

	<p style="text-align: center;"><b>UNIVERSIDAD DON BOSCO</b>  <b>FACULTAD DE INGENIERIA</b>  <b>ESCUELA DE COMPUTACIÓN</b></p>
<p><b>CICLO 2-2019</b></p>	<p style="text-align: center;"><b>GUÍA DE LABORATORIO N.º 8</b></p>
	<p><b>Nombre de la practica:</b> Conexión a Base de Datos (Parte 2)  <b>Lugar de ejecución:</b> Laboratorio de Informática  <b>Materia:</b> Modelamiento y Diseño de Base de datos</p>

## I. Objetivos

1. Crear una interfaz gráfica con Visual C#.NET de manera básica
2. Utilizar las sentencias de DML de base de datos en Visual C#.NET

## II. Introducción Teórica

En la mayoría de aplicaciones incluidas en SQL Server, no todo el desarrollo se realiza en el propio servidor. Ésa es la esencia de la informática cliente servidor; el trabajo se distribuye entre un servidor central y clientes distribuidos.

Para poder ver y modificar datos del servidor desde una aplicación cliente, se utiliza una biblioteca de acceso a datos.

A lo largo de los años, Microsoft ha presentado diversas bibliotecas de acceso a datos del cliente que pueden usar los datos de SQL Server y la última tecnología que Microsoft ha desarrollado es ADO.NET.

## III. Materiales y equipo

- Computadora con SQL Server 2012, 2014, 2016 etc. y Visual Studio C#
- Guía Número 8
- Bases de ejemplo.

## IV. Procedimiento

1. Ingrese a Visual Studio y Cree la siguiente interfaz gráfica <sup>1</sup>

<sup>1</sup> Tomar de referencia la guía de laboratorio N.º7, para incorporar las librerías que correspondan.

Form1

BASE DE DATOS  
GENERADA EN C#

Generar

Clientes Pedidos

- Se va crear la base de datos mostrada, pero desde C#.

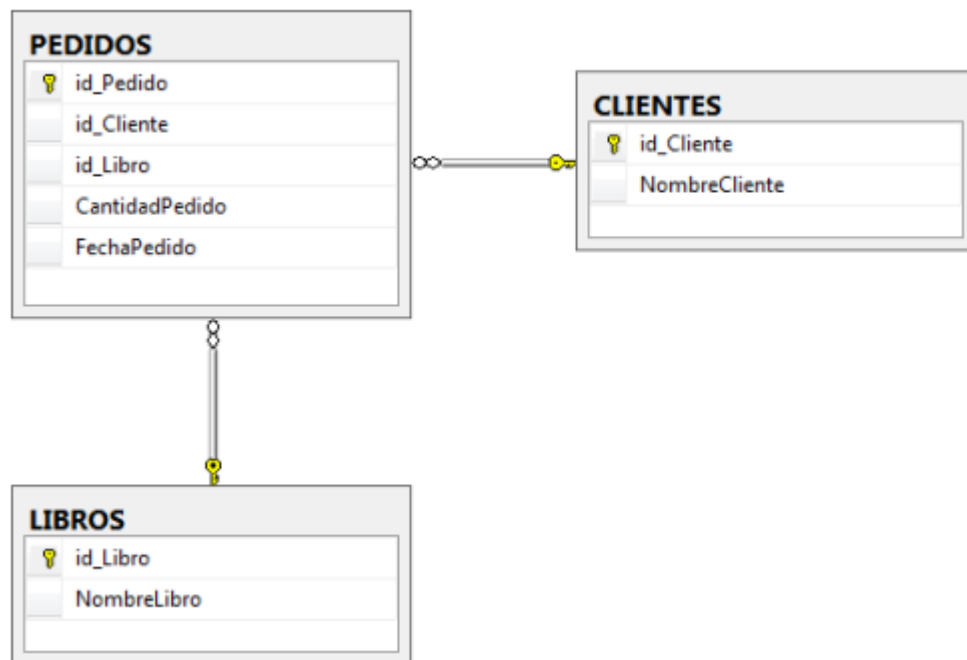


Figura 1.

