

Desarrollo de Software IV

Laboratorio # 13 – Prof. Regis Rivera

Objetivo: Integración de base de datos SQL con C#

PARAMETROS

- **Parte I**
 - Cliente: SQL Server Management Studio
 - Servidor: (local) *[Nombre de su equipo con la instancia de SQL Server]*
 - Autenticación: Windows
 - Base de datos: Nortwind
- **Parte II**
 - IDE: Visual Studio .NET 2022
 - Tipo de Aplicación: Windows Forms
 - Solución: Laboratorio13
 - Proyectos: Laboratorio131, 132... *[Según cada problema propuesto]*
 - Lenguaje: C#

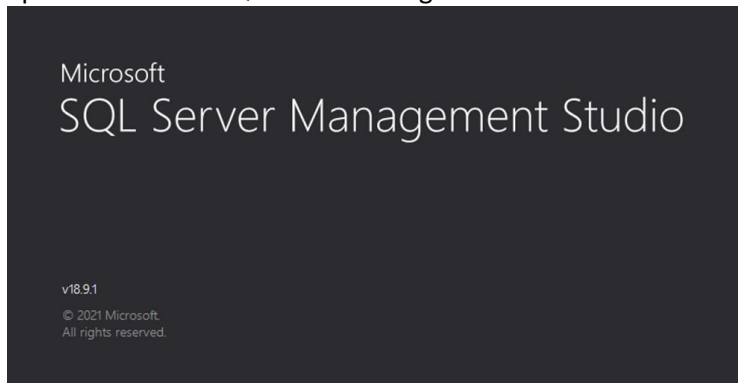
PARTE 1 – Repaso de consultas SQL con la base de datos Norwthwind

1. Ingresar a SQL Server Management Studio

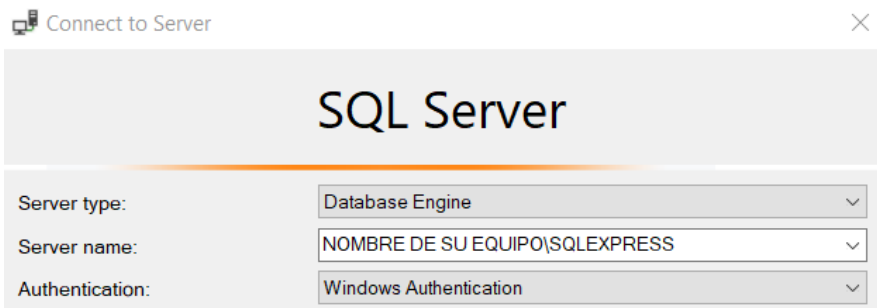


SQL Server Management Studio
Aplicación

2. Si no encuentra esta aplicación, requiere instalar SQL Server (se recomienda la versión Express, al ser más ligera y gratuita)
 - URL para descargar SQL Server Express:
<https://www.microsoft.com/es-es/download/details.aspx?id=101064>
 - Pasos para instalar:
<https://factory.com.co/wp-content/uploads/2022/01/Configuracion-SQL-Server-2019-Express.pdf>
- URL para descargar MS SQL Server Management Studio
<https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>
3. Al ingresar, entrar con autenticación Windows
 - Splash Screen de SQL Server Management Studio



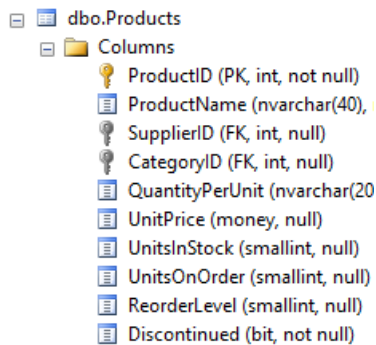
- Login, iniciar con autenticación windows



