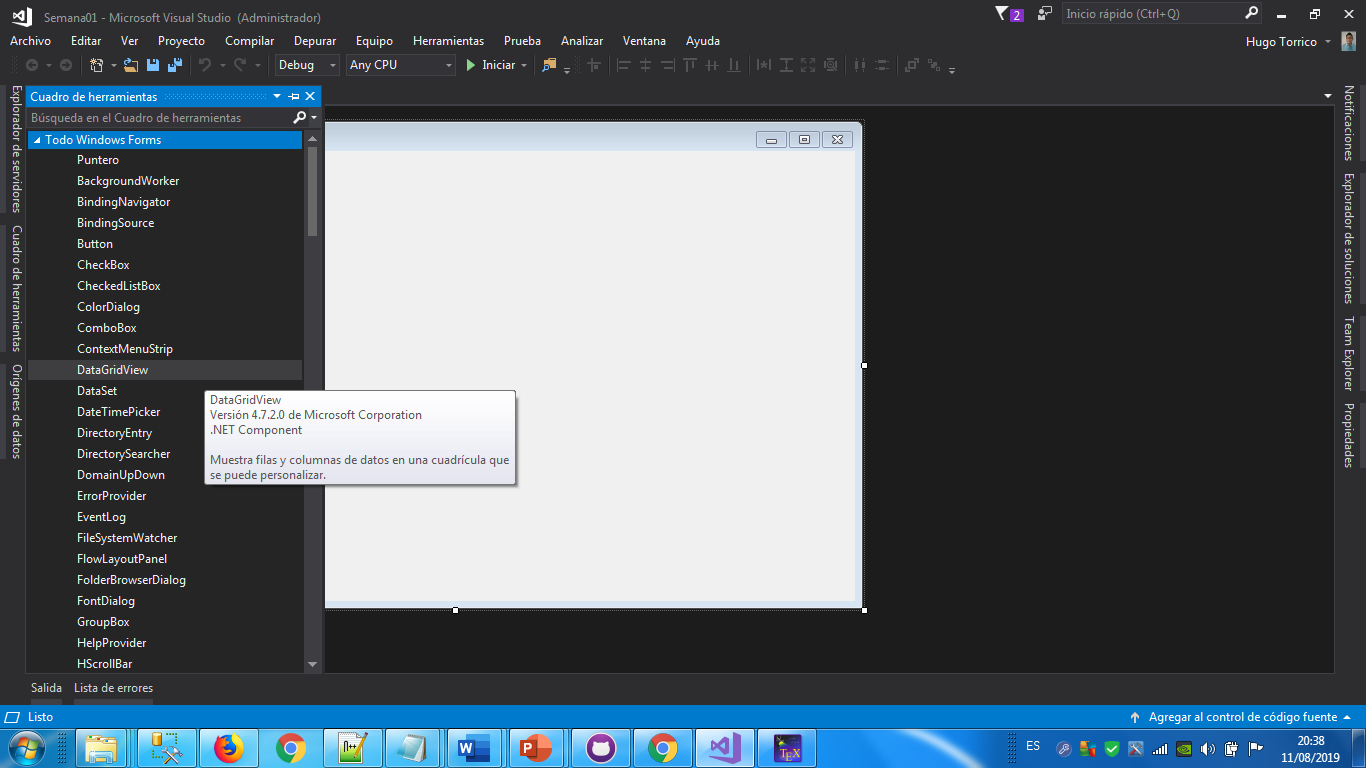
**Enunciado 01 : Listado de Clientes usando el Asistente de Conexión**

**Paso 01**

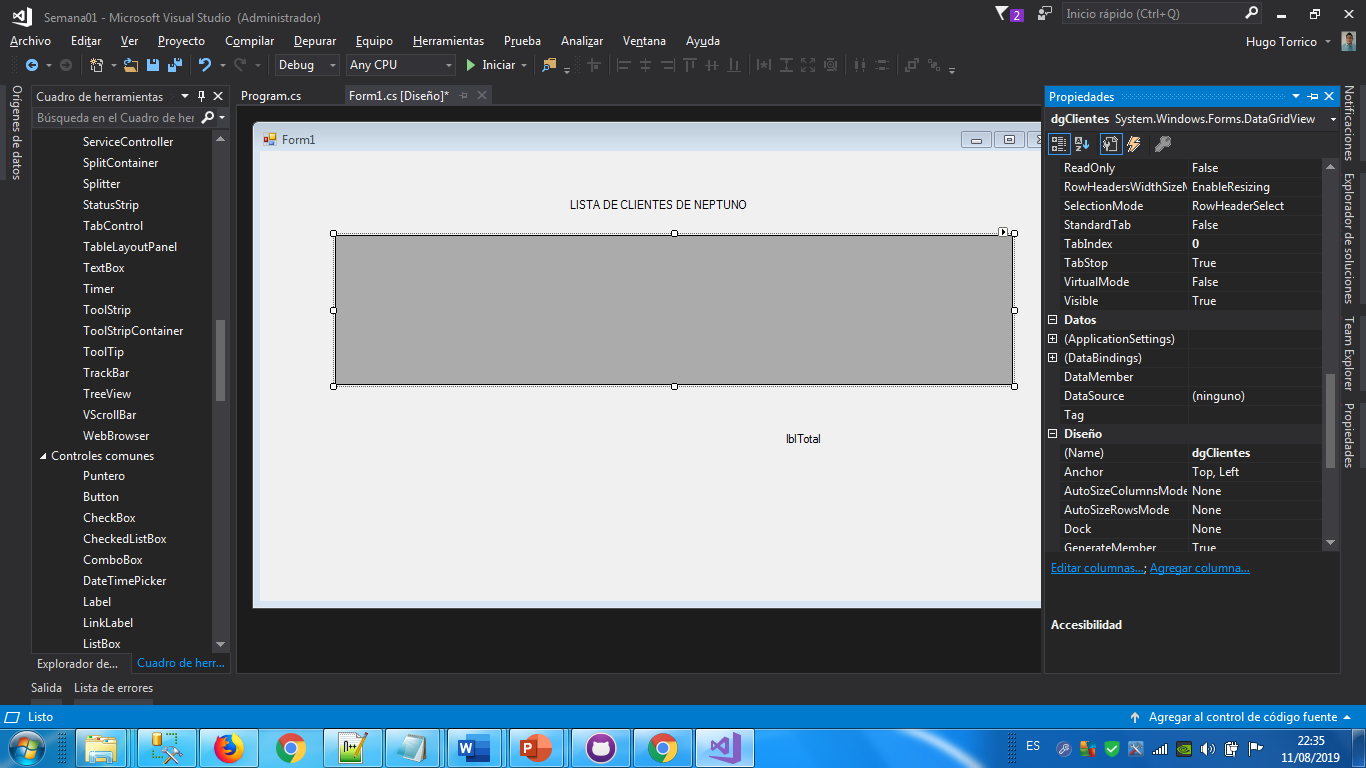
1. Crear un proyecto en Visual Studio, de tipo CSharp, Seleccionar aplicaciones de Windows Forms y Asignar el Nombre Semana01



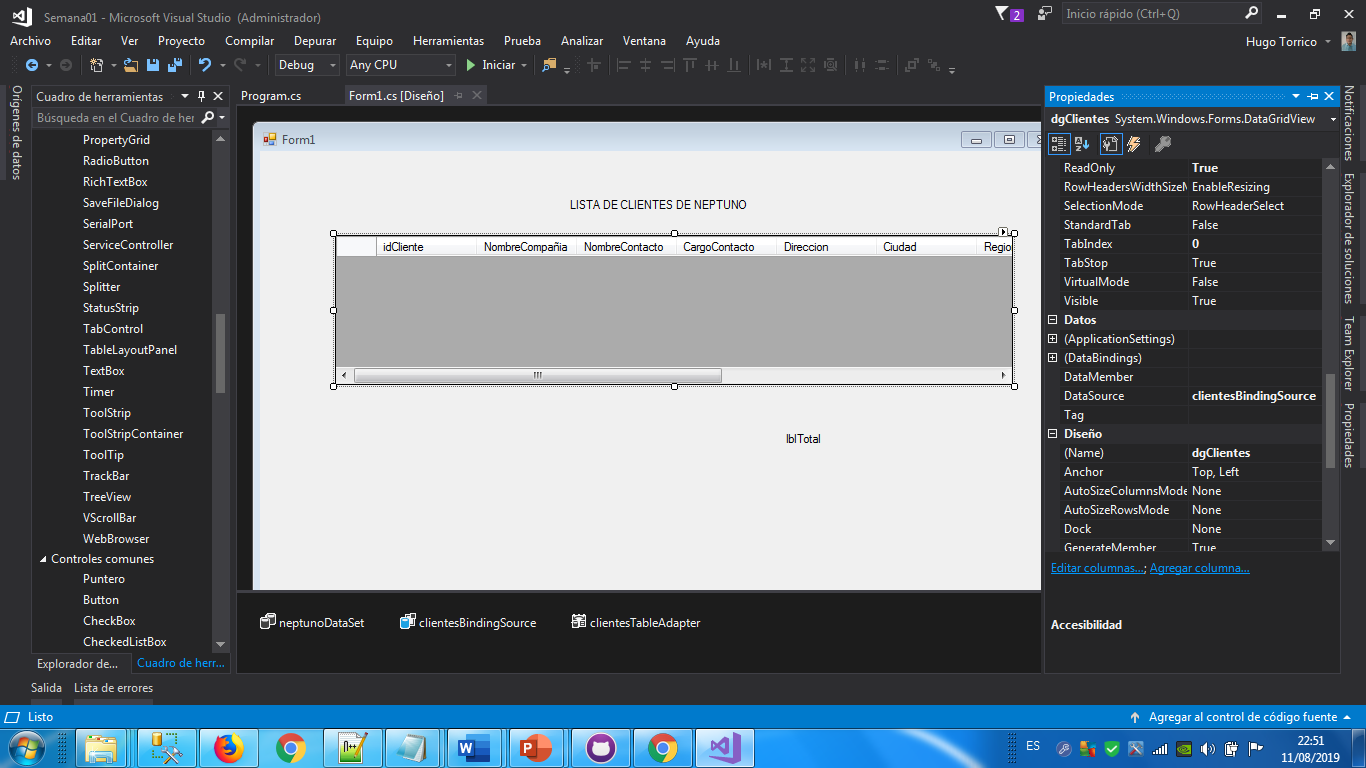
1. Agregar un datagrid y un label de la barra de herramientas