- Colocar las propiedades siguientes a los elementos del **Form1**:

Elemento	Propiedades		
	Name	Visible	Text
<b>button1</b>	btnGenerar	true	Generar
<b>button2</b>	btnClientes	true	Clientes
<b>button3</b>	btnPedidos	true	Pedidos

- Agregar un formulario (**Form2**) y crear la siguiente interfaz:

Form2

## CLIENTES

Nombre

5. Colocar las propiedades siguientes a los elementos del **Form2**:

Elemento	Propiedades		
	Name	Visible	Text
<b>textbox1</b>	txtCliente	true	
<b>button1</b>	btnIngresarC	true	Ingresar
<b>button2</b>	btnLimpiarC	true	Limpiar
<b>Button3</b>	btnVolverC	true	Volver

6. Agregar un nuevo formulario (**Form3**) y crear la siguiente interfaz:

Form3

## PEDIDOS

Id Cliente

Nombre:

label6

Cantidad de Libros

Fecha

7. Colocar las propiedades siguientes a los elementos del **Form3**:

Elemento	Propiedades		
	Name	Visible	Text
comboBox1	cbCliente	true	
numericUpDown1	nUDcant	true	
dateTimePicker1	dTime1	true	
button1	btnIngresarP	true	Ingresar
button2	btnLimpiarP	true	Limpiar
button3	btnVolverP	true	Volver
label6	lbNombreC	false	

**Nota:** No olvidar agregar la librería `using System.Data.SqlClient;` en **TODOS LOS FORMULARIOS DEL PROYECTO**

- al formulario, en el **evento Click** del botón (Generar) del formulario 1 (**Form1**) colocar el siguiente código:

```
//La conexión a usar con autenticación de Windows
//Indicando la base de datos master
SqlConnection cnn = new SqlConnection(
    "Server=" + "localhost" + "; " +
    "database=master; integrated security=yes");
//La orden T-SQL para crear la base de datos es
string CrearDB = "CREATE DATABASE " + "DB_Avanzada";

//----- Creación de la tabla clientes -----
string tClientes = "USE DB_Avanzada;" +
    "CREATE TABLE CLIENTES(" +
    "id_Cliente int IDENTITY(1,1)," +
    "NombreCliente varchar(40)" +
    "CONSTRAINT PK_cliente PRIMARY KEY(id_Cliente));" ;

//----- Creación de la tabla Pedidos -----
string tPedidos = "USE DB_Avanzada;" +
    "CREATE TABLE PEDIDOS(" +
    "id_Pedido int IDENTITY(1,1)," +
    "id_Cliente int not null," +
    "CantidadPedido int," +
    "FechaPedido varchar(70)," +
    "CONSTRAINT PK_pedido PRIMARY KEY(id_Pedido)," +
    "CONSTRAINT FK_Cliente FOREIGN KEY(id_Cliente)" +
    "REFERENCES CLIENTES(id_Cliente));";

SqlCommand cmd = new SqlCommand(CrearDB, cnn);
SqlCommand cmd1 = new SqlCommand(tClientes, cnn);
SqlCommand cmd2 = new SqlCommand(tPedidos, cnn);

try
{
    cnn.Open();
```

```

cmd.ExecuteNonQuery();
cmd1.ExecuteNonQuery();
cmd2.ExecuteNonQuery();
cnn.Close();
MessageBox.Show("Base de datos" +
    "\nCreada correctamente...", "Generar",
    MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message,
        "Error al crear la Base de Datos",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}

```

Nota: Localhost: En Hardware, en el contexto de redes TCP/IP, localhost es un nombre reservado que tienen todas las computadoras, routers o dispositivos independientemente de que dispongan o no de una tarjeta de red ethernet. El nombre localhost es traducido como la dirección IP de loopback 127.0.0.1 en IPv4, o como la dirección ::1 en IPv6.