- Al ingresar, validar la existencia de la base de datos Northwind
 - En el panel izquierdo, confirmar si posee la base de datos Northwind
 - [-] Databases
 - [+] System Databases
 - [+] Database Snapshots
 - [+] Northwind
 - De no poseer la misma, ejecutar el script .SQL adjunto al laboratorio, el cual crea la base de datos Northwind, sus tablas y datos
 - instnwnd.sql
- Una vez tenemos nuestro ambiente de trabajo listo, veamos los siguientes ejemplos a ejecutar para recordar las consultas (queries) que podemos hacer, en este caso, tipo DML

SELECT

Ejemplo 1:



-- El asterisco significa que se quiere seleccionar todos los campos de la
-- tabla Products

```
SELECT * FROM Products
```

Resultado:

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
1	1	Chai	1	1	10 boxes x 20 bags	18.00	39	0	10	0
2	2	Chang	1	1	24 - 12 oz bottles	19.00	17	40	25	0
3	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10.00	13	70	25	0
4	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22.00	53	0	0	0
5	5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35	0	0	0	1
6	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25.00	120	0	25	0

Ejemplo 2:

-- Para seleccionar ciertos campos hay que colocarlos como en un listado
 -- en este ejemplo solo se mostraran los registros almacenados en los campos ProductID
 -- ProductName y UnitPrice

```
SELECT ProductID, ProductName, UnitPrice FROM Products
```

Resultado:

	ProductID	ProductName	UnitPrice
1	1	Chai	18.00
2	2	Chang	19.00
3	3	Aniseed Syrup	10.00
4	4	Chef Anton's Cajun Seasoning	22.00
5	5	Chef Anton's Gumbo Mix	21.35
6	6	Grandma's Boysenberry Spread	25.00
7	7	Uncle Bob's Organic Dried Pears	30.00
8	8	Northwoods Cranberry Sauce	40.00

WHERE

Aplica cualquier condición válida o combinación de condiciones utilizando operadores lógicos (NOT, AND, OR) y relacionales (=, <, >, <=, >=)

Ejemplo 3:

-- Seleccionar los datos de la tabla Products donde el dato almacenado en el campo
 -- UnitPrice sea mayor a 15

```
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE UnitPrice > 15
```

Resultado:

	ProductID	ProductName	UnitPrice
1	1	Chai	18.00
2	2	Chang	19.00
3	4	Chef Anton's Cajun Seasoning	22.00
4	5	Chef Anton's Gumbo Mix	21.35
5	6	Grandma's Boysenberry Spread	25.00
6	7	Uncle Bob's Organic Dried Pears	30.00
7	8	Northwoods Cranberry Sauce	40.00
8	9	Mishi Kobe Niku	97.00

Ejemplo 4:

```
-- Seleccionar los datos de la tabla Products donde el dato almacenado en el campo
-- UnitPrice sea mayor o igual a 15 y menor o igual a 50
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE UnitPrice >= 15 AND UnitPrice <= 50
```

Resultado:

	ProductID	ProductName	UnitPrice
1	1	Chai	18.00
2	2	Chang	19.00
3	4	Chef Anton's Cajun Seasoning	22.00
4	5	Chef Anton's Gumbo Mix	21.35
5	6	Grandma's Boysenberry Spread	25.00
6	7	Uncle Bob's Organic Dried Pears	30.00
7	8	Northwoods Cranberry Sauce	40.00
8	10	Ikura	31.00

Ejemplo 5:

```
--Otra forma de crear la consulta anterior es utilizando la instruccion BETWEEN
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE UnitPrice BETWEEN 15 AND 50
```

Ejemplo 6:

```
--Haciendo uso del operador NOT obtenemos los registros de la tabla Products donde
-- el dato almacenado en el campo UnitPrice sea menor que 15
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE NOT UnitPrice > 15
```

Resultado:

	ProductID	ProductName	UnitPrice
1	3	Aniseed Syrup	10.00
2	13	Konbu	6.00
3	19	Teatime Chocolate Biscuits	9.20
4	21	Sir Rodney's Scones	10.00
5	23	Tunnbröd	9.00
6	24	Guaraná Fantástica	4.50
7	25	NuNuCa Nuß-Nougat-Creme	14.00
8	31	Gorgonzola Telino	12.50
9	33	Geitost	2.50