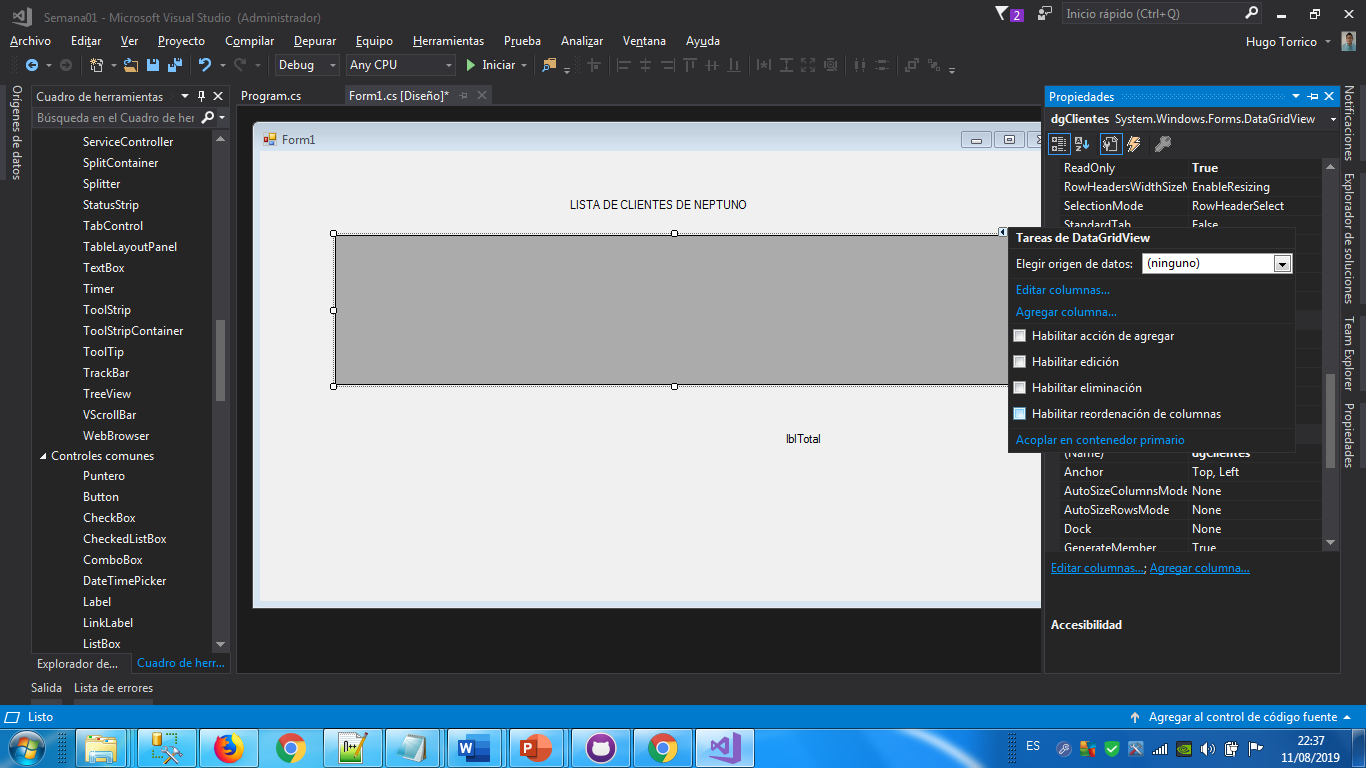
1. Diseñar el Siguiente GUI utilizando los controles de C Sharp, arrastra los controles de la barra de herramientas de las sección izquierda y cambiar el nombre de los controles en la sección de propiedades.



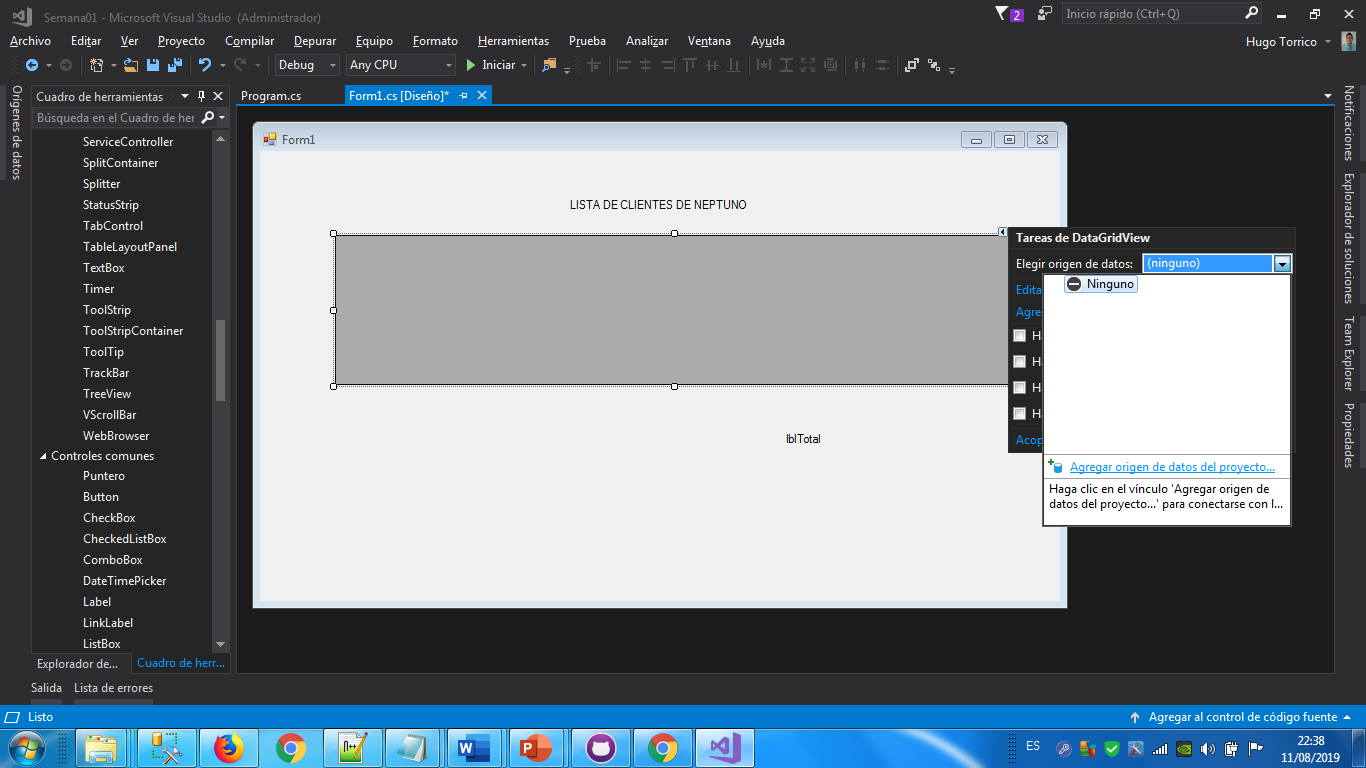
1. Cambiar el nombre de los controles en la propiedad Name, lblTotal y dgClientes.



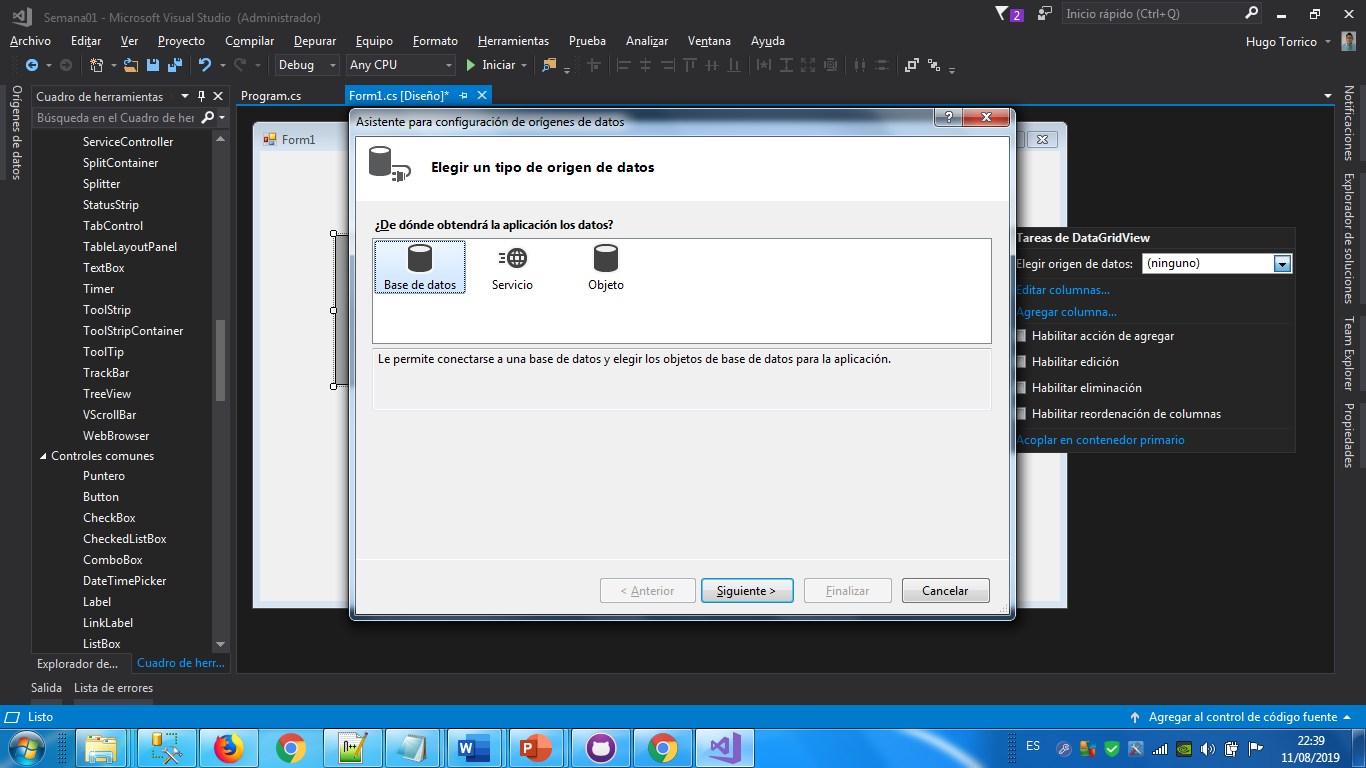
1. Deshabilitar las opciones del Datagridview



