

## Tema 1.4.9. Conexión a BBDD

### Índice

1. Instalación de un Servidor SQL para Bases de Datos.....	1
2. Gestión del Servidor SQL: SQL Server Management Studio SSMS .....	2
3. Conexión en Visual Studio de la Base de Datos.....	3
4. Utilización de una Base de Datos para un Proyecto WPF.....	9
4.1. Conectar la Base datos a nuestro proyecto .....	9
4.2. Creación de la <b>cadena de conexión</b> para las Window (diseño gráfico).....	14
4.3. Ver los datos de una tabla .....	18
4.4. Ver los datos de una tabla desde la MainWindow.xaml.....	20
4.4.1. Diseño MainWindow.xaml.....	20
4.4.2. Configuración de la MainWindow.xaml.cs .....	20

### 1. Instalación de un Servidor SQL para Bases de Datos

Primero necesitas crear una **Base de datos en local**, para ello y si no tienes **LocalDB de SQL Server Express** en tu PC. Ve a la página y busca la opción “Express”:

<https://www.microsoft.com/es-es/sql-server/sql-server-downloads>

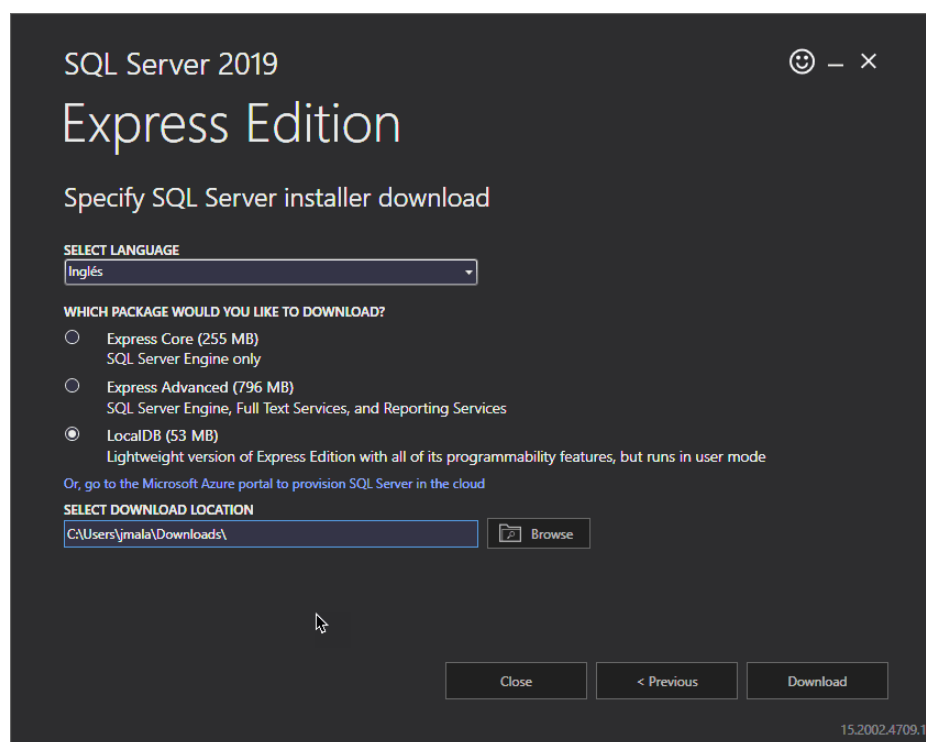
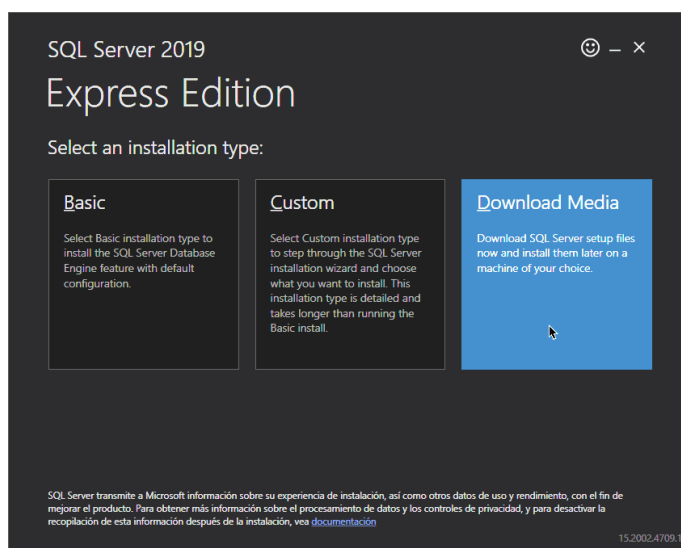


#### Express

SQL Server 2022 Express es una edición gratuita de SQL Server ideal para el desarrollo y la producción de aplicaciones de escritorio, aplicaciones web y pequeñas aplicaciones de servidor.

[Descargar ahora](#)

En la primera pantalla te ofrece tres opciones para hacer una instalación básica de SQL Server Express, una personalizada, y la tercera para descargar los archivos de instalación:



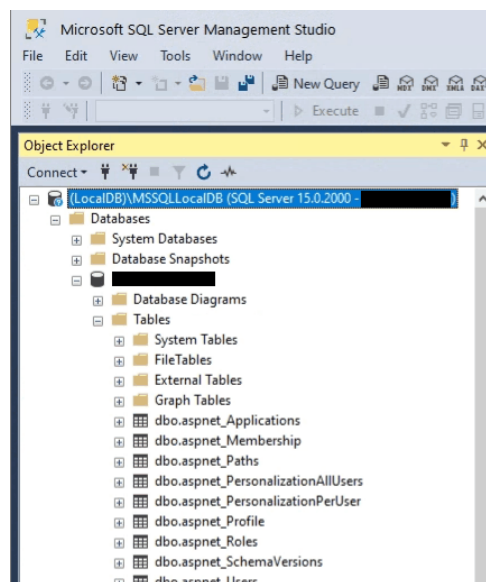
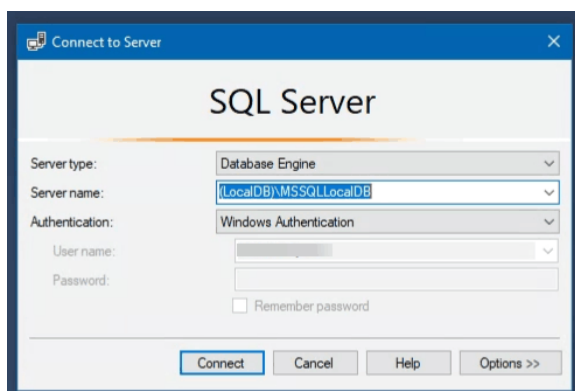
Seleccionamos la opción “LocalDB” y la instalamos.

## 2. Gestión del Servidor SQL: SQL Server Management Studio SSMS

Para conectarte a LocalDB normalmente lo harás desde un programa que use alguna base de datos, pero **para trabajar y gestionar las bases de datos y sus tablas puedes hacerlo desde [SQL Server Management Studio](#) (SSMS)** o desde cualquier programa que lo permita, como por ejemplo el propio Visual Studio.

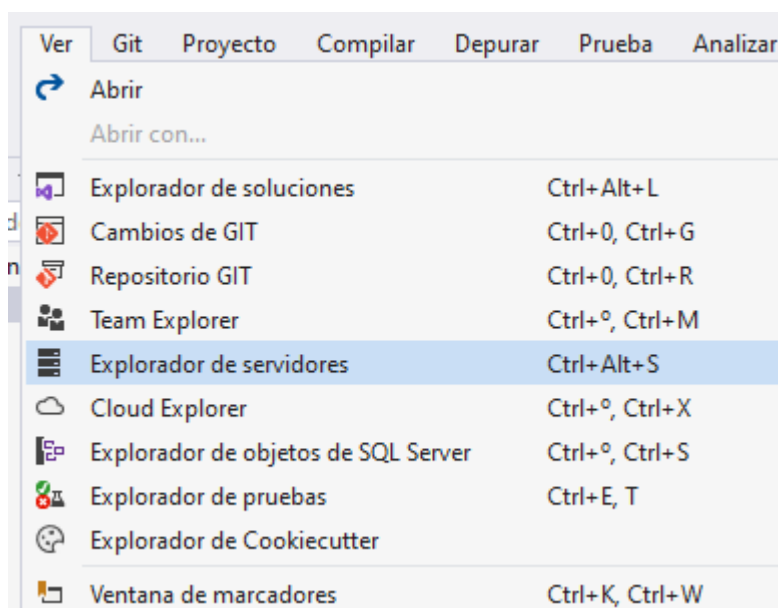
La cadena de conexión para conectarte a una instancia de LocalDB es igual a cualquiera que usarías para SQL Server, pero con un detalle importante: el nombre del servidor de datos. El nombre del servidor es siempre (LocalDB)\<NombreInstancia>