Si la conexión en el nombre del servidor no funciona con el nombre de localhost, colocar en la conexión el nombre del servidor de SQL Server

9. En el **evento Click** del botón (Clientes) del formulario 1 (**Form1**) colocar el siguiente código:

```

Form2 formu1 = new Form2();
formu1.Show();
this.Hide();

```

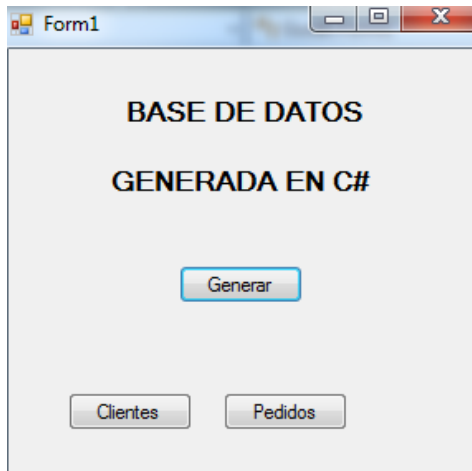
10. En el **evento Click** del botón (Pedidos) del formulario 1 (**Form1**) colocar el siguiente código:

```

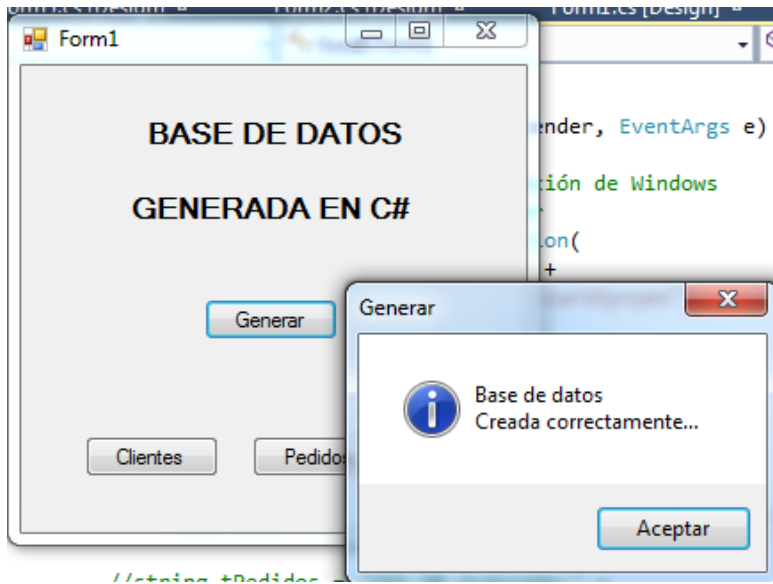
Form3 formu1 = new Form3();
formu1.Show();
this.Hide();

```

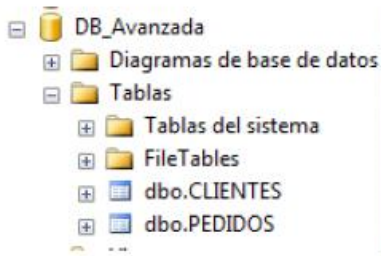
11. Compilar el proyecto y dar clic en el botón Generar



12. Muestra los siguientes resultados:



13. Abrir el programa SQL Server y verificar que la base de datos ha sido creada.



14. Ubícate en el Explorador de soluciones y dando clic derecho sobre el nombre del proyecto, seleccionar la opción agregar y luego seleccionar nuevo elemento.

15. En cuadro de dialogo nuevo elemento selecciona Clase y colócale el nombre de **conexion.cs**

16. Digita las siguientes líneas de código como se muestra a continuación:

```
class conexion
{
    //Parámetros para la cadena de conexion
    public string servidor, usuario, clave, db;
    public string cadena;
    //función que tendrá la cadena de conexión
    2 references
    public void conec()
    {
        servidor = "localhost";
        db = "DB_Avanzada";
        usuario = "sa";
        clave = "123456";
        cadena = "server="+ servidor +";uid="+ usuario + ";pwd=" + clave + ";database="+ db;
    }
}
```