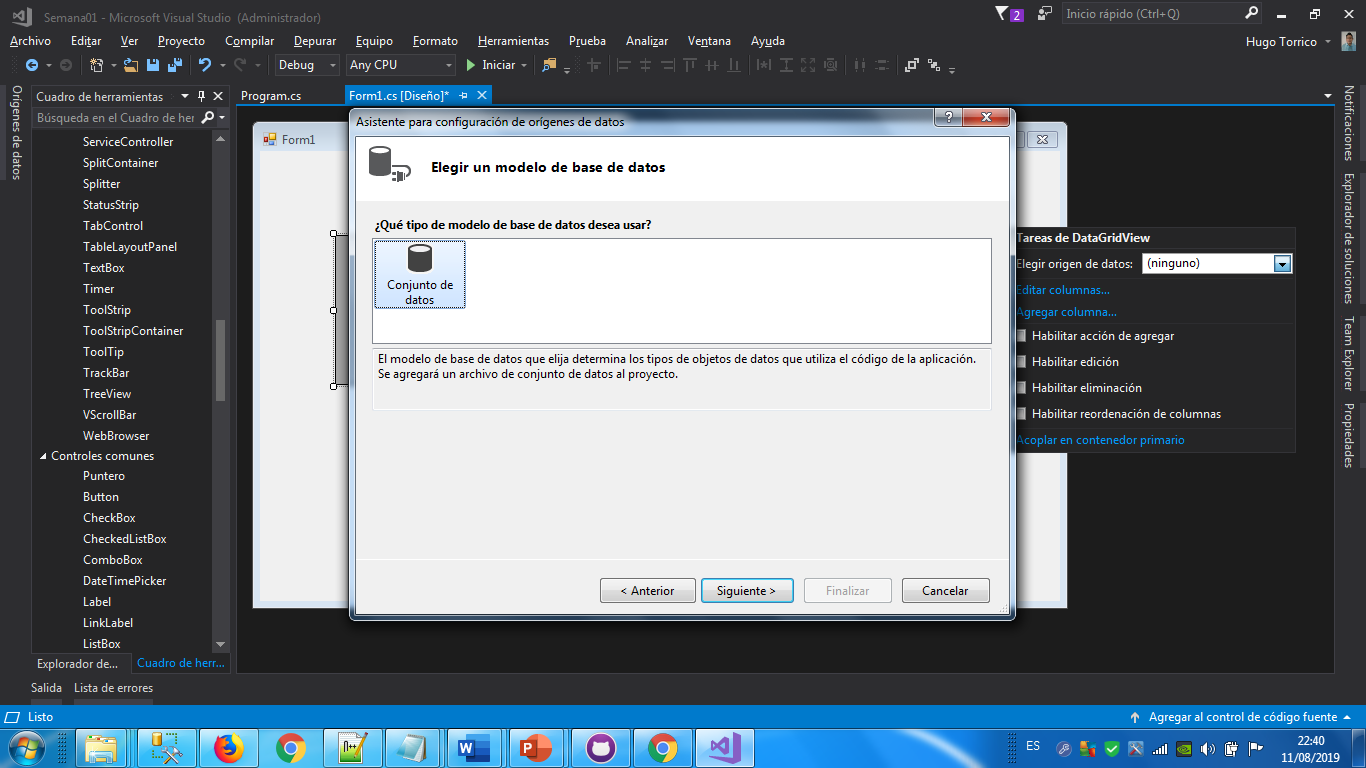
1. Clic Sobre la pestaña del Datagridview>Seleccione la opción Agregar Origen e Datos del Proyecto…



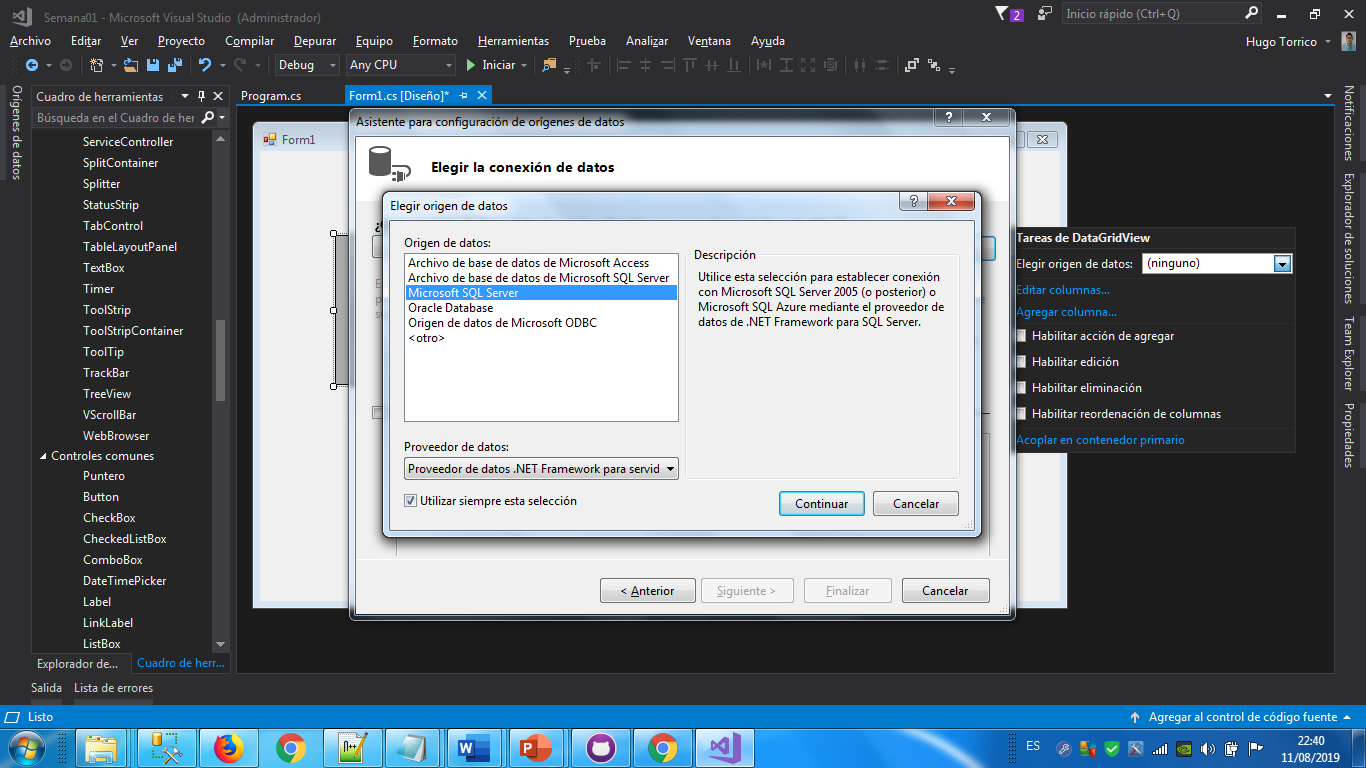
1. Seleccione de donde se obtendrán los datos a mostrar en la grilla, para lo cual debe seleccionar la base de datos



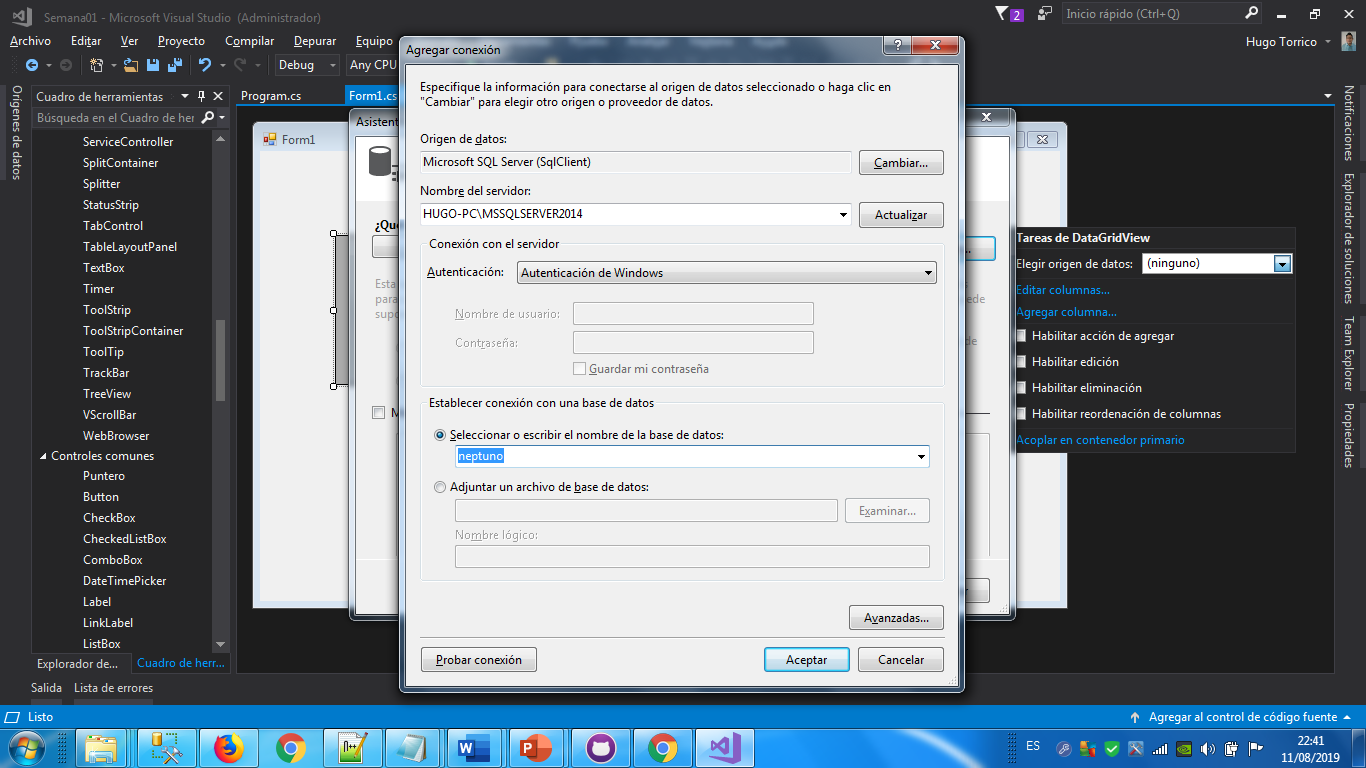
1. Seleccione el Modelo de Datos>Conjunto e Datos