<https://aka.ms/ssmsfullsetup>

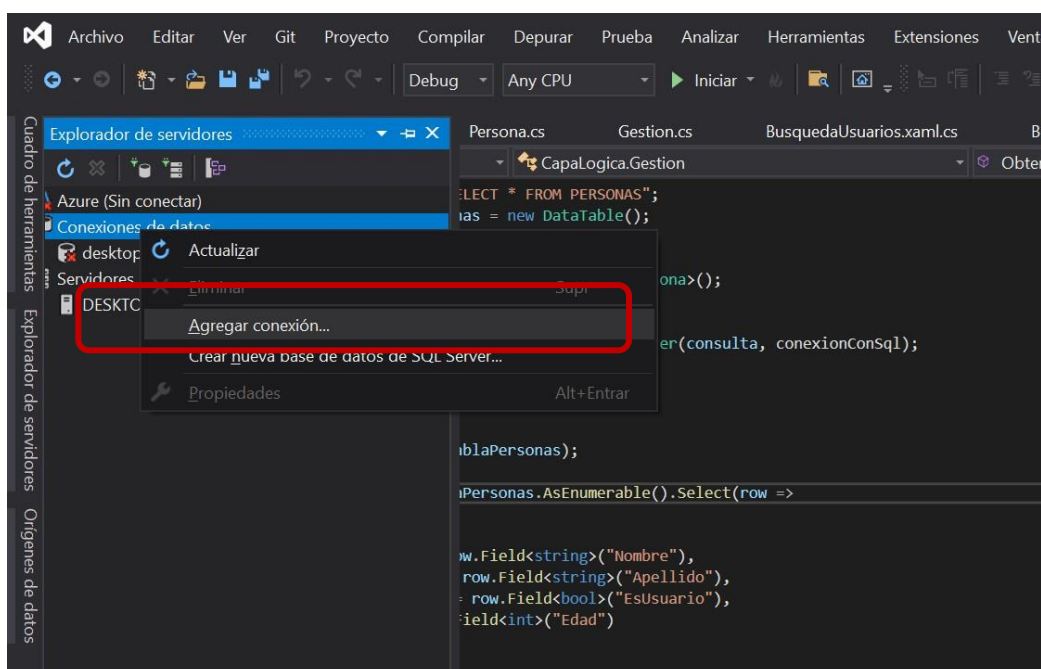


### 3. Conexión en Visual Studio de la Base de Datos

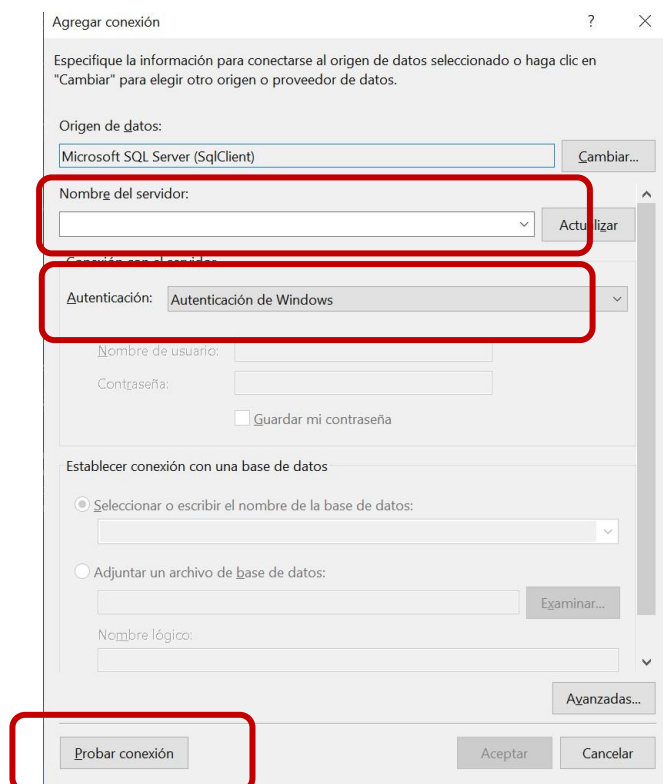
Abrir el “Explorador de Servidores” para conectarnos a nuestra Base de Datos:



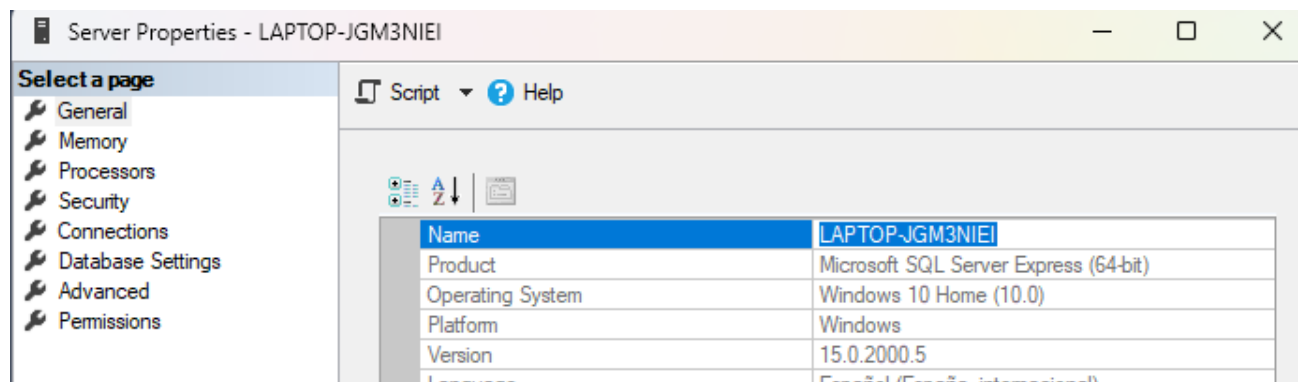
Y agregamos “Conexión al Servidor”:



Se nos despliega la ventana de “Agregar conexión”, donde tendremos que indicar el nombre del servidor, la autenticación, el nombre de la base de datos y probar la conexión:

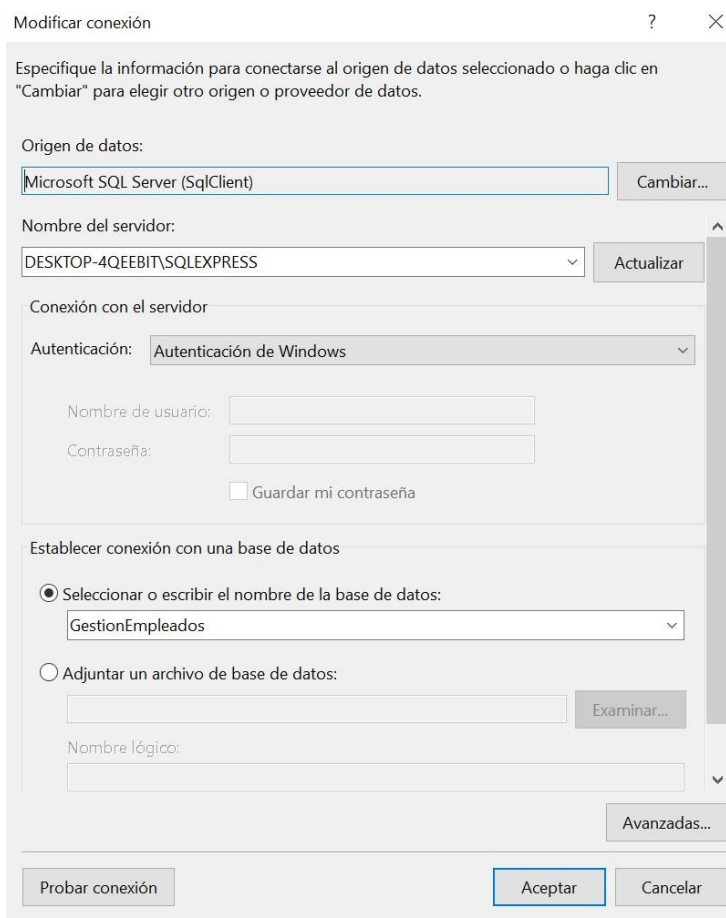


El nombre del Servidor que vamos a utilizar lo podemos ver en el gestor de Base de Datos SSMM y ahí dentro de propiedades podemos copiar el “Nombre”. Una opción para indicar el nombre del servidor es ir a SQL Server y seleccionar las propiedades del servidor (botón derecho sobre el nombre del servidor), para acceder al nombre.

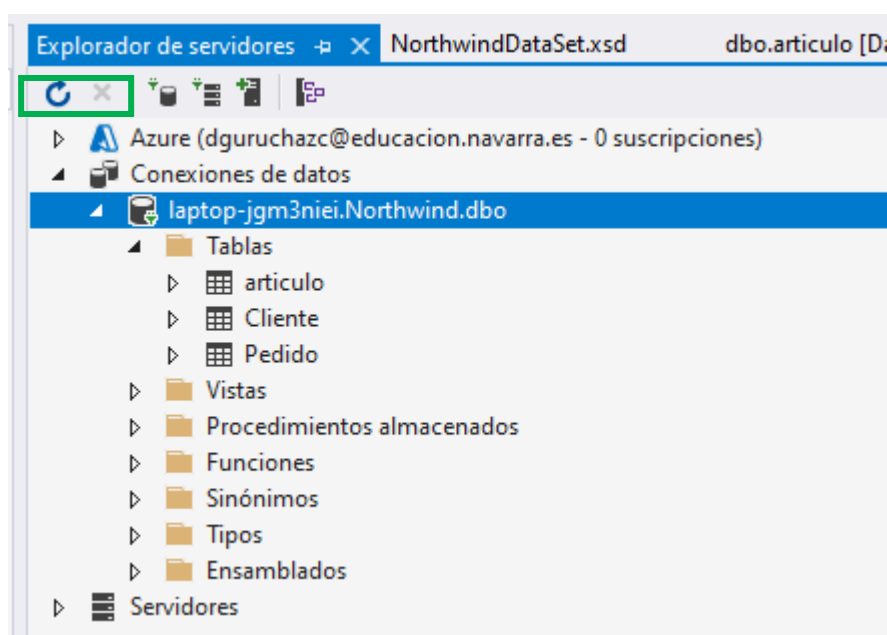


#### Nombre del Servidor SQL

Y añadimos el nombre de la Base de Datos, y si no está nos la creará. El resultado será similar al siguiente:

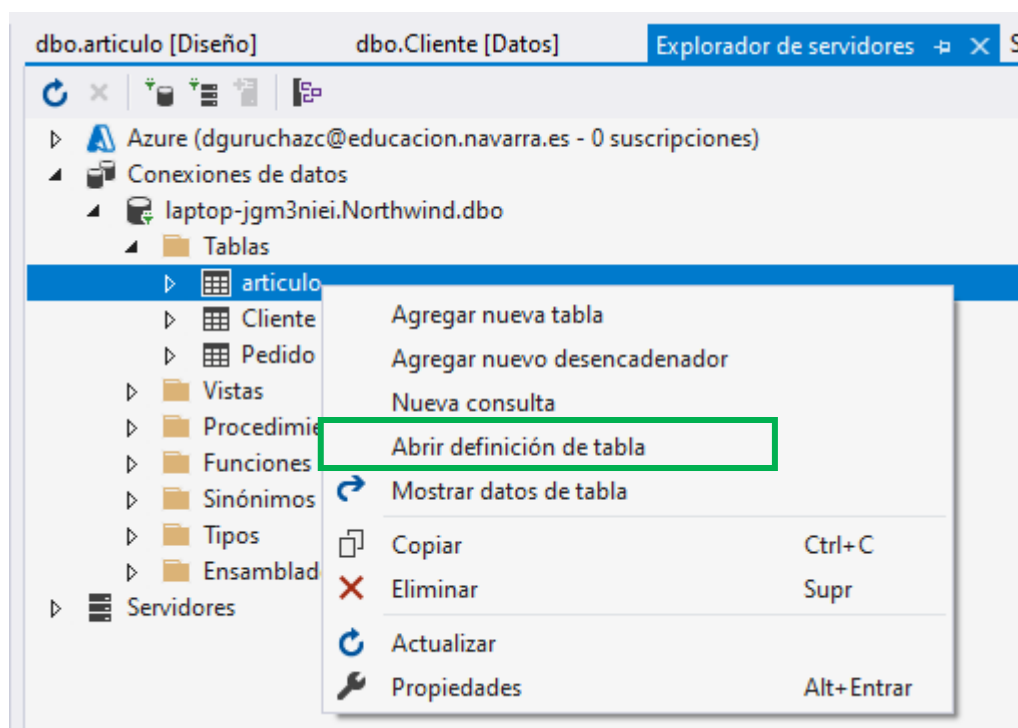


Y ya veríamos ver la Base de Datos en Visual Studio (con sus tablas si las hubiera, Vistas, Procedimientos, Funciones, ...):

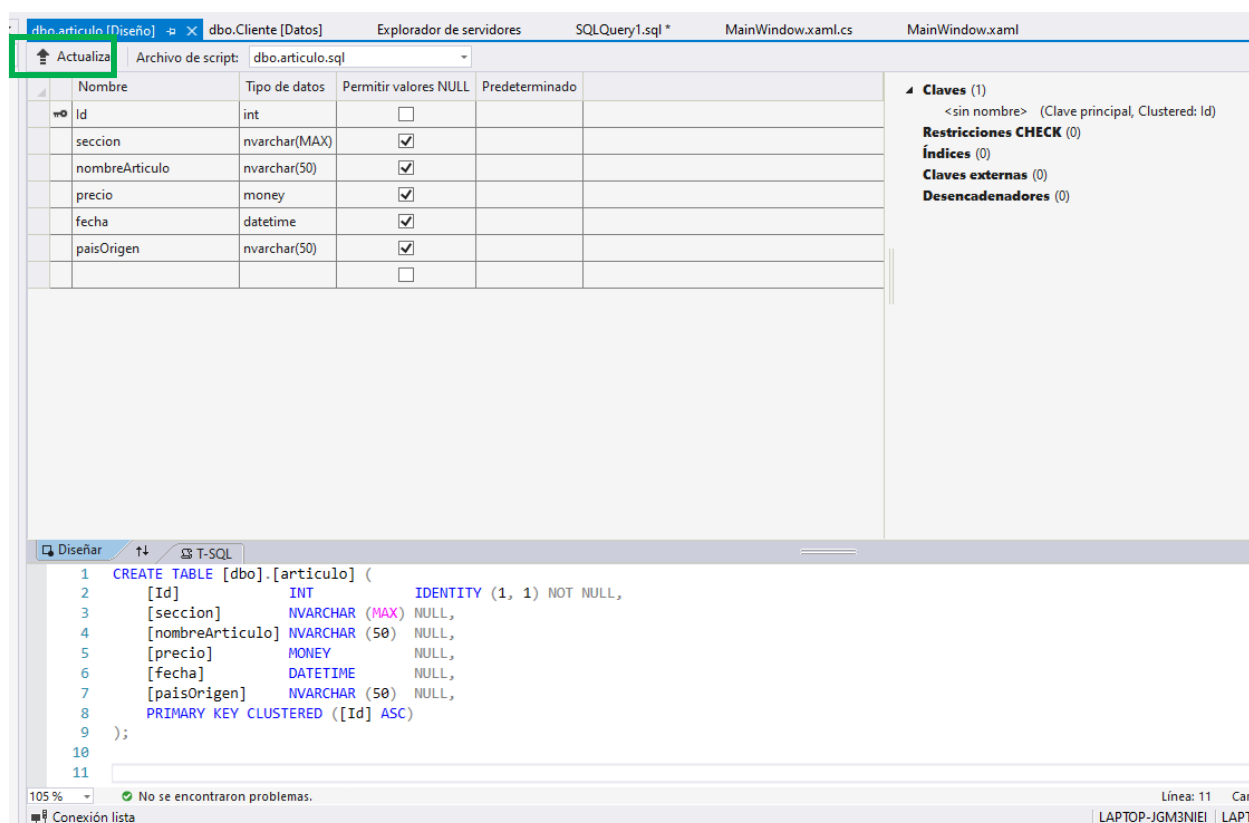


Vista de la Base de Datos. Importante “Actualizar”.

Y a partir de aquí podríamos crear las tablas que necesitemos, e ir modificándolas, definiéndolas, añadiendo datos, etc.:

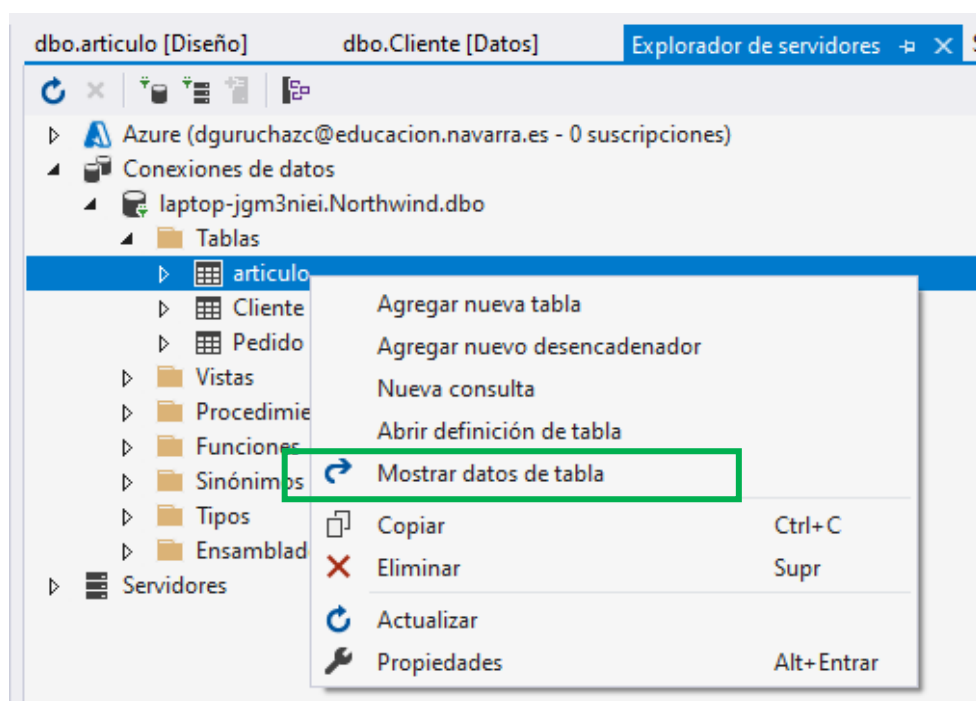


Abrimos definición de tabla para diseñar/modificar sus campos



Vista "Definición" de una tabla. Importante ir "Actualizando" todo con el Servidor.