Ejemplo 7

```
-- Seleccionar los registros de la tabla Products donde el dato almacenado en el campo
--ProductID sea mayor a 50 y el dato almacenado en el campo UnitPrice sea menor a 10
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE ProductID > 15 OR UnitPrice < 10
```

Resultado:

	ProductID	ProductName	UnitPrice
1	13	Konbu	6.00
2	16	Pavlova	17.45
3	17	Alice Mutton	39.00
4	18	Camarvon Tigers	62.50
5	19	Teatime Chocolate Biscuits	9.20
6	20	Sir Rodney's Marmalade	81.00
7	21	Sir Rodney's Scones	10.00
8	22	Gustaf's Knäckebröd	21.00
9	23	Tunnbröd	9.00

LIKE

La cláusula LIKE es útil para formar patrones de comparación con caracteres comodín.

Los comodines más utilizados son:

- `_`: Cualquier carácter.
- `%`: Cualquier cadena de cero o más caracteres
- `[]`: Cualquier carácter individual del intervalo ([a-z])

Puede utilizarse únicamente LIKE o NOT LIKE, indicando igual o no igual respectivamente

Ejemplo 8:

```
-- Seleccionar los campos EmployeeID y LastName de la tabla Employees donde el dato almacenado
-- en el campo LastName comience con la letra D
SELECT EmployeeID, LastName FROM Employees
WHERE LastName LIKE 'D%'
```

Resultado:

	EmployeeID	LastName
1	1	Davolio
2	9	Dodsworth

Ejemplo 9:

```
-- Seleccionar los campos EmployeeID y LastName de la tabla Employees donde el dato almacenado
-- en el campo LastName termine con la letra N
SELECT EmployeeID, LastName FROM Employees
WHERE LastName LIKE '%N'
```

Resultado:

	EmployeeID	LastName
1	5	Buchanan
2	8	Callahan

Ejemplo 10:

```
-- Seleccionar los campos EmployeeID, LastName y Title de la tabla Employees donde el dato
-- almacenado en el campo Title se encuentre la palabra SALES, no importando en que posición
SELECT EmployeeID, LastName, Title FROM Employees
WHERE Title LIKE '%SALES%'
```

Resultado:

	EmployeeID	LastName	Title
1	1	Davolio	Sales Representative
2	2	Fuller	Vice President, Sales
3	3	Leverling	Sales Representative
4	4	Peacock	Sales Representative
5	5	Buchanan	Sales Manager
6	6	Suyama	Sales Representative
7	7	King	Sales Representative
8	8	Callahan	Inside Sales Coordinator
9	9	Dodsworth	Sales Representative

Ejemplo 11:

```
--Seleccionar los campos EmployeeID y LastName de empleados EXCEPTO aquellos donde el dato almacenado
--en LastName comience con la letra D
SELECT EmployeeID, LastName FROM Employees
WHERE LastName NOT LIKE 'D%'
```

Resultado:

	EmployeeID	LastName
1	5	Buchanan
2	8	Callahan
3	2	Fuller
4	7	King
5	3	Leverling
6	4	Peacock
7	6	Suyama

ORDER BY

Si se utiliza en conjunto la cláusula ASC, los registros se mostrarán en orden ascendente (de menor a mayor) según el campo(s) especificado(s) en la cláusula ORDER BY. Si se utiliza la cláusula DESC, los registros serán mostrados en orden descendente (de mayor a menor).

Ejemplo 12:

```
-- Ordenar de forma ascendente los registros almacenados en los campos ProductID, ProductName
-- y UnitPrice de la tabla Products, se ordenaran por medio del campo ProductID
SELECT ProductID, ProductName, UnitPrice
FROM Products
ORDER BY ProductID ASC
--De forma predeterminada los datos se ordenan de forma ascendente, por lo tanto la instrucción
--ASC opcional
```

Resultado:

	ProductID	ProductName	UnitPrice
1	1	Chai	18.00
2	2	Chang	19.00
3	3	Aniseed Syrup	10.00
4	4	Chef Anton's Cajun Seasoning	22.00
5	5	Chef Anton's Gumbo Mix	21.35
6	6	Grandma's Boysenberry Spread	25.00
7	7	Uncle Roh's Organic Dried Pears	30.00

Ejemplo 13:

```
-- Ordenar de forma descendente los registros almacenados en los campos ProductID, ProductName
-- y UnitPrice de la tabla Products, se ordenaran por medio del campo ProductID
SELECT ProductID, ProductName, UnitPrice
FROM Products
ORDER BY ProductID DESC
```

Resultado:

	ProductID	ProductName	UnitPrice
1	77	Original Frankfurter grüne Soße	13.00
2	76	Lakkalikööri	18.00
3	75	Rhönbräu Klosterbier	7.75
4	74	Longlife Tofu	10.00
5	73	Röd Kaviar	15.00
6	72	Mozzarella di Giovanni	34.80
7	71	Flotemysost	21.50

DISTINCT

Los registros con ciertos datos duplicados sean ignorados en el resultado.

Ejemplo 14:

```
-- Seleccionar todos los registros no repetidos almacenados en el campo OrderID
-- de la tabla Order Details
SELECT DISTINCT OrderID FROM [Order Details]
```

Resultado:

	OrderID
1	10248
2	10249
3	10250
4	10251
5	10252

TOP

TOP n, especifica que solo se mostrará el primer conjunto de filas del resultado de la consulta.

El conjunto de filas puede ser un número o un porcentaje de las filas (TOP n PERCENT)

TOP n WITH TIES: Esta cláusula permite incluir en la selección, todos los registros que tengan el mismo valor del campo por el que se ordena

Ejemplo 15:

```
--Mostrar los primeros cinco registros de la tabla Order Details
SELECT TOP 5 OrderID, ProductID, Quantity
FROM [Order Details]
```

Resultado:

	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

Ejemplo 16:

```
--En el ejemplo siguiente se mostraran el 10% de todos los pedidos almacenados en la tabla Order Details
SELECT TOP 10 PERCENT OrderID, ProductID, Quantity
FROM [Order Details]
```

Resultado:

	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40
6	10250	41	10
7	10250	51	35
8	10250	65	15
9	10251	22	6

AS

Para colocar el sobrenombre a la columna se utiliza la instrucción AS.

Ejemplo 17:

```
--Seleccionar los datos almacenados en el campo CategoryName de la tabla Categories y renombrar
-- a la columna con el nombre Nombre de Categoria
SELECT CategoryName AS [Nombre de Categoria]
FROM Categories
```

Resultado:

	Nombre de Categoria
1	Beverages
2	Condiments
3	Confections
4	Dairy Products
5	Grains/Cereals
6	Meat/Poultry
7	Produce
8	Seafood

Ejemplo 18:

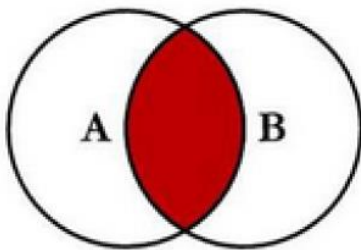
```
--Se quiere conocer cual seria la fecha de envio (ShippedDate) con un retraso de 5 dias
--Mostrar los campos OrderID, OrderDate y ShippedDate de la tabla Orders
SELECT OrderID, OrderDate, ShippedDate, ShippedDate + 5 AS RetrasoEnvio
FROM Orders
```

Resultado:

	Orderid	OrderDate	ShippedDate	Retraso Envio
1	10248	1996-07-04 00:00:00.000	1996-07-16 00:00:00.000	1996-07-21 00:00:00.000
2	10249	1996-07-05 00:00:00.000	1996-07-10 00:00:00.000	1996-07-15 00:00:00.000
3	10250	1996-07-08 00:00:00.000	1996-07-12 00:00:00.000	1996-07-17 00:00:00.000
4	10251	1996-07-08 00:00:00.000	1996-07-15 00:00:00.000	1996-07-20 00:00:00.000
5	10252	1996-07-09 00:00:00.000	1996-07-11 00:00:00.000	1996-07-16 00:00:00.000
6	10253	1996-07-10 00:00:00.000	1996-07-16 00:00:00.000	1996-07-21 00:00:00.000
7	10254	1996-07-11 00:00:00.000	1996-07-23 00:00:00.000	1996-07-28 00:00:00.000
8	10255	1996-07-12 00:00:00.000	1996-07-15 00:00:00.000	1996-07-20 00:00:00.000
9	10256	1996-07-15 00:00:00.000	1996-07-17 00:00:00.000	1996-07-22 00:00:00.000

INNER JOIN

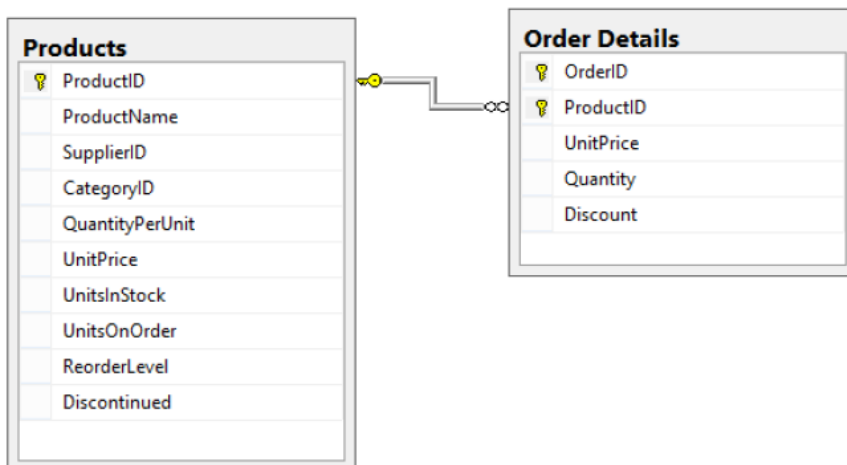
Se utiliza para mostrar los datos coincidentes entre las tablas de donde se quiere mostrar la información:



```
SELECT <lista_campos>
FROM <TablaA A>
INNER JOIN <TablaB B>
ON A.Key=B.Key
```

Nota: **ON** se utiliza para colocar los nombres de los campos con los cuales se ha realizado la relación entre las tablas.

Ejemplo 19: Se desea conocer todos los productos que se encuentran en una orden



Para obtener los registros coincidentes en ambas tablas habría que realizar la siguiente consulta:

```

SELECT OrderID, P.ProductID, ProductName
FROM Products P
INNER JOIN [Order Details] OD
ON P.ProductID=OD.ProductID

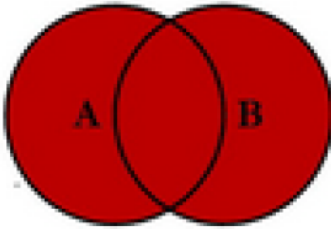
```

Y se obtendría el siguiente resultado:

	OrderID	ProductID	ProductName
1	10285	1	Chai
2	10294	1	Chai
3	10317	1	Chai
4	10348	1	Chai
5	10354	1	Chai
6	10370	1	Chai
7	10406	1	Chai
8	10413	1	Chai
9	10477	1	Chai
10	10522	1	Chai

FULL JOIN

Muestra los registros de la tabla izquierda y la tabla derecha más los registros coincidentes entre ambas



```

SELECT <lista_campos>
FROM <TablaA A>
FULL JOIN <TablaB B>
ON A.Key=B.Key

```

Ejemplo 20: En la siguiente consulta se muestra los productos que tengan o no asignado un proveedor y los proveedores independientemente si estos han ofrecido o no un producto

La consulta SQL es:

```

SELECT ProductName, CompanyName, ContactName
FROM Products P
FULL JOIN Suppliers S
ON P.SupplierID=S.SupplierID

```

Se obtiene el siguiente resultado:

	ProductName	CompanyName	ContactName
1	Producto X	NULL	NULL
2	Chai	Exotic Liquids	Charlotte Cooper
3	Chang	Exotic Liquids	Charlotte Cooper
4	Aniseed Svrup	Exotic Liquids	Charlotte Cooper
	ProductName	CompanyName	ContactName
75	Raclette Courdavault	Gai pâturage	Eliane Noz
76	Camembert Pierrot	Gai pâturage	Eliane Noz
77	Siróp d'érable	Forêts d'érables	Chantal Goulet
78	Tarte au sucre	Forêts d'érables	Chantal Goulet
79	NULL	Coca Cola	Iñaky Perez

Observar que el primer registro indica que el Producto X no tiene proveedor asignado y en el último registro el Proveedor Coca Cola no ha ofrecido ningún producto.

Guardar el script con el nombre: **Laboratorio13-[su_nombre].sql** y guardarlo bajo el proyecto Laboratorio 13 que se creará en la parte II de esta guía, el cual, posteriormente será parte de los archivos que suban a su repositorio en github

PORTE 2 – Integración de base de datos con C# bajo el IDE Visual Studio .NET

1. Crear la solución Laboratorio 13 bajo c# de tipo Aplicación de Windows Forms
2. Hacer este diseño en el formulario



3. Agregar esta instrucción para asociar a SQL Client

```
using System.Data;
using System.Data.SqlClient;
```

4. Declarar estas variables en la clase default Form

```
string connectionString =  
@"Server=.\sqlexpress;Database=Northwind;TrustServerCertificate=true;Integrated Security=SSPI;"
```

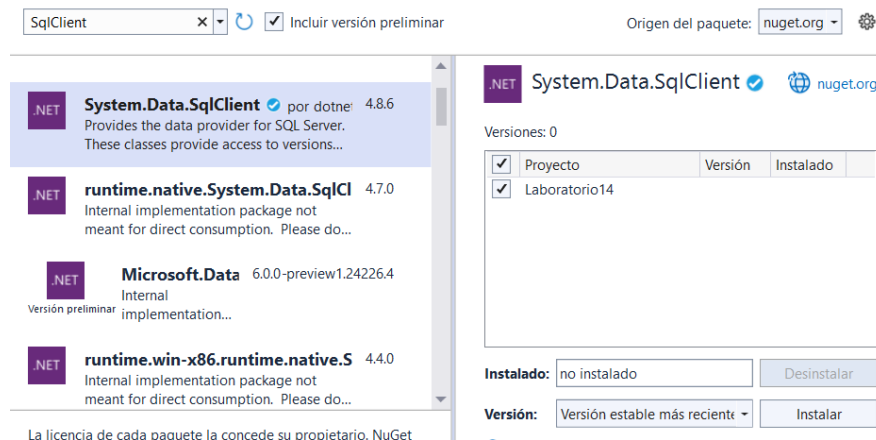
5. Agregar este código en el evento click del botón

```
SqlConnection conexion = new SqlConnection(connectionString);  
conexion.Open();  
MessageBox.Show("Se abrió la conexión con el servidor SQL Server y  
se seleccionó la base de datos");  
conexion.Close();  
MessageBox.Show("Se cerró la conexión.");
```

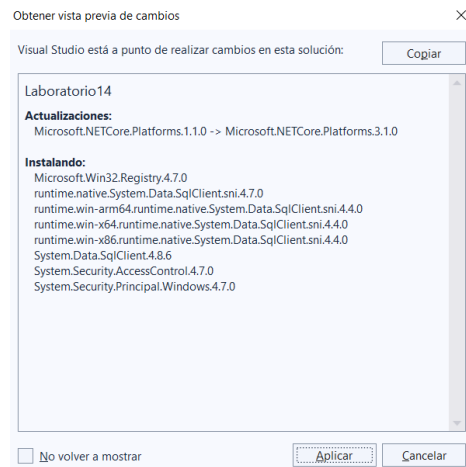
6. Probar la aplicación, ver resultados

Si el IDE no reconoce la clase SqlConnection, siga estos pasos:

- Ir al menú herramientas / administrador de paquetes nuget / administrar paquetes nuget para la solución
- Buscar SqlClient, click en instalar (para su proyecto Laboratorio 13)

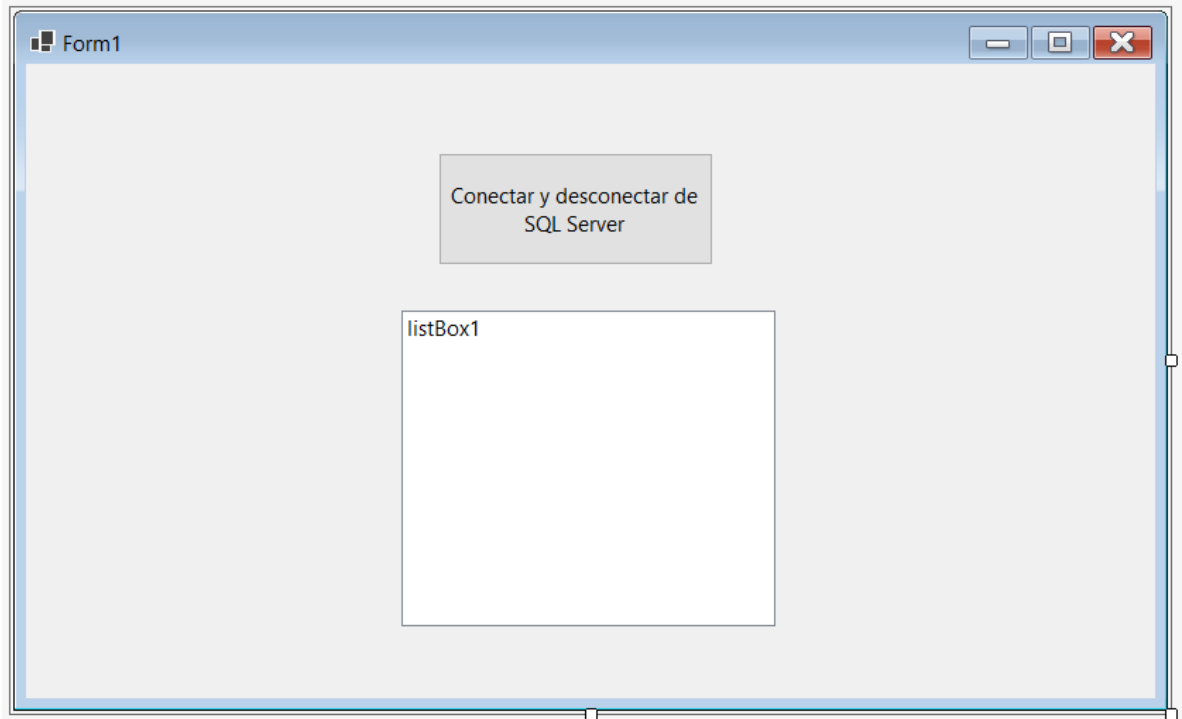


- Luego aplicar y luego acepto



Parte 3 - Listar tabla Products de Northwind

7. Agregar un listbox al formulario



8. Listar los productos mediante el query y mostrarlos en el listBox
`select` ProductName `from` [dbo].[Products]

Guardar el laboratorio 13 y subirlo a su GitHub