1. Seleccionar el Origen de Datos



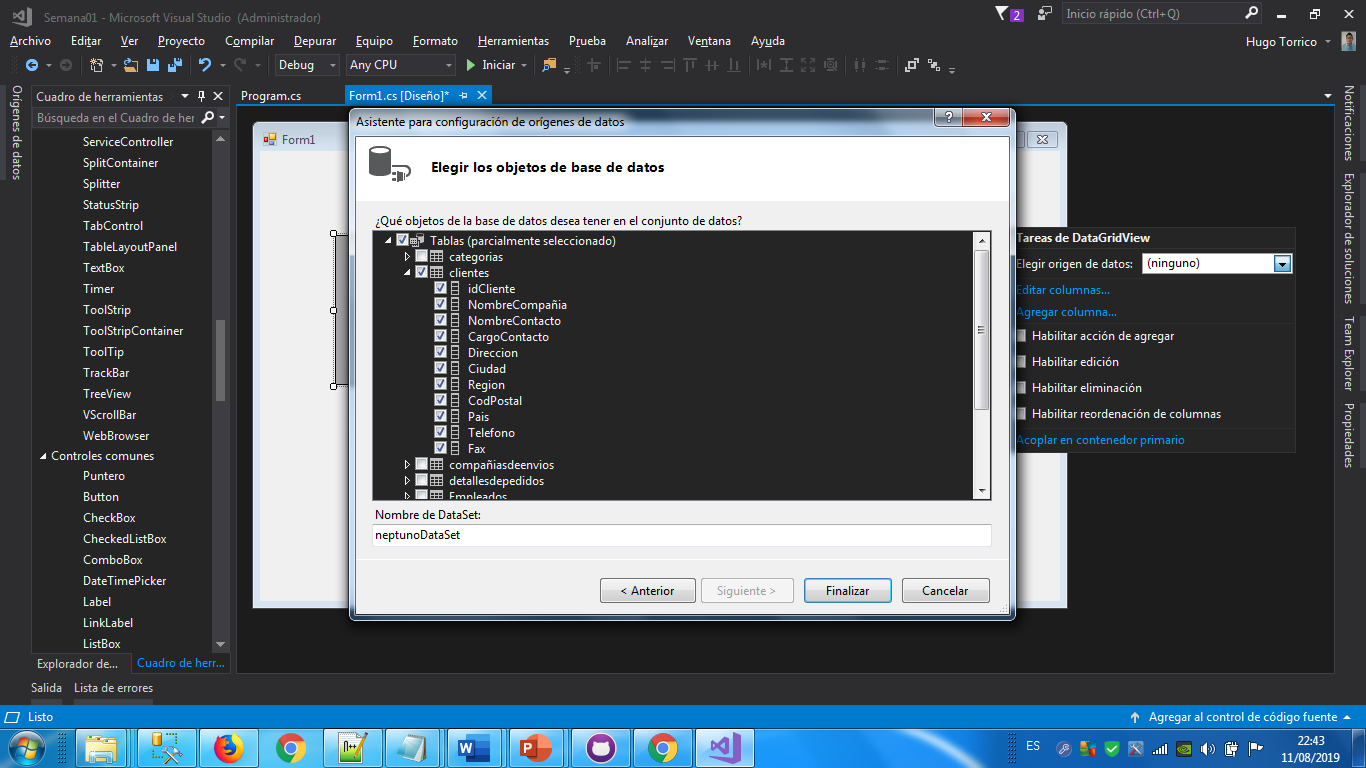
1. Ingresar los Valores para la Conexión(no olvidar realizar la conexión local, con la autentificación Windows y Seleccionar la base de datos Neptuno



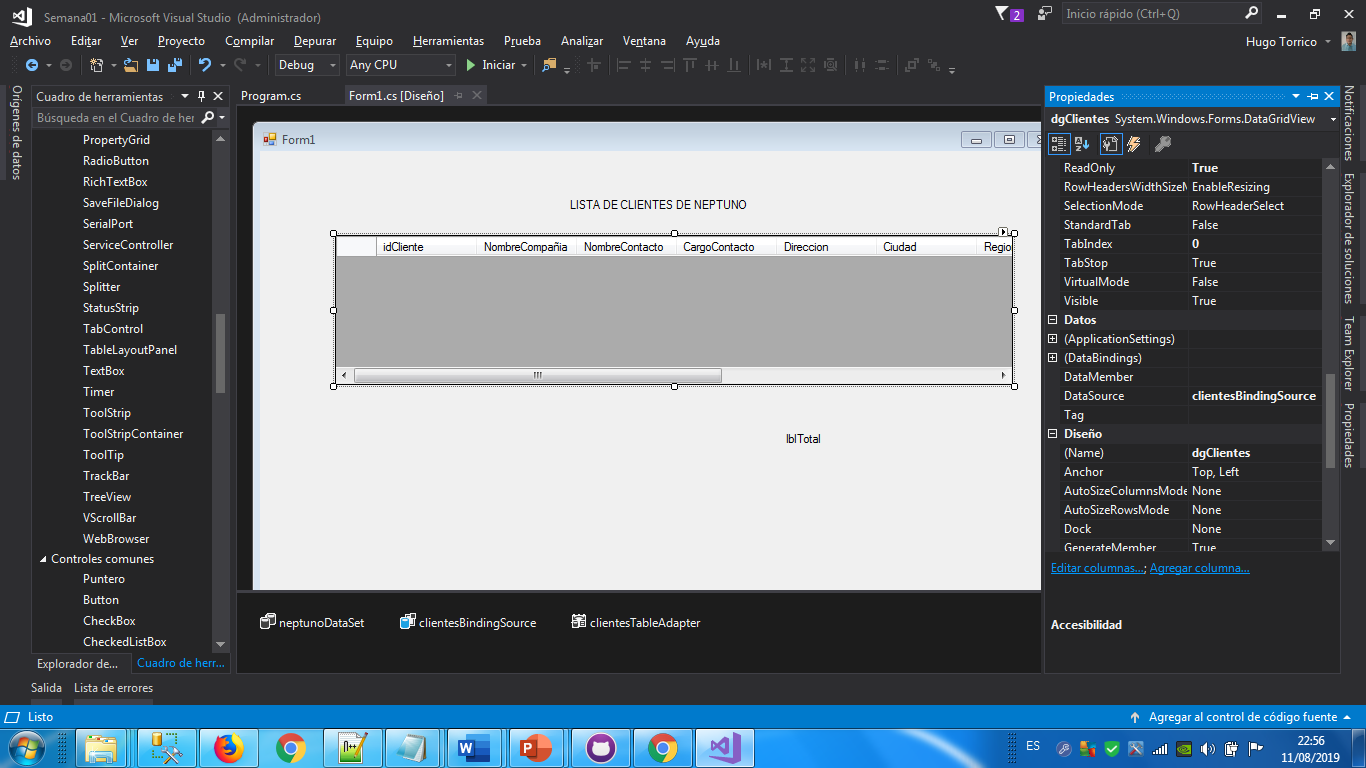
1. Visualizar la ruta conexión a la base de datos



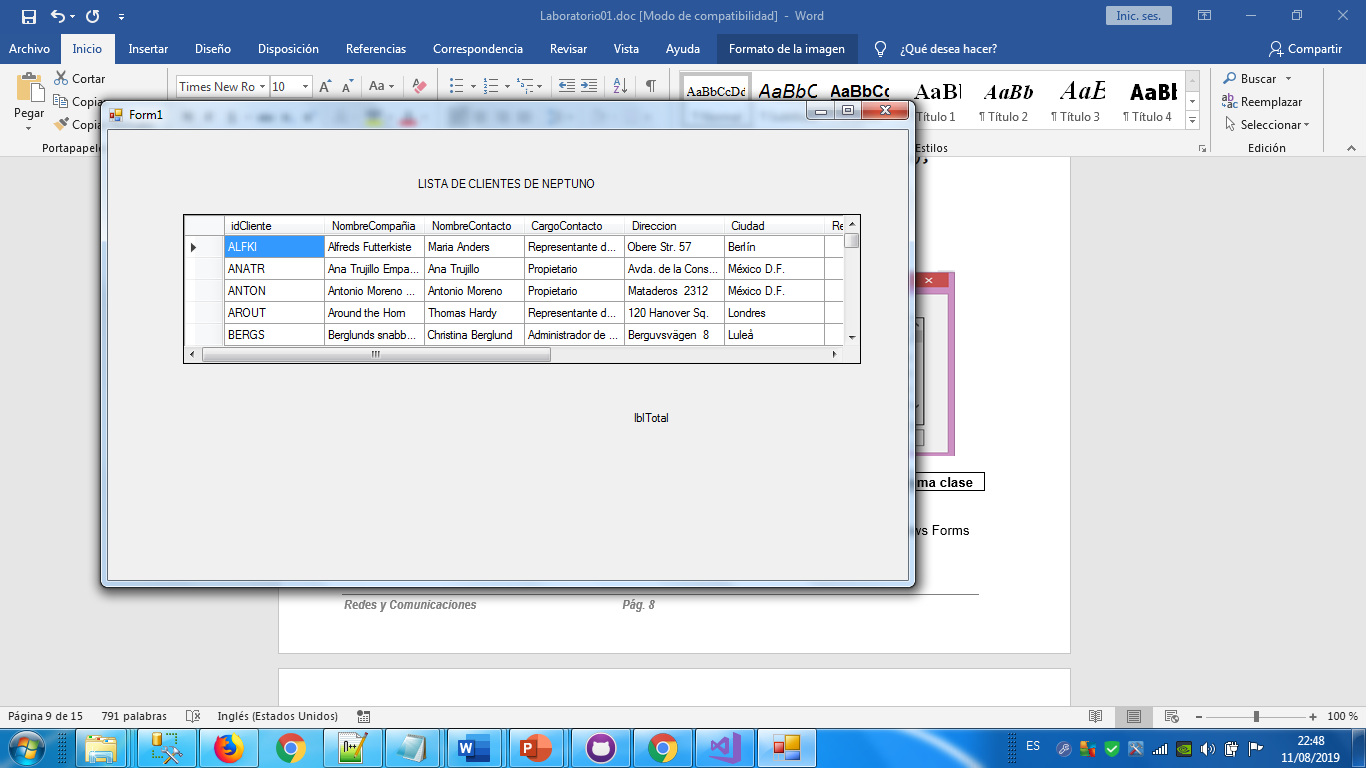
1. Seleccionar el objeto de la base de datos (tabla Clientes)