Para ver, añadir o modificar los datos de las tablas, abriremos en la tabla implicada dentro del "Explorador de Servidores", la opción "Mostrar datos de tabla":



Seleccionamos "Mostrar datos de tabla"

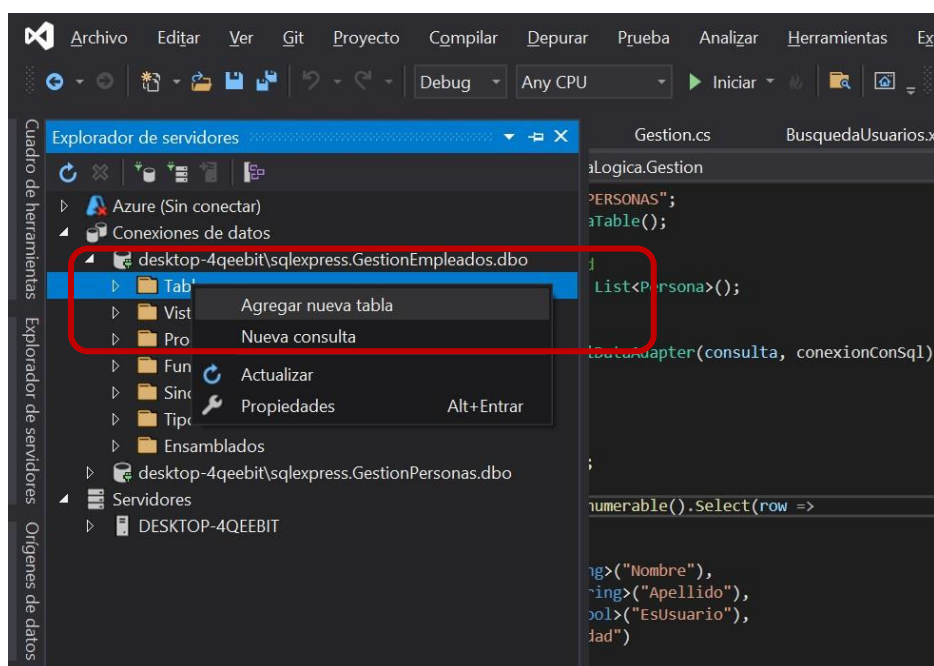


Y una vez en esa tabla, podemos ver y modificar los datos de ella:

Id	seccion	nombreArticulo	precio	fecha	paisOrigen
1	deportes	Zapatilla Trail	110,0000	04/05/2023 0:00...	Italia
2	ferretería	Sartén	25,0000	01/02/2023 0:00...	Francia
3	juguetería	Coche bomberos	45,0000	10/06/2023 0:00...	NULL
4	cerámica	Plato tiesto	15,0000	12/07/2023 0:00...	NULL
5	confección	Tela roja	11,0000	20/06/2023 0:00...	NULL
NULL	NULL	NULL	NULL	NULL	NULL

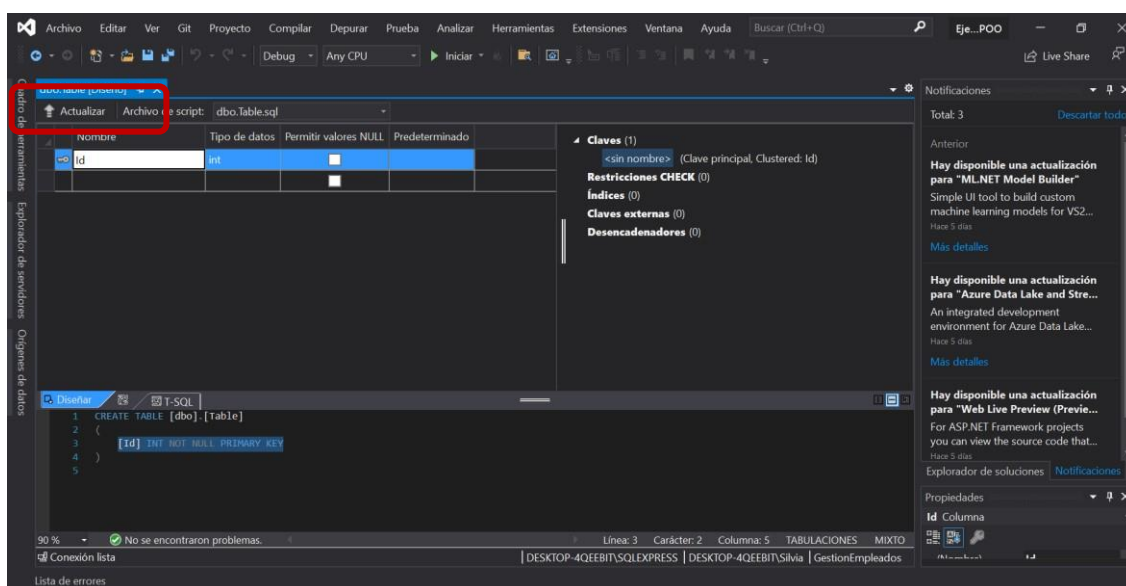
Datos de una tabla desde Visual Estudio

También podemos “Agregar una nueva Tabla” si lo necesitamos:



Se abrirá una pantalla donde podréis crear los campos de la nueva tabla, similar a:



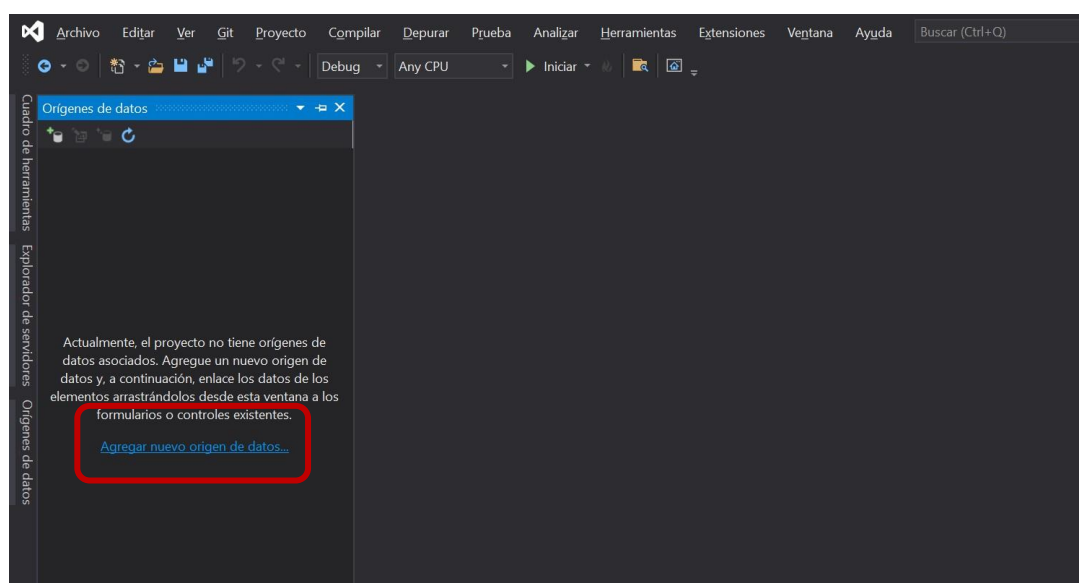


Para que se reflejen los cambios en SQL Server será necesario **actualizar** los datos.

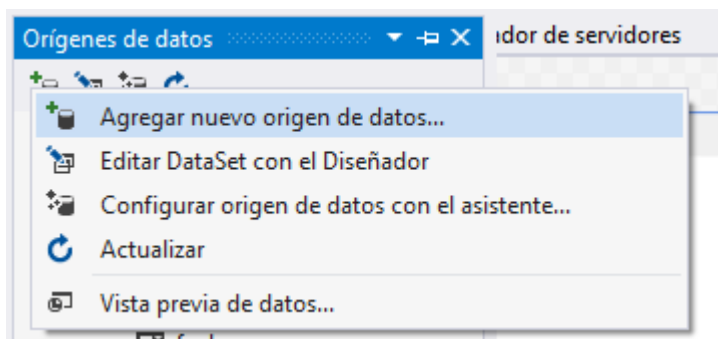
## 4. Utilización de una Base de Datos para un Proyecto WPF

### 4.1. Conectar la Base datos a nuestro proyecto

Para poder utilizar una BdD en una aplicación que vamos a desarrollar, lo primero es en la MainWindow, ir a la pestaña de la derecha “Orígenes de datos” y ahí “Agregar nuevo origen de datos” para poder acceder a la BdD que vamos a necesitar:

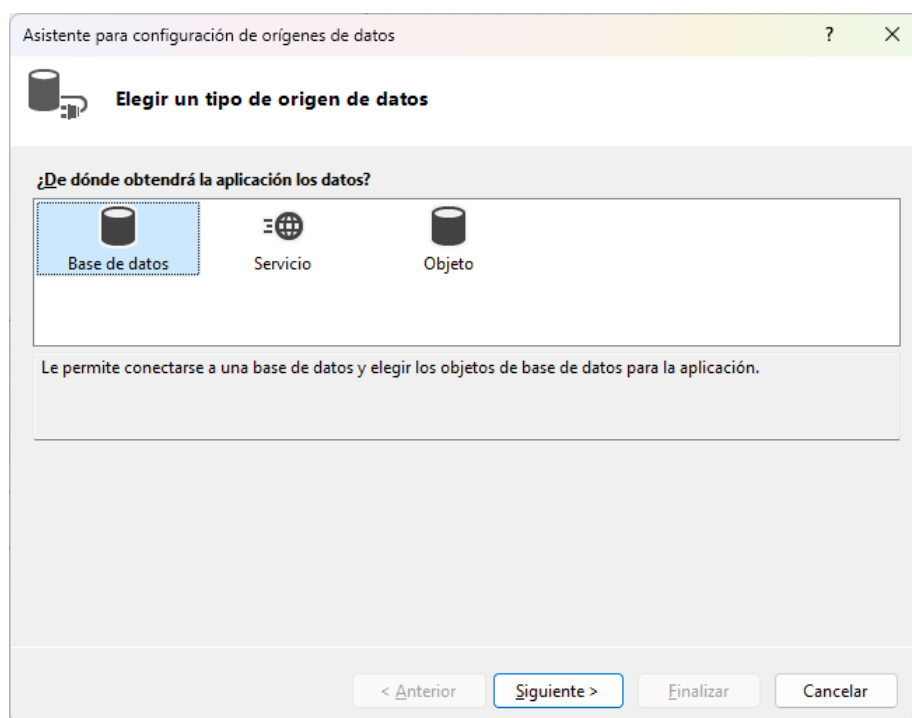


En el caso que no aparezca ese acceso directo o necesitamos crear un nuevo acceso a otra Base de Datos, seleccionaremos:



Agregamos “Nuevo origen de datos”

Y ahí vamos avanzando con el asistente:



Agregaremos una Base de Datos

En la ventana de “Elegir la conexión de datos”, crearemos la conexión a la BdD que tengamos creada, o también podemos conectarnos a más BdD que vayamos a necesitar (con “Nueva conexión”):

En el caso de que tenga usuario y contraseña, tendremos 2 opciones para seleccionar (no incluir o incluir los datos confidenciales (contraseña) en la cadena de conexión que pasará a nuestro proyecto:

Opciones para seleccionar en caso de que hayamos configurado la BdD con contraseña

Damos a siguiente e, Importante, copiar la “Cadena de Conexión” que luego utilizaremos para en el Diseño de la Base de Datos:

Asistente para configuración de orígenes de datos

**Guardar cadena de conexión en el archivo de configuración de la aplicación**

El almacenamiento de las cadenas de conexión del archivo de configuración de aplicación facilita el mantenimiento y la implementación. Para guardar la cadena de conexión en el archivo de configuración de la aplicación, escriba un nombre en el cuadro y, a continuación, haga clic en Siguiente.

¿Desea guardar la cadena de conexión en el archivo de configuración de la aplicación?

☒ Sí, guardar la conexión como:

< Anterior   Siguiente >   Finalizar   Cancelar

Cadena de Conexión de la BdD

En la siguiente ventana podemos seleccionar los **objetos** de la BdD que necesitamos (tablas, vistas, ...):

Asistente para configuración de orígenes de datos

**Elegir los objetos de base de datos**

¿Qué objetos de la base de datos desea tener en el conjunto de datos?

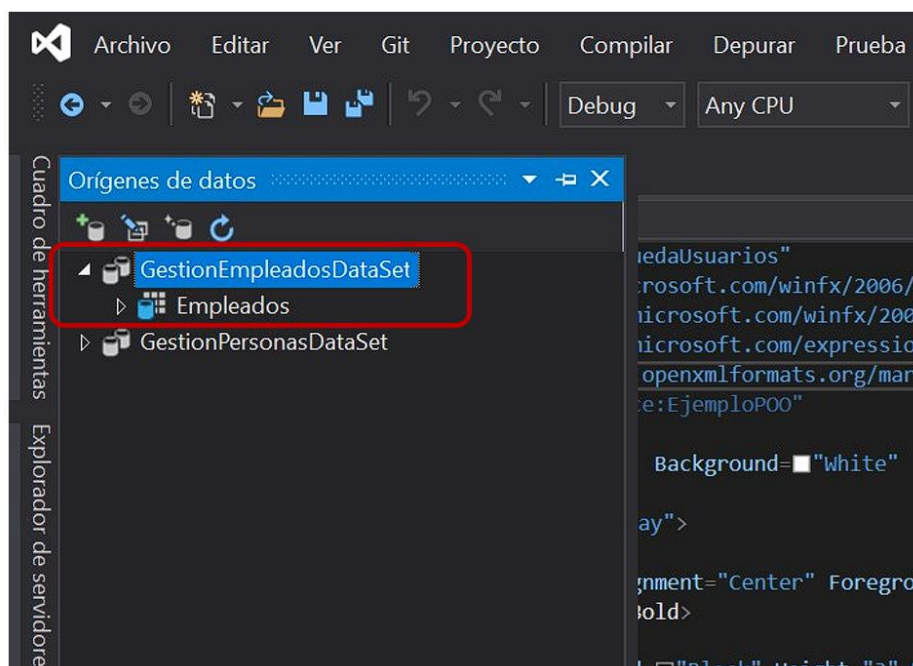
- ☒ Tablas
- ☒ Vistas
- ☒ Procedimientos almacenados
- ☒ Funciones

Nombre de DataSet:

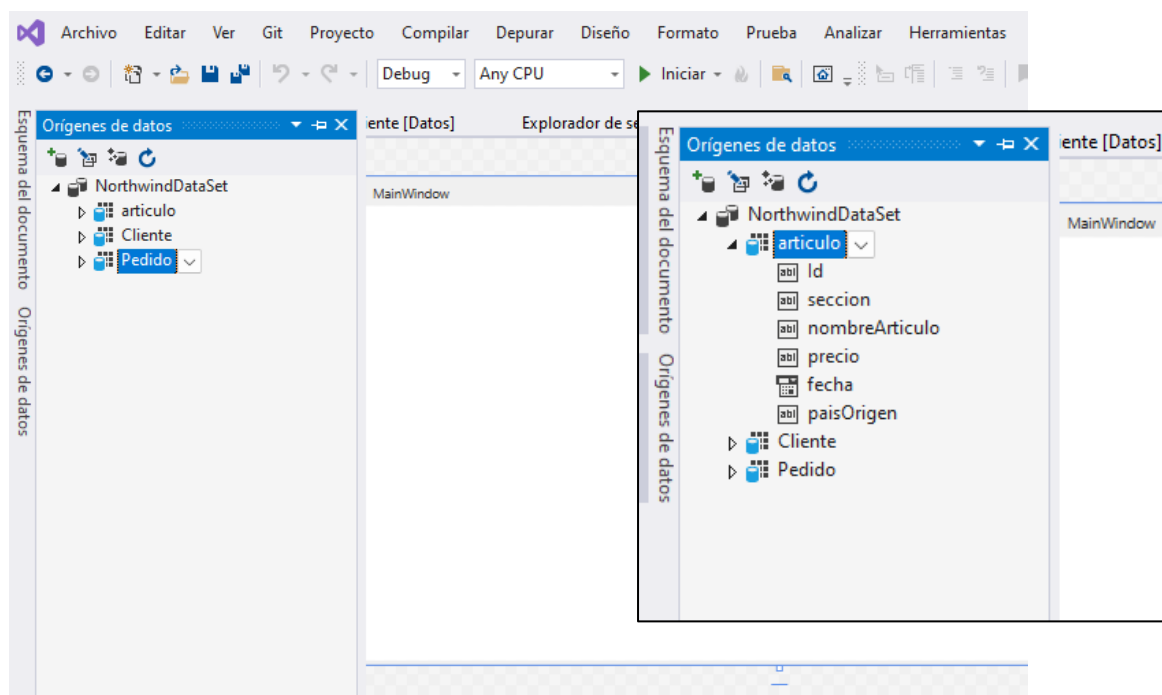
< Anterior   Siguiente >   Finalizar   Cancelar

Vista "Elegir los objetos de base de datos"

En este punto ya nos ha creado un **DataSet** con la información de los objetos de tipo tabla:

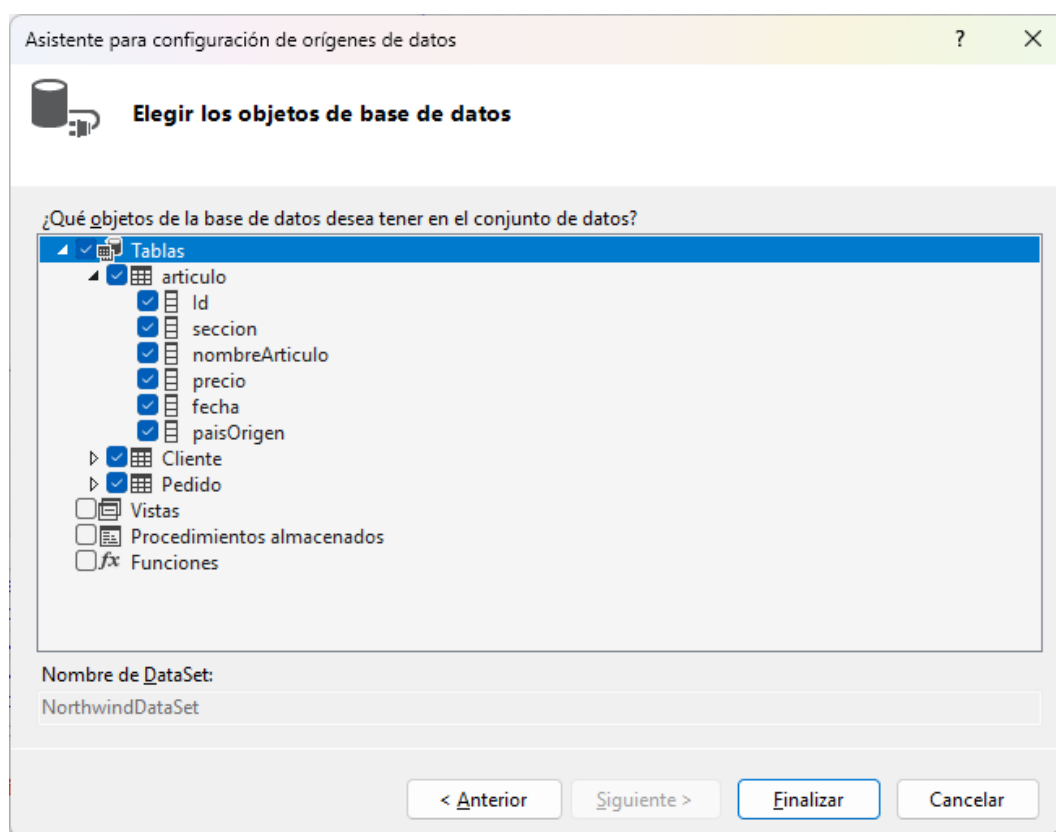
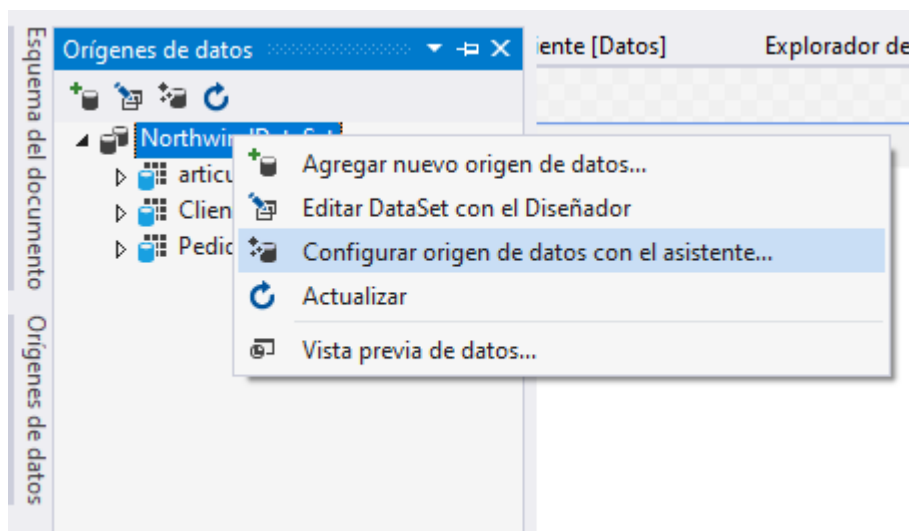