17. En el **Form2** colocar el siguiente código:

```

//Variables
private SqlConnection conn;
private SqlCommand insert1;
private string sCn;

public Form2()
{
    InitializeComponent();

    //Objeto de conexión
    conexion cn = new conexion();
    //Acceso a la función conec de la clase conexión
    cn.conec();
    //Variable para la cadena de conexión
    sCn = cn.cadena;
    //Conexión
    conn = new SqlConnection(sCn);
}

```

18. En el **evento Click** del botón ("Ingresar" del **Form2**), colocar el siguiente código:

```

try
{
    conn.Open();
    //Variable insertar
    string insertCliente;
    insertCliente = "INSERT INTO CLIENTES(NombreCliente)";
    insertCliente += "VALUES (@nombreCliente)";
    insert1 = new SqlCommand(insertCliente, conn);
    insert1.Parameters.Add(
        new SqlParameter("@nombreCliente", SqlDbType.VarChar));
    insert1.Parameters["@nombreCliente"].Value = txtCliente.Text;
    insert1.ExecuteNonQuery();
    txtCliente.Clear();
    MessageBox.Show("Registro ingresado...", "Ingresar",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    conn.Close();
}
catch
{
    MessageBox.Show("El registro no fue ingresado...",
        "Error",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}

```

19. En el **evento Click** del botón ("Limpiar" del **Form2**), colocar el siguiente código:

```

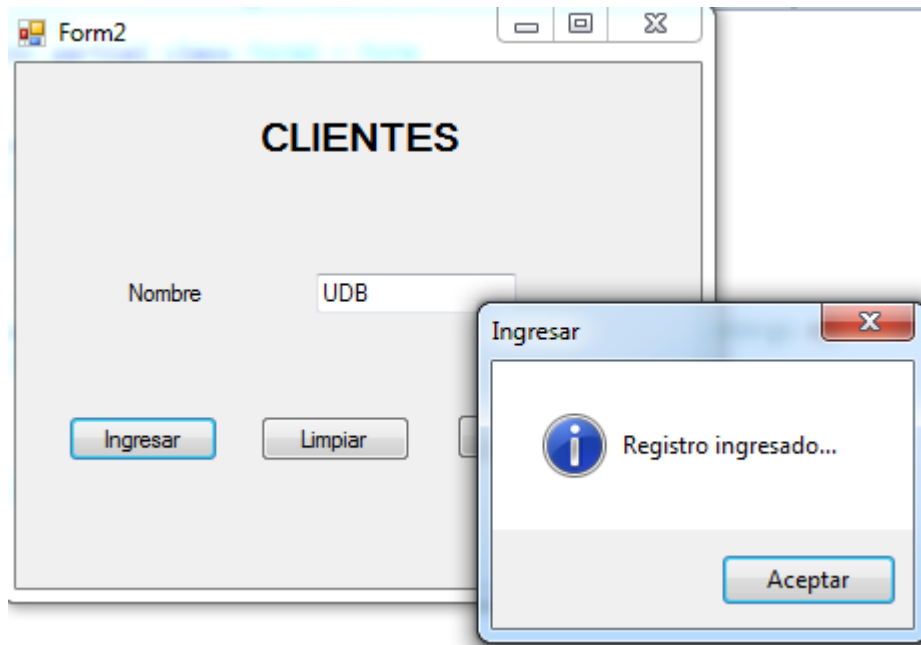
txtCliente.Clear();

```

20. En el **evento Click** del botón (“Volver” del **Form2**), colocar el siguiente código

```
Form1 formu1 = new Form1();  
formu1.Show();  
this.Hide();
```

21. Generar la aplicación y verificar que puede ejecutar los eventos Ingresar, Limpiar y Volver.



22. Abrir el programa SQL Server y verificar que el registro ha sido incorporado exitosamente.

	id_Cliente	NombreCliente
	1	UDB

23. En el **Form3** colocar el siguiente código:

```
public partial class Form3 : Form  
{  
    //variables  
    private SqlConnection conn;  
    private SqlCommand insert1;  
    private string sCn;  
    private SqlDataAdapter da1;  
    private SqlDataReader dr1;
```



```

public Form3()
{
    InitializeComponent();

    conexion cn = new conexion();
    cn.conec();
    sCn = cn.cadena;
    conn = new SqlConnection(sCn);

}

//Llenar dinámicamente combobox
public void llenar()
{
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter(
        "SELECT id_Cliente FROM CLIENTES", conn);
    da.Fill(ds, "CLIENTES");
    cbCliente.DataSource = ds.Tables[0].DefaultView;
    cbCliente.ValueMember = "id_Cliente";

}

```

24. Hacer doble clic en un espacio vacío del formulario y en el **evento Load del Form3**, llamar a la función **llenar()**

```

private void Form3_Load(object sender, EventArgs e)
{
    llenar();
}

```

25. En el **evento SelectedIndexChanged** del control (ComboBox, "cbCliente" del **Form3**), colocar el siguiente código:

```

int numero;
if (!Int32.TryParse(this.cbCliente.Text, out numero))
{
}
else
{
    try
    {
        conn.Open();
        int num1 = int.Parse(cbCliente.Text);
        string seleccion;
        seleccion = "SELECT * FROM CLIENTES WHERE Id_Cliente = " +
            num1;
        dal = new SqlDataAdapter(seleccion, conn);
        SqlParameter prm = new SqlParameter("Id_Cliente", SqlDbType.Int);
    }
}

```

```

        prm.Value = num1;
        da1.SelectCommand.Parameters.Add(prm);
        dr1 = da1.SelectCommand.ExecuteReader();
        while (dr1.Read())
        {
            lbNombreC.Visible = true;
            lbNombreC.Text = dr1["NombreCliente"].ToString().Trim();
        }
        conn.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex + "");
    }
    if (dr1 != null)
    {
        dr1.Close();
    }
}

```

26. En el **evento Click** del botón ("Ingresar" del **Form3**), colocar el siguiente código:

```

try
{
    conn.Open();
    string insertPedido;
    insertPedido = "INSERT INTO PEDIDOS(id_Cliente, CantidadPedido, FechaPedido)";
    insertPedido += "VALUES (@id_Cliente, @CantidadPedido, @FechaPedido)";
    insert1 = new SqlCommand(insertPedido, conn);
    insert1.Parameters.Add(new SqlParameter("@id_Cliente", SqlDbType.Int));
    insert1.Parameters["@id_Cliente"].Value = cbCliente.Text;
    insert1.Parameters.Add(new SqlParameter("@CantidadPedido", SqlDbType.Int));
    insert1.Parameters["@CantidadPedido"].Value = nUDcant.Value;
    insert1.Parameters.Add(new SqlParameter("@FechaPedido", SqlDbType.VarChar, 200));
    insert1.Parameters["@FechaPedido"].Value = dTime1.Text;
    insert1.ExecuteNonQuery();
    cbCliente.Text = "";
    nUDcant.Value = 0;
    dTime1.Text = "";
    MessageBox.Show("Registro agregado...",
        "Pedido",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    conn.Close();
}

```

```

catch
{
    MessageBox.Show("El registro no fue agregado...",
        "Error",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}

```

27. En el **evento Click** del botón (“Limpiar” del **Form3**), colocar el siguiente código:

```

cbCliente.Text = "";
nUDcant.Value = 0;
dTime1.Text = "";
lbNombreC.Text = "";

```

28. En el **evento Click** del botón (“Volver” del **Form3**), colocar el siguiente código:

```

Form1 formu1 = new Form1();
formu1.Show();
this.Hide();

```

## V. Ejercicio Complementario

- Complete el ejercicio con todas las tablas mostradas en la Figura 1.

## VI. Análisis de resultados

- Realice el formulario que hace falta para la tabla libros.
- Configure el botón eliminar que se pueda realizar el mantenimiento en todos los formularios.

## VII. Bibliografía

Francisco Charte Ojeda, SQL Server 2008. Madrid, España: ANAYA, 2009 1era edición