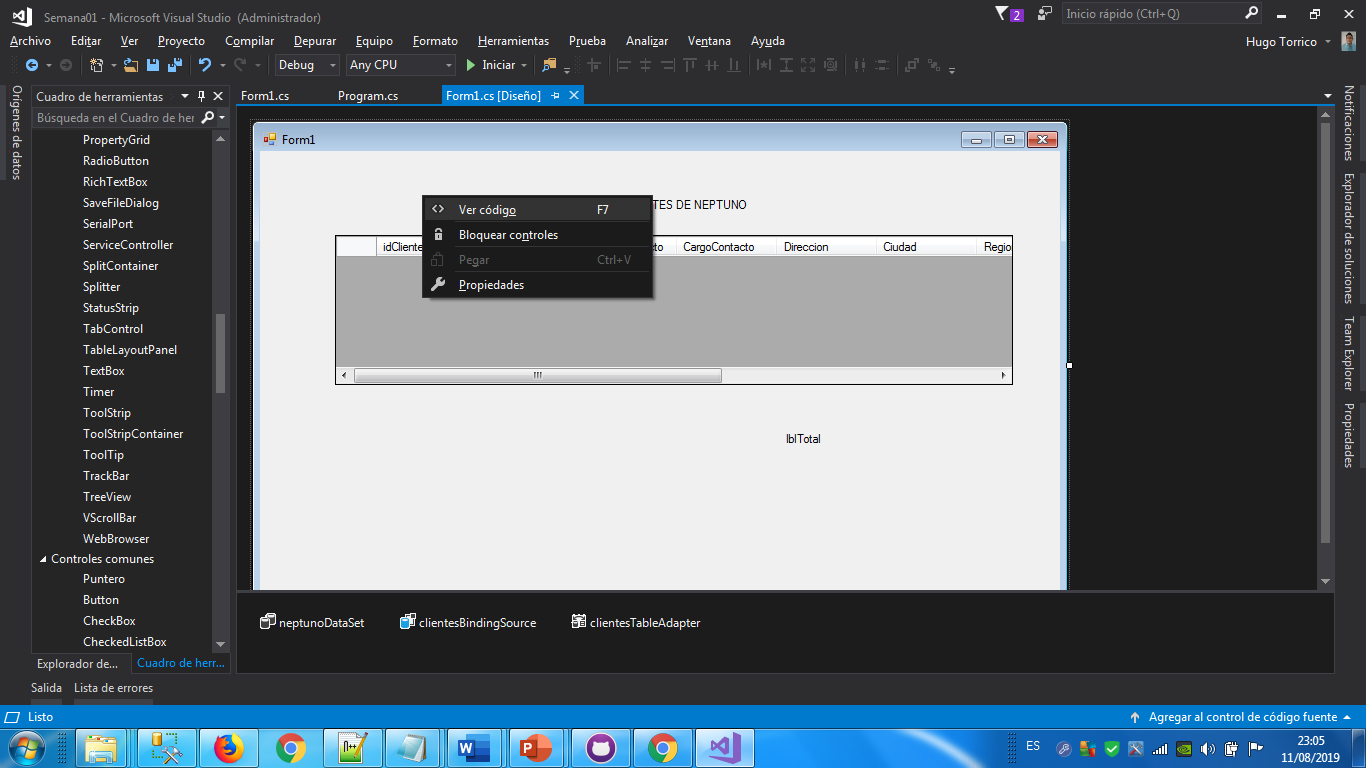
1. Finalmente visualizamos los campos de la tabla



1. Ejecutar la aplicación



1. Seleccionar el formulario, click derecho y ver código.



1. Complementar con la opción de contar las filas del Datagridview

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Semana01

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: esta línea de código carga datos en la tabla 'neptunoDataSet.clientes' Puede moverla o quitarla según sea necesario.

this.clientesTableAdapter.Fill(this.neptunoDataSet.clientes);

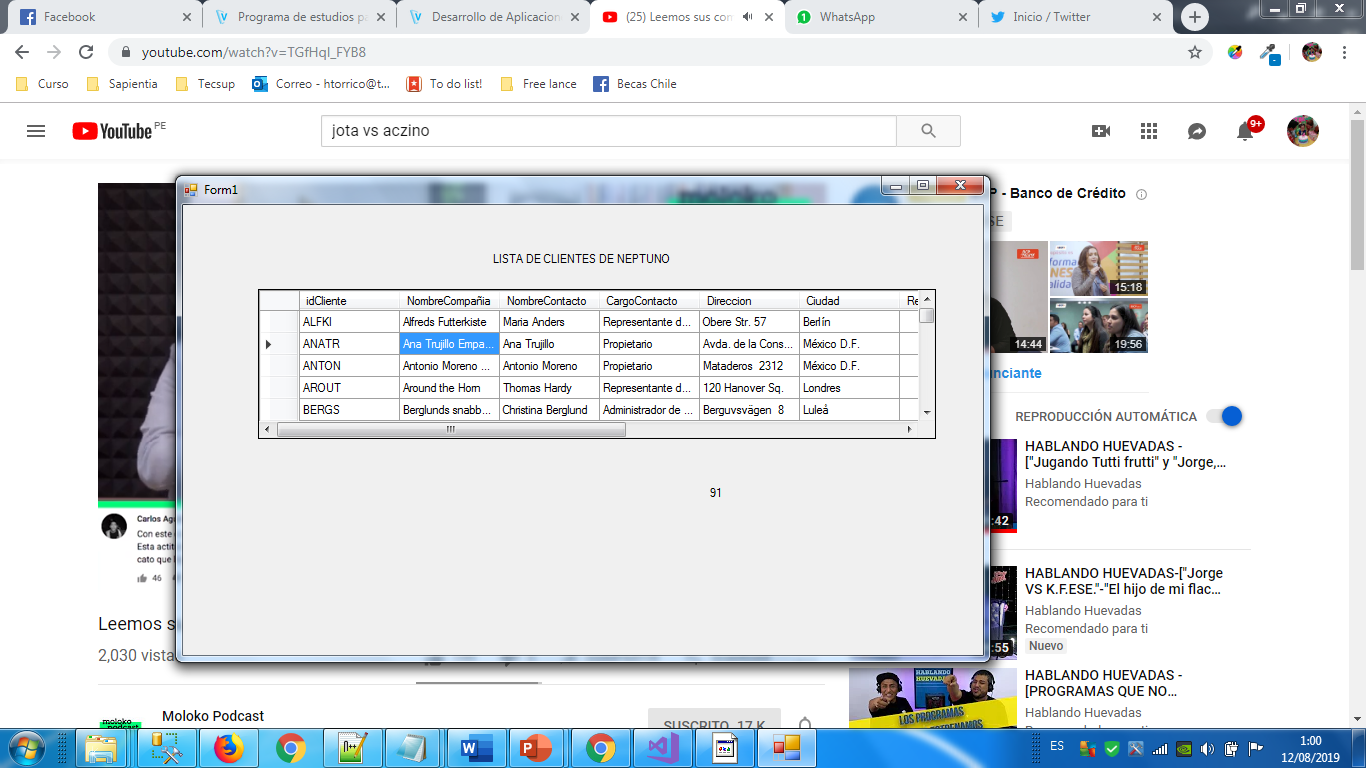
lblTotal.Text = dgClientes.RowCount.ToString();

}

}

}

1. Volver a Ejecutar



**Enunciado 02 : Listado de Clientes implementando la cadena de conexión en la misma clase**

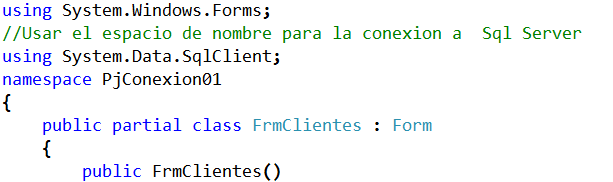
**Pasos:**

1. Crear un proyecto nuevo en Visual Studio 2017 de tipo Visual C# > Aplicación de Windows Forms y asígnele el nombre pjConexion2.
2. Diseñe la siguiente GUI.

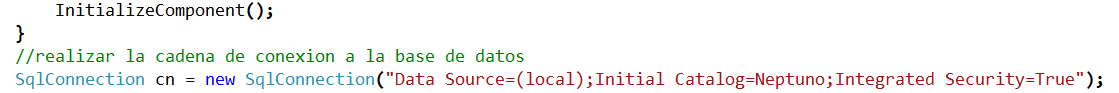


3.- Implementar el siguiente código





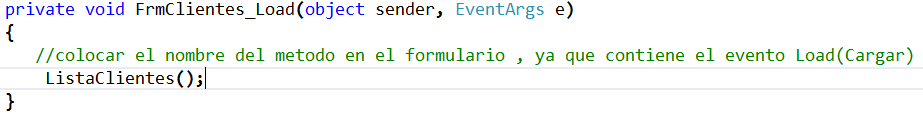
Definir la siguiente Ruta de Conexión a la Base de Datos Neptuno



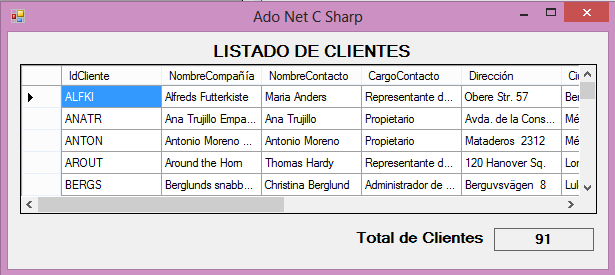
Crear el siguiente Método



Colocar el método dentro del formulario en el evento Load



Ejecutar la aplicación.

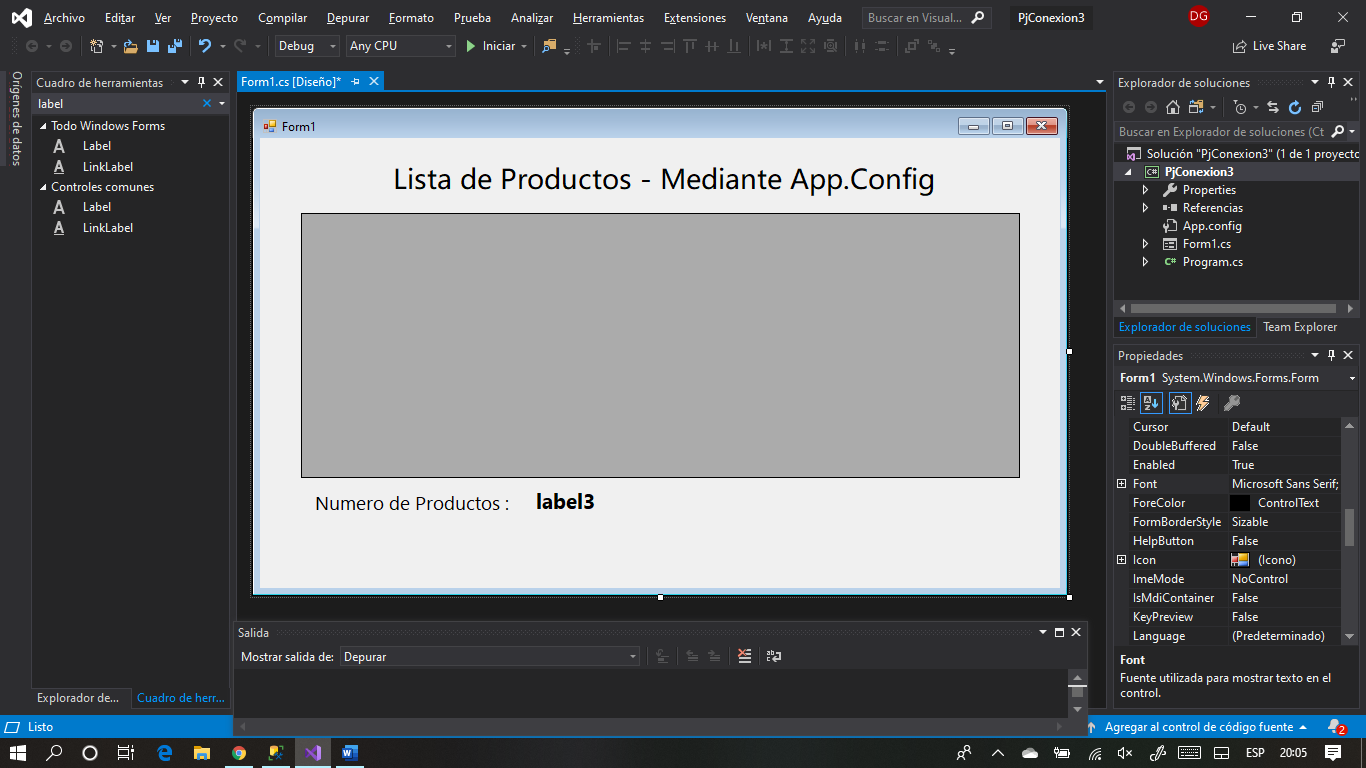


**Enunciado 03 : Listado de Productos utilizando ConfigurationManager**

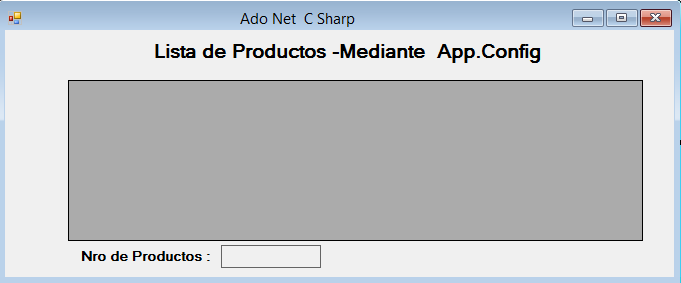
Aplicación que permita una lista de los productos de la base de datos Neptuno, también debe determinar el número de productos Registrados

**Pasos:**

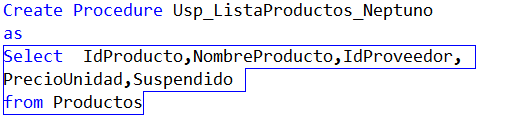
1. Crear un proyecto nuevo en C Sharp > Aplicación de Windows Forms y asígnele el nombre PjConexion3.
2. Diseñar la siguiente GUI usando los Controles de C Sharp

=================================================================================

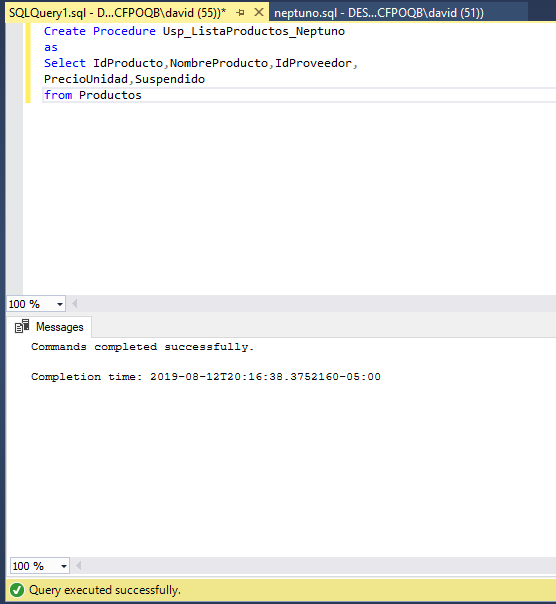
=================================================================================



1. Crear el Siguiente Procedimiento Almacenado.



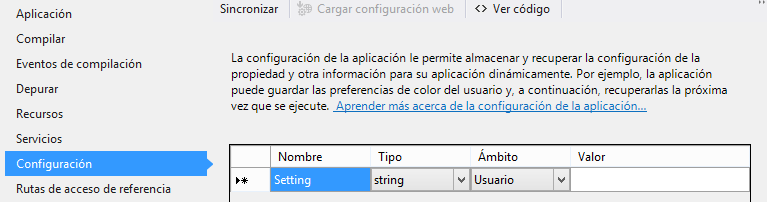
====================================================================================



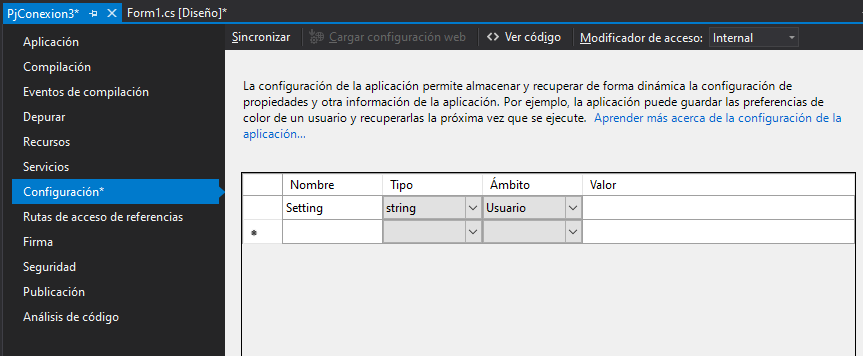
====================================================================================

1. Crear la Siguiente Conexión a la base de datos mediante el App.Config.

Ingresar al Menu Proyecto>Propiedades de PjConexion03...>Seleccionar la opción “Configuración”.



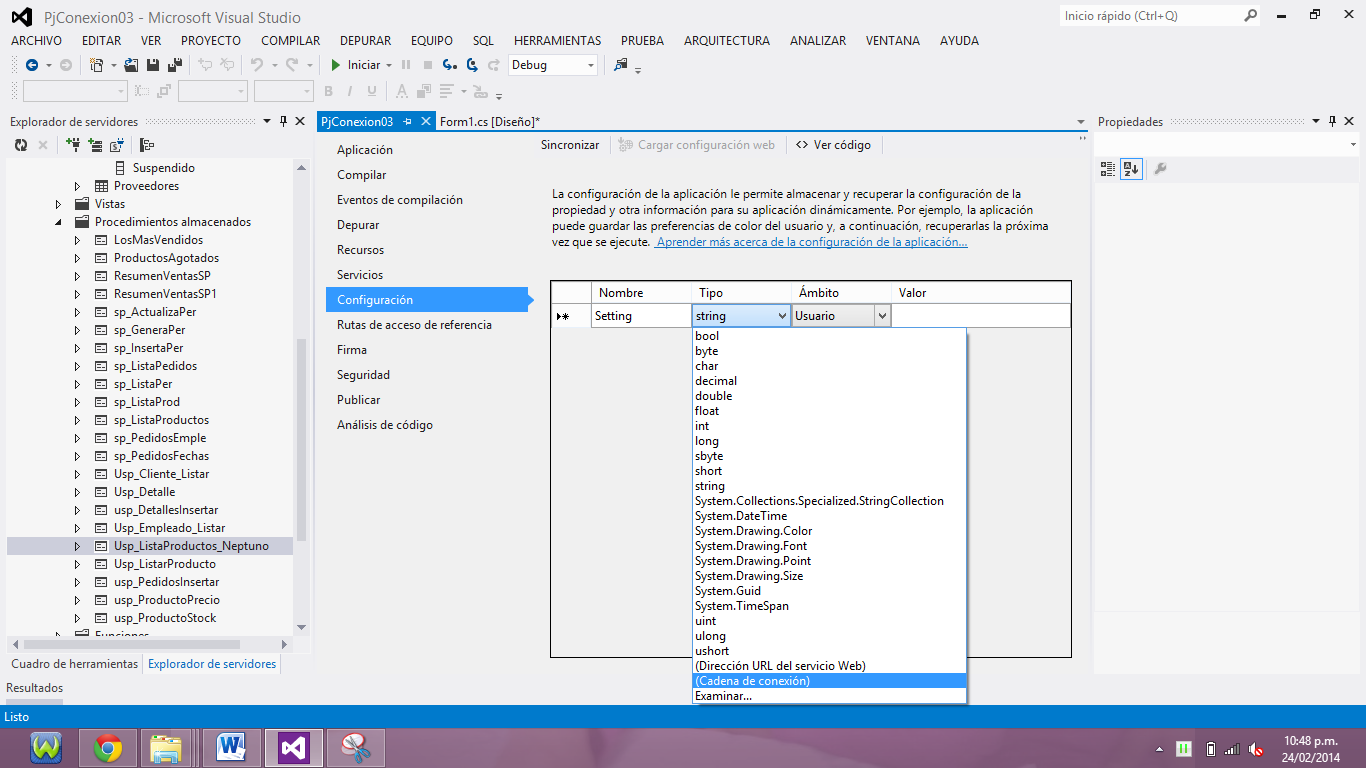
=================================================================================



=================================================================================

1. Seleccionar En la opción Tipo>Cadena de Conexión.

Después de seleccionar la Cadena de Conexión , ingresar a la opción Valor y especificar la ruta de conexión a la base de datos

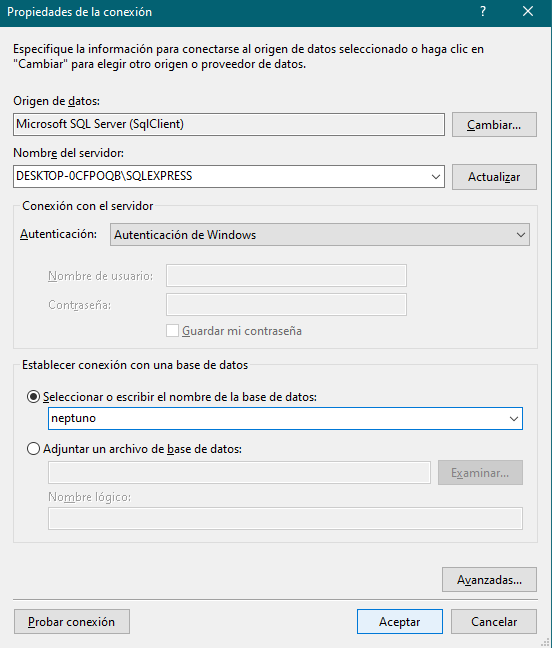


=================================================================================



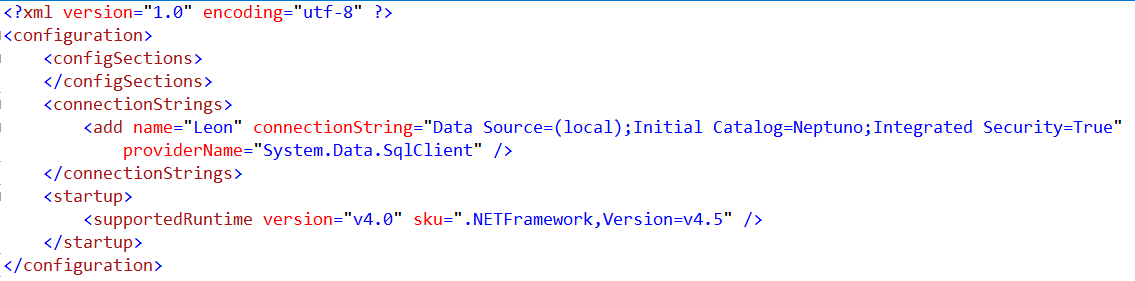
=================================================================================

=================================================================================

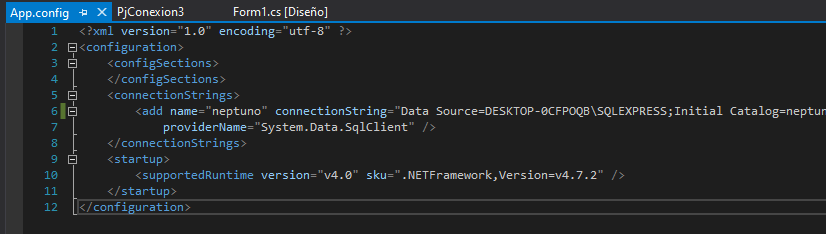


=================================================================================

1. Verificar la conexión a la base de datos mediante el App.Config y Cambiar de nombre

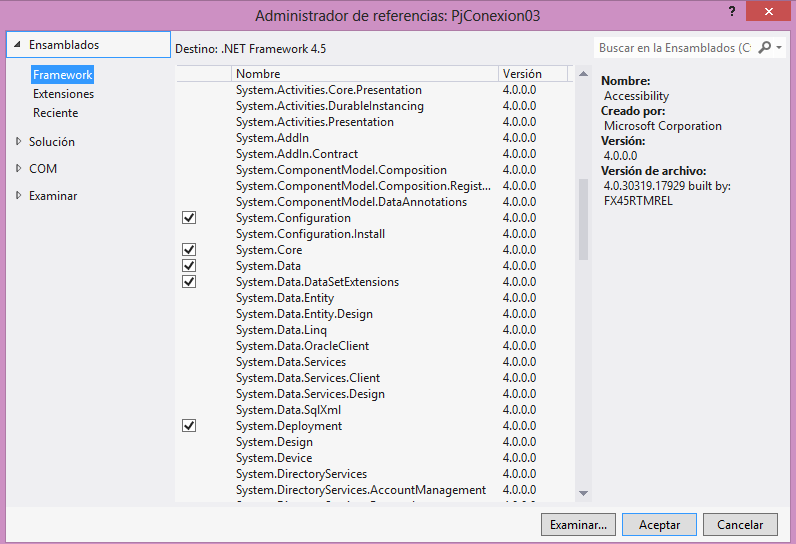


=================================================================================

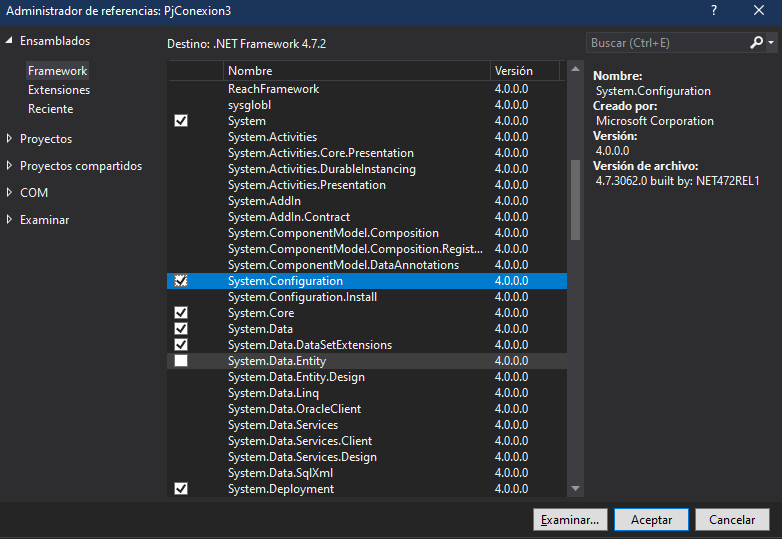


=================================================================================

1. Ingresar al Menu Proyecto >Agregar Referencia….

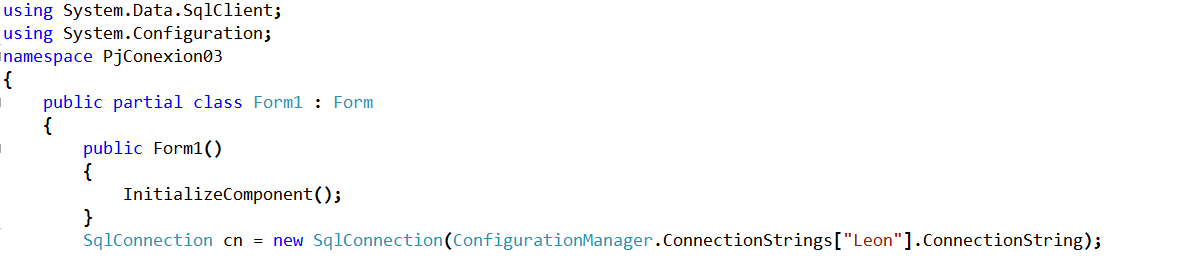


=================================================================================

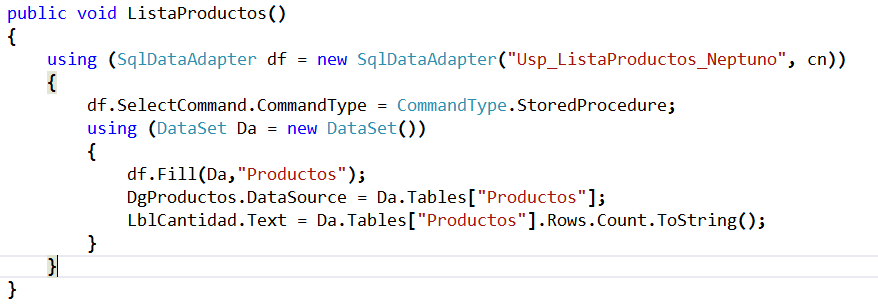


=================================================================================

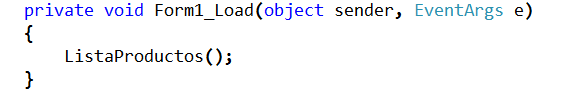
1. Definir los espacios de nombres requeridos y la conexión al App.config

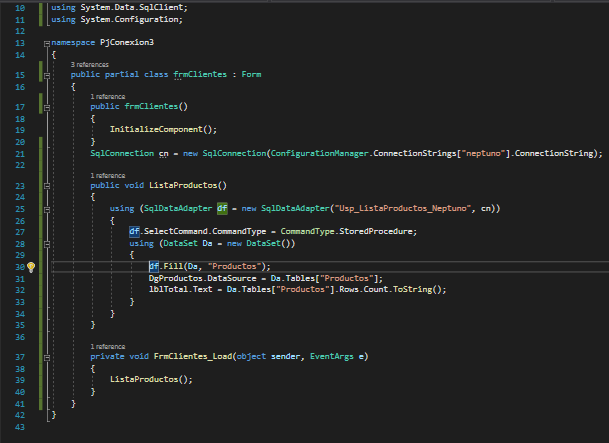


1. Crear un método para la lista de productos

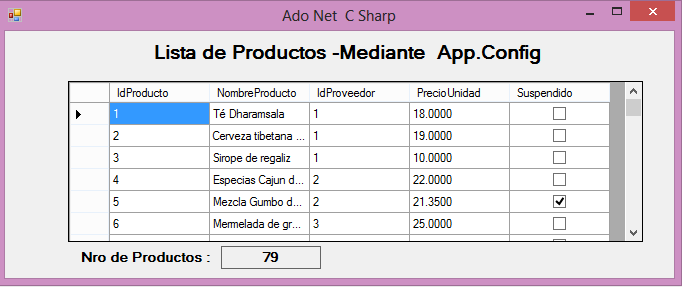


1. Llamar al método al formulario y ejecutar la Aplicación

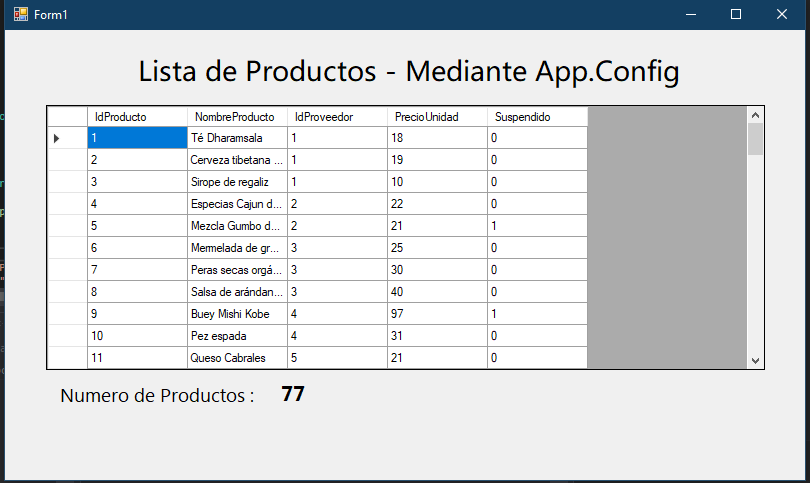


=================================================================================

=================================================================================



=================================================================================



=================================================================================

**Casos Propuestos**

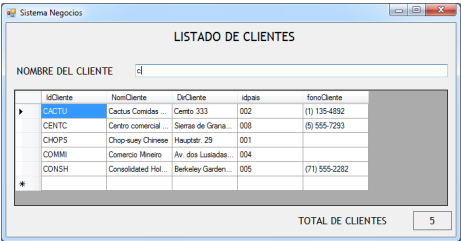
**Caso Propuesto Nº 01:**

Aplicación que permite mostrar un listado de los clientes registrados en la base Negocios filtrado por las iniciales de su nombre de cliente, al final deberá determinar el total de clientes encontrados. Use:

- ConfigurationManager para la conexión a la base de datos.

- Procedimiento almacenado para listar a los clientes.

- Consulta simple para el filtrado de los clientes.



CODIGO:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Configuration;

namespace CasoPropuesto1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

SqlConnection cn = new SqlConnection(ConfigurationManager.ConnectionStrings["neptuno"].ConnectionString);

public void ListaClientes()

{

using (SqlDataAdapter df = new SqlDataAdapter("Usp\_ListaClientes\_Neptuno", cn))

{

df.SelectCommand.CommandType = CommandType.StoredProcedure;

using (DataSet Da = new DataSet())

{

df.Fill(Da, "Clientes");

DgClientes.DataSource = Da.Tables["Clientes"];

txtTotal.Text = Da.Tables["Clientes"].Rows.Count.ToString();

}

}

}

public void BusquedaClientes()

{

using (SqlDataAdapter Df = new SqlDataAdapter("Usp\_ListaClientes\_Neptuno\_2", cn))

{

SqlParameter sqlParameter = new SqlParameter();

sqlParameter.Value = txtFiltro.Text;

sqlParameter.SqlDbType = SqlDbType.VarChar;

sqlParameter.Size = 100;

sqlParameter.ParameterName = "@idCliente";

Df.SelectCommand.Parameters.Add(sqlParameter);

Df.SelectCommand.CommandType = CommandType.StoredProcedure;

using (DataSet Da = new DataSet())

{

Df.Fill(Da, "Clientes");

DgClientes.DataSource = Da.Tables["Clientes"];

txtTotal.Text = Da.Tables["Clientes"].Rows.Count.ToString();

}

}

}

private void Label1\_Click(object sender, EventArgs e){}

private void Form1\_Load(object sender, EventArgs e)

{

ListaClientes();

}

private void BtnFiltro\_Click(object sender, EventArgs e)

{

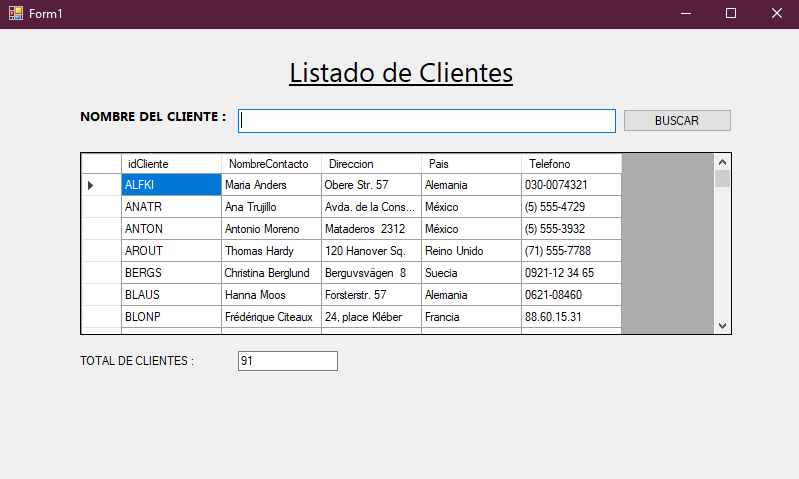
BusquedaClientes();

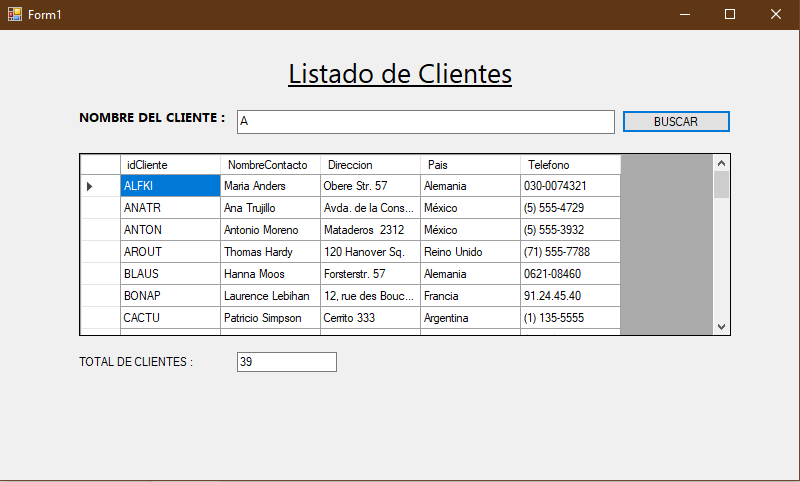
}

}

}

CAPTURAS





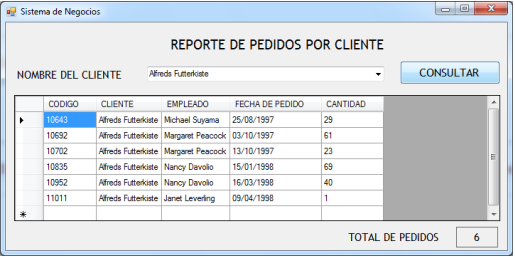
**Caso Propuesto Nº 02:**

Aplicación que permite mostrar un listado de pedidos realizados por un determinado cliente. Use:

- ConfigurationManager para la conexión a la base de datos.

- Procedimiento almacenado para listar los pedidos.

- Consulta combinada para el filtrado de los pedidos realizados según el cliente seleccionado desde el cuadro combinado.



CÓDIGO:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Configuration;

namespace CasoPropuesto1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

SqlConnection cn = new SqlConnection(ConfigurationManager.ConnectionStrings["neptuno"].ConnectionString);

public void ListaClientes()

{

using (SqlDataAdapter df = new SqlDataAdapter("Usp\_ListaClientes\_Neptuno", cn))

{

df.SelectCommand.CommandType = CommandType.StoredProcedure;

using (DataSet Da = new DataSet())

{

df.Fill(Da, "Clientes");

DgClientes.DataSource = Da.Tables["Clientes"];

txtTotal.Text = Da.Tables["Clientes"].Rows.Count.ToString();

}

}

}

public void BusquedaClientes()

{

using (SqlDataAdapter Df = new SqlDataAdapter("Usp\_ListaClientes\_Neptuno\_2", cn))

{

SqlParameter sqlParameter = new SqlParameter();

sqlParameter.Value = txtFiltro.Text;

sqlParameter.SqlDbType = SqlDbType.VarChar;

sqlParameter.Size = 100;

sqlParameter.ParameterName = "@idCliente";

Df.SelectCommand.Parameters.Add(sqlParameter);

Df.SelectCommand.CommandType = CommandType.StoredProcedure;

using (DataSet Da = new DataSet())

{

Df.Fill(Da, "Clientes");

DgClientes.DataSource = Da.Tables["Clientes"];

txtTotal.Text = Da.Tables["Clientes"].Rows.Count.ToString();

}

}

}

private void Label1\_Click(object sender, EventArgs e){}

private void Form1\_Load(object sender, EventArgs e)

{

ListaClientes();

}

private void BtnFiltro\_Click(object sender, EventArgs e)

{

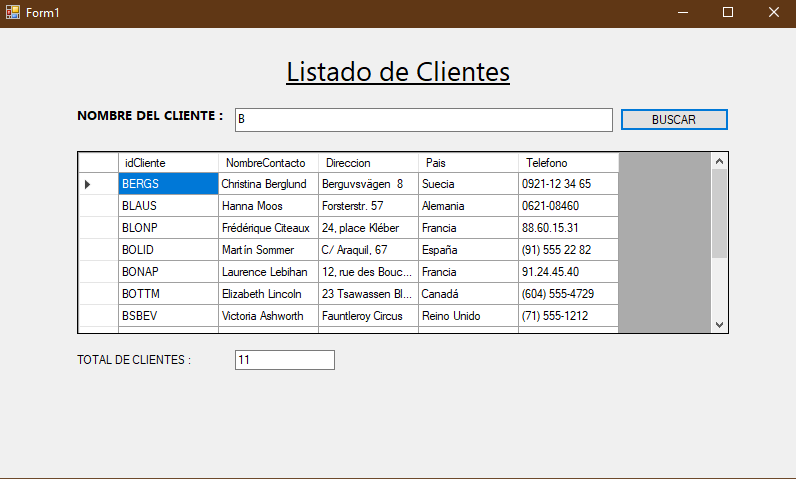
BusquedaClientes();

}

}

}

CAPTURAS



ENLACE DE GIT:

https://github.com/DavidGarciaGiron/Semana01-DesAEA.git