Una vez establecida la conexión ya vemos en el panel “Orígenes de datos” todo lo que hemos traído desde nuestra Base de datos de SQL:



Las tablas del DataSet de ejemplo “NortwindDataSet” y los elementos de la tabla “artículos”

Si hay algún dato que hemos modificado y **no aparece**, siempre podremos ir a “Configurar origen de datos con el asistente...” y ahí añadir más elementos que necesitamos:



Seleccionar o quitar elementos de la BdD

#### 4.2. Creación de la **cadena de conexión** para las Window (diseño gráfico)

Ahora necesitamos conectar la Aplicación Gráfica (Ventanas) con este “Origen de Datos” y para ello vamos a crear una conexión desde la ventana de código c# de la MainWindow donde añadiremos la siguiente sentencia:

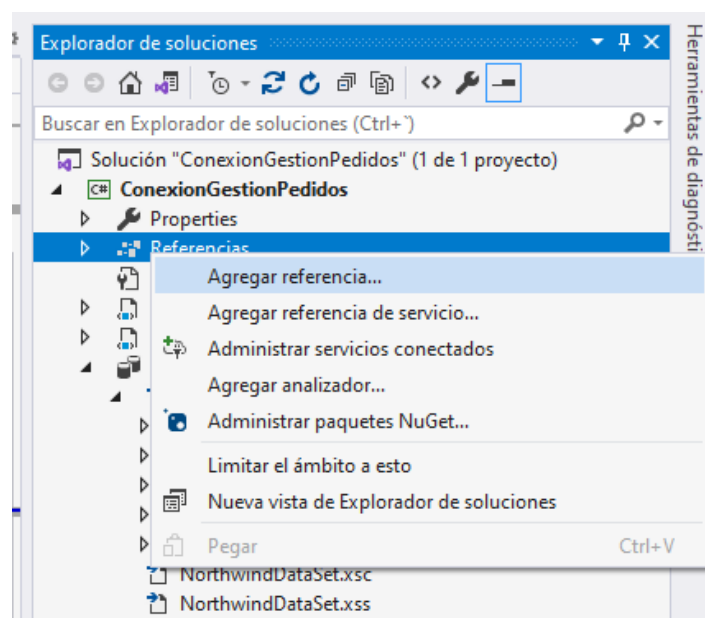
```
dbo.articulo [Datos]    dbo.articulo [Diseño]    dbo.Cliente [Datos]    Explorador de servidores    SQLQuery1.sql *    MainWindow.xaml.cs *    MainWindow.xaml
ConexionGestionPedidos
10  using System.Windows.Input;
11  using System.Windows.Media;
12  using System.Windows.Media.Imaging;
13  using System.Windows.Navigation;
14  using System.Windows.Shapes;
15  using System.Configuration;
16
17  namespace ConexionGestionPedidos
18  {
19      /// <summary>
20      /// Lógica de interacción para MainWindow.xaml
21      /// </summary>
22      2 referencias | 0 cambios | 0 autores, 0 cambios
23      public partial class MainWindow : Window
24      {
25          0 referencias | 0 cambios | 0 autores, 0 cambios
26          public MainWindow()
27          {
28              InitializeComponent();
29              string miConexion = ConfigurationManager.ConnectionStrings["ConexionGestionPedidos.Properties.Settings.NorthwindConnectionString"].ConnectionString;
30          }
31      }
32
```

MainWindow.xaml.cs

El código es:

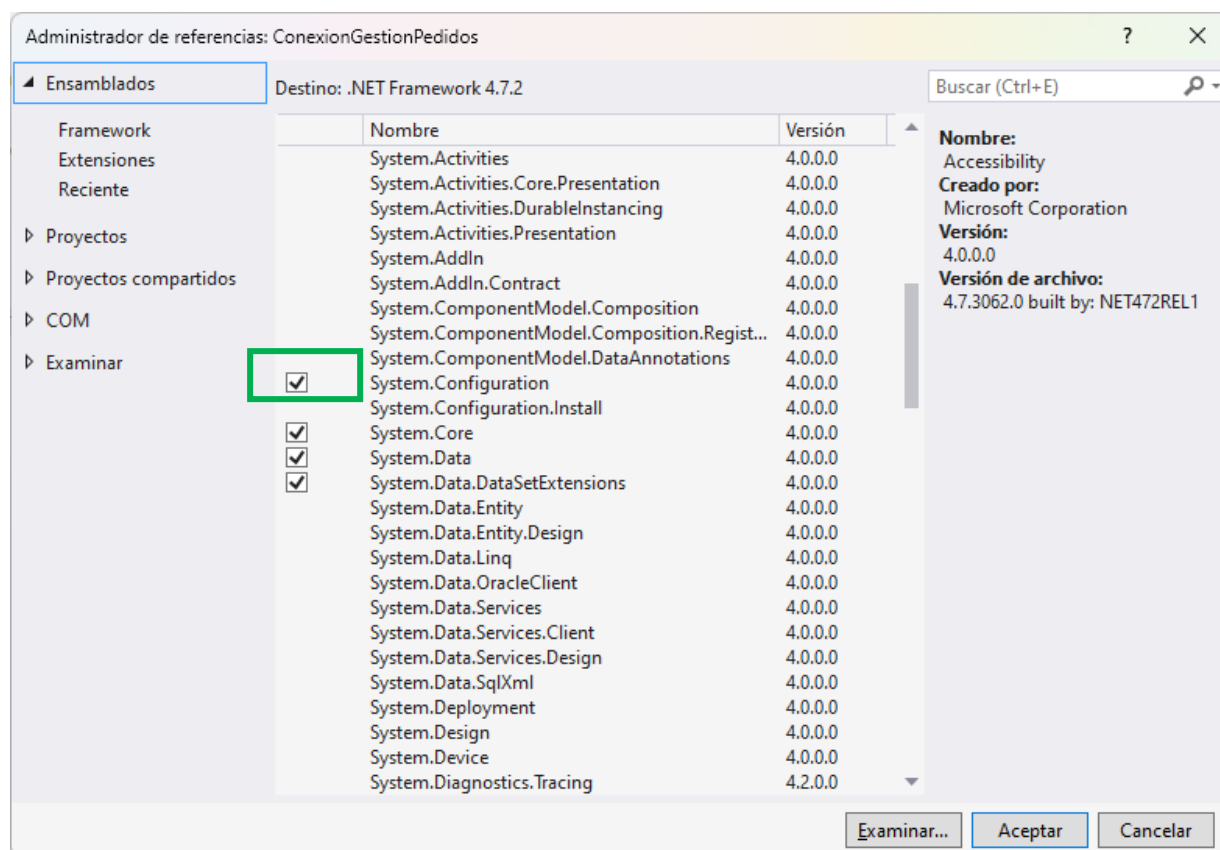
```
string miConexion =
ConfigurationManager.ConnectionStrings["ConexionGestionPedidos.Properties.Settings.NorthwindC
onnectionString"].ConnectionString;
```

Nos dará **error** si no tenemos configurado “ConnectionStrings”, y para solucionarlo tenemos que ir a “Referencias” en el “Explorador de Soluciones” y ahí “Agregar referencia”:



Y en el “Administrador de referencias” seleccionar “System.Configuration”.





Después añadimos “`using System.Configuration;`” en la ventana de nuestro proyecto donde estamos añadiendo la conexión:

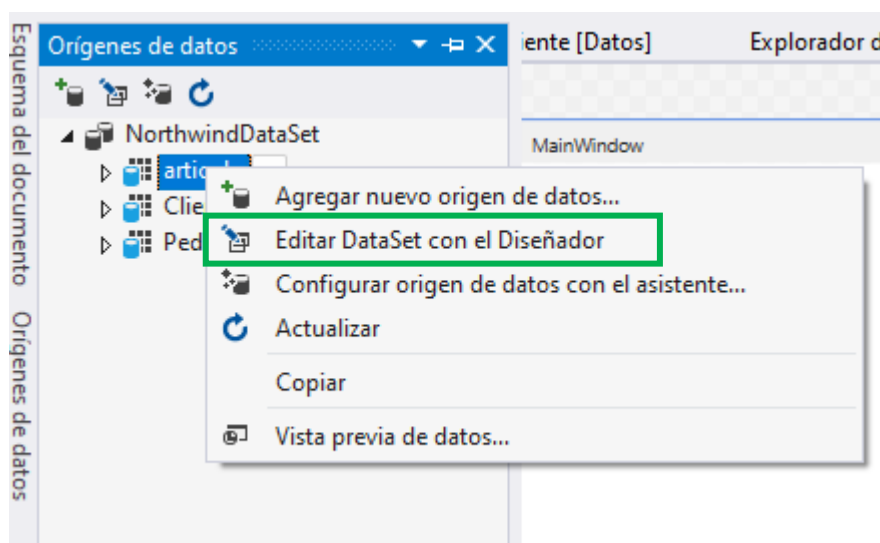
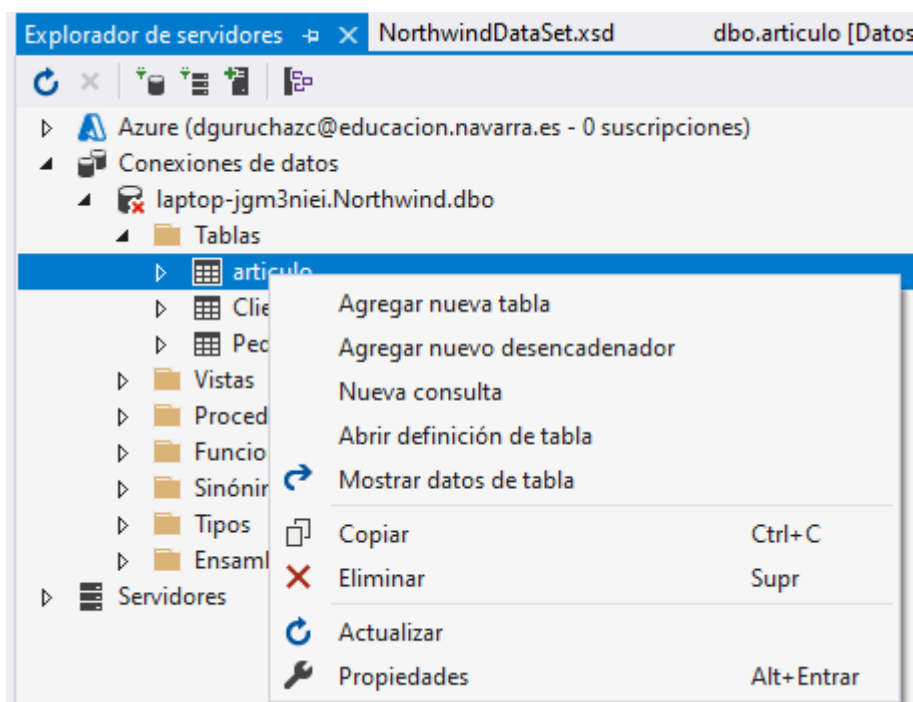
```
12 | using System.Windows.Media.Imaging;
13 | using System.Windows.Navigation;
14 | using System.Windows.Shapes;
15 | using System.Configuration;
16 |
17 | namespace ConexionGestionPedidos
18 | {
19 |     /// <summary>
20 |     /// Lógica de interacción para MainWindow.xaml
21 |     /// </summary>
22 |     2 referencias | 0 cambios | 0 autores, 0 cambios
23 |     public partial class MainWindow : Window
24 |     {
25 |         0 referencias | 0 cambios | 0 autores, 0 cambios
26 |         public MainWindow()
27 |         {
28 |         }
29 |     }
30 | }
```

MainWindow.xaml.cs

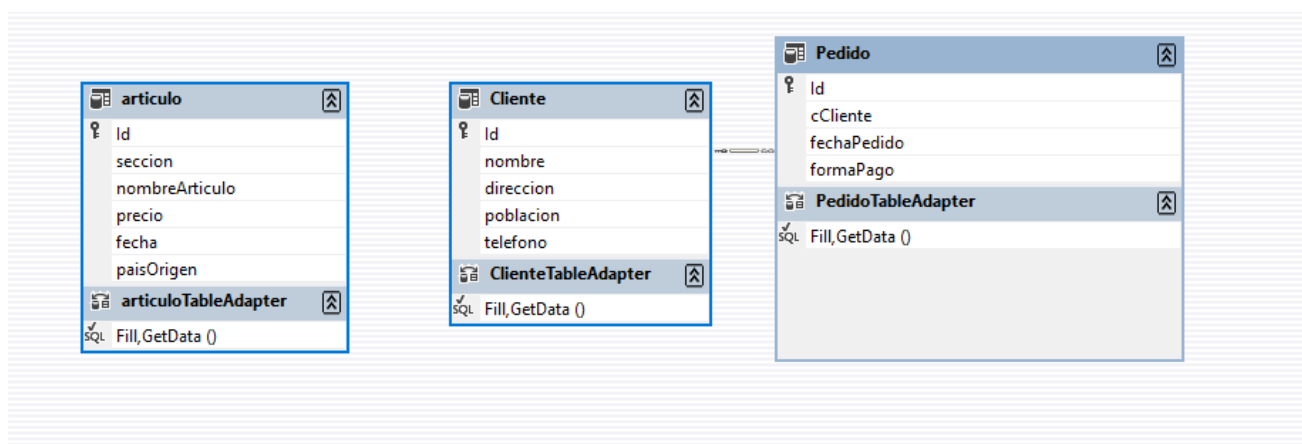
Edición de la Base de datos desde nuestra base de Datos.

Desde la ventana de “Explorador de Servidores” podremos seleccionar la tabla que necesitemos y con el botón derecho del ratón utilizaremos las opciones:

- Agregar nueva tabla
- Agregar Nueva consulta
- Abrir definición de tabla → Definir o modificar campos
- Mostrar datos de tabla → Ver, modificar añadir, borrar datos de una tabla



También desde “Orígenes de datos” seleccionamos “Editar DataSet con el Diseñador” para ver las “Relaciones” entre las tablas:



Vista del “Diseñador” con las tablas y sus relaciones

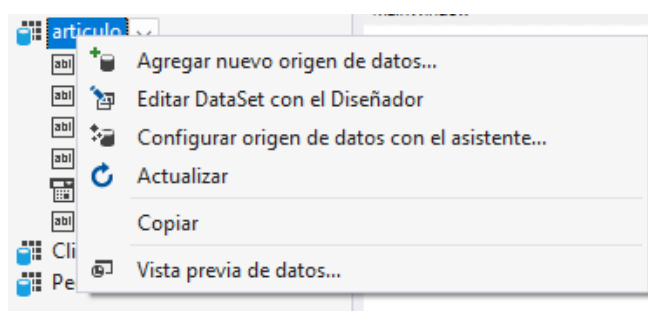
#### 4.3. Ver los datos de una tabla

Tendríamos 2 maneras de ver los datos dentro de una tabla:

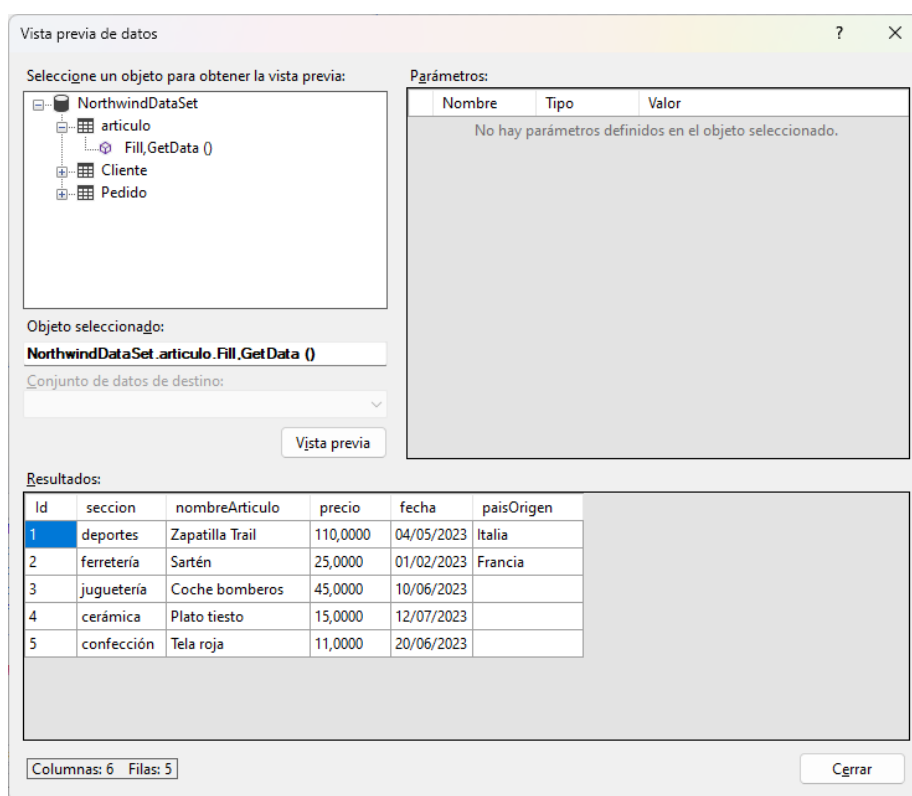
- 1) Desde el “Explorador de servidores” con la opción “Mostrar datos de tabla” al seleccionar una tabla:

	Id	seccion	nombreArticulo	precio	fecha	paisOrigen
▶	1	deportes	Zapatilla Trail	110,0000	04/05/2023 0:00...	Italia
	2	ferretería	Sartén	25,0000	01/02/2023 0:00...	Francia
	3	juguetería	Coche bomberos	45,0000	10/06/2023 0:00...	NULL
	4	cerámica	Plato tiesto	15,0000	12/07/2023 0:00...	NULL
	5	confección	Tela roja	11,0000	20/06/2023 0:00...	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

- 2) Desde la pestaña “Origen de datos” con la opción “Vista previa de los datos” al seleccionar una tabla.



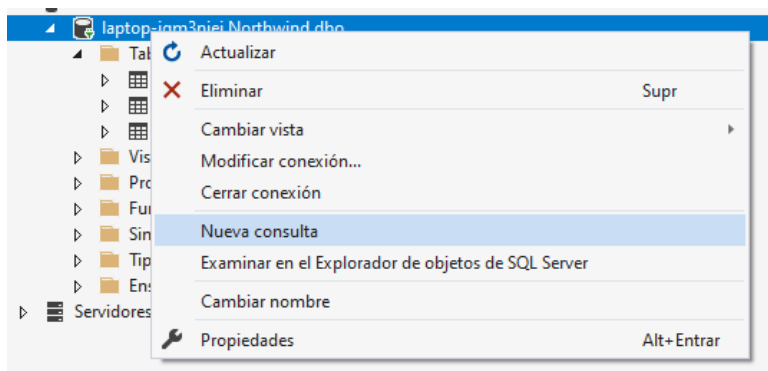
Opciones al seleccionar una tabla



Vista previa de datos de una tabla

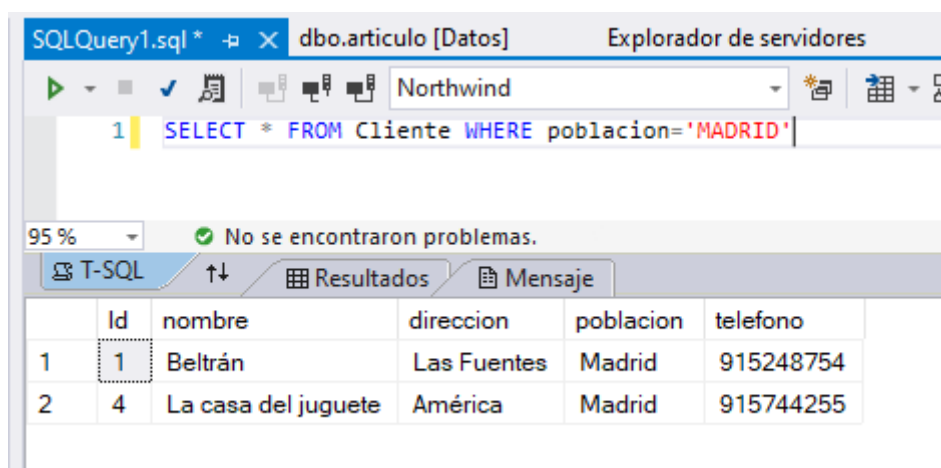
#### 4.4. Hacer Consultas tipo SQL sobre las tablas

Sobre la Base de Datos, dentro del “Explorador de Servidores”, seleccionamos “Nueva consulta”:



Y ya utilizando el lenguaje SQL sobre la ventana “SQLQuery.sql” que se ha abierto, hacemos por ejemplo una consulta sobre nuestras tablas:

```
SELECT * FROM Cliente WHERE poblacion='MADRID'
```



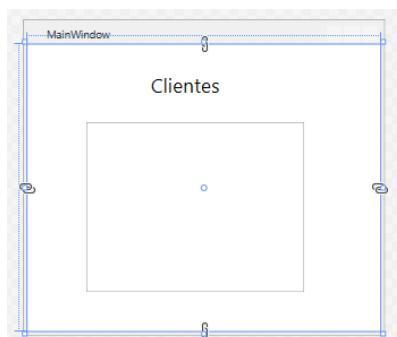
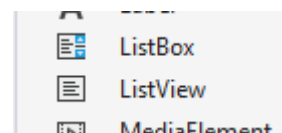
Ventana SQLQuery1.sql con una consulta

#### 4.4. Ver los datos de una tabla desde la MainWindow.xaml

##### 4.4.1. Diseño MainWindow.xaml

Para ver datos de una tabla desde una ventana, primero deberemos crear en el entorno gráfico una lista desde “ListBox” o “ListView”, controles que están en el “Cuadro de herramientas”.

También tenemos que dar un nombre a esa lista (por ejmplo `<ListBox Name="listaClientes" ...>`)



Ejemplo de Label con ListBox

##### 4.4.2. Configuración de la MainWindow.xaml.cs

a) Después tenemos que ir a la ventana MainWindow.xaml.cs y conectar con nuestro “origen de datos” y lo primero que necesitamos será crear desde la Clase Principal, un Objeto perteneciente a la **Clase “SqlConnection”**:

<https://learn.microsoft.com/es-es/dotnet/api/system.data.sqlclient.sqlconnection?view=dotnet-plat-ext-7.0>

Nota: Hay que añadir la sentencia “using System.Data.SqlClient;” en las líneas de importación:

```
14 using System.Windows.Shapes;  
15 using System.Configuration;  
16 using System.Data.SqlClient;
```

Y declaramos un campo de esta clase, para crear el Objeto “miConexionSql” :

```
SqlConnection miConexionSql;
```

```
public partial class MainWindow : Window
{
    SqlConnection miConexionSql;

    0 referencias | 0 cambios | 0 autores, 0 cambios
    public MainWindow()
    {
        InitializeComponent();
    }
}
```

Y ahora dentro del método de MainWindow, tenemos que instanciar la clase **SqlConnection** a través del Objeto que acabamos de crear:

```
miConexionSql = new SqlConnection(miConexion);
```

```
0 referencias | 0 cambios | 0 autores, 0 cambios
public MainWindow()
{
    InitializeComponent();

    string miConexion = ConfigurationManager.ConnectionStrings["miConexion"].ConnectionString;

    miConexionSql = new SqlConnection(miConexion);
}
```

Y ya estaríamos conectados.

**b)** Ahora tenemos que EXTRAER los datos que necesitamos. Para ello tenemos que crear dentro de la **clase** pero fuera del MainWindow, un método para extraer esa información (si quisiéramos utilizarlo desde otras Clases debería ser “**public**” y si no “**private**”).

Y después añadiremos la instrucción “string **consulta** = “SELECT \* FROM **CLIENTE**”; para que nos envíe la información de la **Tabla** “**Cliente**” y que se almacene en “**consulta**”

```
private void MuestraClientes()
{
    string consulta = "SELECT * FROM CLIENTE";
}
```

**c)** Para poder mostrar los datos tenemos que utilizar la clase “**SqlDataAdapter**”:

<https://learn.microsoft.com/es-es/dotnet/api/system.data.sqlclient.sqldataadapter?view=dotnet-plat-ext-7.0>

Esta clase representa un conjunto de comandos de datos y una conexión a una base de datos que se usan para rellenar [DataSet](#) o [DataTable](#) y actualizar una base de datos de SQL Server. Sirve como punto intermedio donde se “adaptan” los datos para que puedan aparecer en nuestra lista.

Introduciríamos la siguiente sentencia:

```
SqlDataAdapter miAdaptadorSql = new SqlDataAdapter(consulta, miConexionSql);  
  
using (miAdaptadorSql)
```

d) Y una vez añadido esto, introduciríamos el código para requerir el campo o campos de la tabla que necesitamos:

```
{  
  
    DataTable clientesTabla = new DataTable();  
  
    miAdaptadorSql.Fill(clientesTabla);  
  
    listaClientes.DisplayMemberPath = "nombre";  
    listaClientes.SelectedValuePath = "Id";  
    listaClientes.ItemsSource = clientesTabla.DefaultView;  
  
}
```

En este ejemplo, nos mostrará:

1. Los datos del campo “nombre”.
2. Adicionalmente hay que añadir el campo ID de esa tabla y
3. Hay que añadir el origen de los datos de esa lista.

e) Y ya creado el **método** ahora tenemos que ejecutarlo y para eso los tenemos que “llamar”, por lo que hay que añadirlo después de la “Inicialización de los componentes” de que “establezca la conexión con la base de datos” y de que establezca la conexión de tipo SQL. Por lo que después de todo eso agregamos:

```
miConexionSql = new SqlConnection(miConexion);
```

```
MuestraClientes();
```



0 referencias | 0 cambios | 0 autores, 0 cambios

```
public MainWindow()  
{  
    InitializeComponent();  
  
    string miConexion = ConfigurationManager.ConnectionStrings  
  
    miConexionSql = new SqlConnection(miConexion);  
  
    MuestraClientes();  
}
```

Y una vez hecho todo eso, ejecutaríamos la aplicación y veríamos los resultados buscados:

