



**Medina Ortiz Jonatan Jeter**  
**Serna Villarreal Mario Augusto**  
**Ovalle Hernández Bruno**

**Manual del Programador**

**231301**

**2022-2023/II**

## Carrito

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

namespace ProyectoSQLServer
{
    public partial class CarritoForm : Form
    {
        List<Cliente> clienteList = new List<Cliente>(); //Lista de clientes
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        long idA = 0; //Variable para id de Cliente para cuando se modificara o
eliminar
        long idC = 0; //Variable para id de Carrito para cuando se modificara o
eliminar

        public CarritoForm()
        {
            InitializeComponent();
            conectaBD(); //Funcion para hacer la conexion con la base de datos
            //Inicializamos herramientas para el buen uso de nuestro programa
            comboBoxCliente.SelectedIndex = -1;
            comboBoxFormaPago.SelectedIndex = -1;
            textBoxTotal.Text = "0";
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                llenalista(); //Funcion que llena la lista
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                return -1;
            }
        }

        public void muestra()
        {
            string query1 = string.Concat("SELECT O.Id_Carrito, C.Nombre_Cliente,
O.Forma_De_Pago,O.Fecha_Venta,cast(O.Total_Carrito AS DECIMAL(10,2)), " +
"CONCAT( T.Banco, ' | ' ,RIGHT(T.Numero_Tarjeta,4)) FROM
Datos.Tarjeta_Cliente T, " +
"Operaciones.Carrito_Venta O,Usuario.Cliente C WHERE
T.Id_Tarjeta=O.Id_Tarjeta_Cliente AND C.Id_Cliente=T.Id_Cliente");

            SqlCommand command = new SqlCommand(query1, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewCarrito.DataSource = null;
            dataGridViewCarrito.DataSource = dataTable; //Llenamos el dataGrid con los
valores de las tuplas
            //Cambiamos los header de el DataGrid
            dataGridViewCarrito.Columns[0].HeaderText = "Id Carrito";
            dataGridViewCarrito.Columns[1].HeaderText = "Nombre Cliente";
            dataGridViewCarrito.Columns[2].HeaderText = "Forma de Pago";
            dataGridViewCarrito.Columns[3].HeaderText = "Fecha Venta";
            dataGridViewCarrito.Columns[4].HeaderText = "Total";
            dataGridViewCarrito.Columns[5].HeaderText = "Banco | Número Tarjeta";
            //dataGridViewCarrito.Columns[6].HeaderText = "Número Tarjeta";
            limpia(); //Funcion para limpiar los textbox
        }
    }
}
```

```

        public void limpia()
        {
            textBoxTotal.Text = "0"; //Limpiamos textbox de Total
            comboBoxCliente.SelectedIndex = -1; //Ponemos el valor de combobox cliente
en default
            comboBoxFormaPago.SelectedIndex = -1; //Ponemos el valor de combobox forma
de pago en default
            dateTimePickerFecha.Value = DateTime.Now; //Ponemos fecha del dia actual al
datetimepicker
        }

        public void llenalista()
        {
            string query = string.Concat("SELECT * FROM Usuario.Cliente"); //Sentencia
para mostrar todas la tuplas de Cliente

            SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
            SqlDataReader sqlDataReader = command.ExecuteReader(); //Declaramos variable
SqlDataReader que llevara la ejecucion del comando
            while (sqlDataReader.Read()) //Leemos lo que nos esntrego la ejecucion del
comando
            {
                //Llenamos la lista de clientes que tenemos en la base
                Cliente cliente = new Cliente();
                cliente.Id_Cliente = sqlDataReader.GetInt64(0);
                cliente.Nombre_Cliente = sqlDataReader.GetString(1);
                cliente.Correo_Electronico = sqlDataReader.GetString(2);

                clienteList.Add(cliente);
            }
cliente
            comboBoxCliente.DataSource = clienteList; //Le damos valor al combobox

            comboBoxCliente.DisplayMember = "Nombre_Cliente";
            comboBoxCliente.ValueMember = "Id_Cliente";
        }

        private void dataGridViewCarrito_CellClick(object sender,
DataGridViewCellEventArgs e) //Evento que se activa cuando hacemos click en una fila del
datagrid
        {
            //Y llena las herramientas con los valores de la fila seleccionada
            {
                textBoxTotal.Text =
dataGridViewCarrito.CurrentRow.Cells[4].Value.ToString(); //Llenamos el textbox de
total con los datos del datagridview
                dateTimePickerFecha.Value =
DateTime.Parse(dataGridViewCarrito.CurrentRow.Cells[3].Value.ToString()); //Llenamos
datetimepicker con el valor de datagrid
                idC =
Convert.ToInt64(dataGridViewCarrito.CurrentRow.Cells[0].Value.ToString()); //Damos valor
al id Carrito
                for (int i = 0; i < clienteList.Count; i++)
                {
                    if (clienteList[i].Nombre_Cliente ==
dataGridViewCarrito.CurrentRow.Cells[1].Value.ToString())
                    {
cliente
                        idA = clienteList[i].Id_Cliente; //Le damos el valor del id del
cliente
                    }
                    if (clienteList[i].Id_Cliente == idA)
                    {
                        int id =
comboBoxCliente.FindStringExact(clienteList[i].Nombre_Cliente); //Seleccionamos el id
del cliente
                        comboBoxCliente.SelectedIndex = id; //Damos valor al combobox
                    }
                }
                for (int i = 0; i < comboBoxFormaPago.Items.Count; i++)
                {
                    if (dataGridViewCarrito.CurrentRow.Cells[2].Value.ToString() ==
comboBoxFormaPago.Items[i].ToString())
                    {
                        int id =
comboBoxFormaPago.FindStringExact(dataGridViewCarrito.CurrentRow.Cells[2].Value.ToString
()); //Seleccionamos el id de la forma de pago
                        comboBoxFormaPago.SelectedIndex = id; //Damos valor al combobox
                    }
                }
            }
        }
    }

```

```

    }
}

private void buttonAgregar_Click(object sender, EventArgs e)
{
    insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void insertaRegistro()
{
    try
    {
        conexion.Open(); //Abre la conexion
        string fecha = dateTimePickerFecha.Value.ToString("yyyy-MM-dd");
        string consulta = "INSERT INTO Operaciones.Carrito_Venta" +
            "(Id_Tarjeta_Cliente, Forma_De_Pago, Fecha_Venta, Total_Carrito) " +
            "VALUES(" + "(SELECT Id_Tarjeta FROM Datos.Tarjeta_Cliente T WHERE "
            + clienteList[comboBoxCliente.SelectedIndex].Id_Cliente + " = T.Id_Cliente)" + "," +
            + comboBoxFormaPago.Text + "',' + fecha + "',' + textBoxTotal.Text
            + ")";

        //Sentencia para insertar alumnos con los datos de las textbox
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
        variable SqlDataAdapter
        command.ExecuteNonQuery(); //Ejecutamos el comando
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error" + ex.Message); //Error en la conexion
    }
}

private void buttonModificar_Click(object sender, EventArgs e)
{
    modificaRegistro(); //Mandamos a llamar funcion para modificar la tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void modificaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string fecha = dateTimePickerFecha.Value.ToString("yyyy-MM-dd");
        string consulta = "UPDATE Operaciones.Carrito_Venta SET
        Id_Tarjeta_Cliente=" + "(SELECT T.Id_Tarjeta FROM Datos.Tarjeta_Cliente T WHERE
        T.Id_Cliente = "
            + clienteList[comboBoxCliente.SelectedIndex].Id_Cliente + ")"
            + ",Forma_De_Pago='" + comboBoxFormaPago.Text.ToString() +
            "','Fecha_Venta = '" + fecha + "','Total_Carrito=" + textBoxTotal.Text;
        //Sentencia en SQL para modificar la tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
        variable de tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos la sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos los textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error"); //Error en la conexion
    }
}

private void buttonEliminar_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

```

```

public void eliminaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "DELETE Operaciones.Carrito_Venta WHERE Id_Carrito=" +
idC; //Sentencia en SQL para eliminar tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cerramos conexxion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}

//FUNCIONES PARA ABRIR OTROS FORMS

private void buttonDetalle_Click(object sender, EventArgs e)
{
    DetalleCarrito detalleCarrito = new DetalleCarrito();
    detalleCarrito.ShowDialog();
}

private void buttonDevolucion_Click(object sender, EventArgs e)
{
    DevolucionCarrito devolucionCarrito = new DevolucionCarrito();
    devolucionCarrito.ShowDialog();
}
}
}

```

## Clases

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

namespace ProyectoSQLServer
{
    internal class Clases
    {
    }

    /*
    TarjetaC: Esta clase representa una tarjeta de crédito y tiene las siguientes
    propiedades:
        Id_Tarjeta: Identificador de la tarjeta (tipo long).
        Id_Cliente: Identificador del cliente asociado a la tarjeta (tipo long).
        Num_Tarjeta: Número de la tarjeta (tipo string).
        CVV: Código de seguridad de la tarjeta (tipo string).
        Banco: Nombre del banco emisor de la tarjeta (tipo string).
        Fecha: Fecha asociada a la tarjeta (tipo DateTime).
        Además, la clase tiene dos constructores: uno vacío y otro que acepta los
        valores de todas las propiedades.
    */

    public class TarjetaC
    {
        public long Id_Tarjeta { get; set; }
        public long Id_Cliente { get; set; }
        public string Num_Tarjeta { get; set; }
        public string CVV { get; set; }
        public string Banco { get; set; }
        public DateTime Fecha { get; set; }

        public TarjetaC() { }
        public TarjetaC(long id_Tarjeta, long id_Cliente, string num_Tarjeta, string
cVV, string banco, DateTime fecha)
        {
            Id_Tarjeta = id_Tarjeta;
            Id_Cliente = id_Cliente;
            Num_Tarjeta = num_Tarjeta;
            CVV = cVV;
            Banco = banco;
            Fecha = fecha;
        }
    }

    /*
    Carrito: Esta clase representa un carrito de compras y contiene las siguientes
    propiedades:
        Id_Carrito: Identificador del carrito (tipo long).
        Id_Tarjeta: Identificador de la tarjeta asociada al carrito (tipo long).
        FormaPago: Forma de pago utilizada en el carrito (tipo string).
        FechaVenta: Fecha de venta del carrito (tipo DateTime).
        Total: Total de la venta (tipo decimal).
        Cliente: Cliente asociado al carrito (tipo string).
        idynombre: Propiedad de nombre y id juntos (tipo string).
        La clase también tiene dos constructores, uno sin parámetros y otro que
        acepta los valores de todas las propiedades.
    */

    public class Carrito
    {
        public long Id_Carrito { get; set; }
        public long Id_Tarjeta { get; set; }
        public string FormaPago { get; set; }
        public DateTime FechaVenta { get; set; }
        public decimal Total { get; set; }
        public string Cliente { get; set; }
        public string idynombre { get; set; }

        public Carrito() { }
```

```

public Carrito(long id_Carrito, long id_Tarjeta, string formaPago, DateTime fechaVenta,
decimal total, string cliente, string idynombre)
{
    Id_Carrito = id_Carrito;
    Id_Tarjeta = id_Tarjeta;
    FormaPago = formaPago;
    FechaVenta = fechaVenta;
    Total = total;
    Cliente = cliente;
    this.idynombre = idynombre;
}

}

/*
Orden: Esta clase representa una orden y tiene las siguientes propiedades:
    Num_Orden: Número de orden (tipo long).
    Id_Producto: Identificador del producto asociado a la orden (tipo long).
    Fecha_Orden: Fecha de la orden (tipo DateTime).
    Total: Total de la orden (tipo decimal).
    Al igual que las clases anteriores, Orden también tiene dos constructores.
*/

public class Orden
{
    public long Num_Orden { get; set; }
    public long Id_Producto { get; set; }
    public DateTime Fecha_Orden { get; set; }
    public decimal Total { get; set; }

    public Orden() { }
    public Orden(long num, long p, DateTime date, int t)
    {
        this.Num_Orden = num;
        this.Id_Producto = p;
        this.Fecha_Orden = date;
        this.Total = t;
    }
}

/*
Producto: Esta clase representa una orden y tiene las siguientes propiedades:
    Id_Producto: Identificador del producto (tipo long).
    Nombre_Producto: Nombre del producto (tipo string).
    Precio_Publico_Producto: Precio público del producto (tipo decimal).
    Precio_Proveedor_Producto: Precio proveedor del producto (tipo decimal).
    Color: Color del producto (tipo string).
    Stock: Cantidad en stock del producto (tipo int).
    Al igual que las clases anteriores, Producto también tiene dos
constructores.
*/

public class Producto
{
    public long Id_Producto { get; set; }
    public string Nombre_Producto { get; set; }
    public decimal Precio_Publico_Producto { get; set; }
    public decimal Precio_Proveedor_Producto { get; set; }
    public string Color { get; set; }
    public int Stock { get; set; }

    public Producto() { }
    public Producto(long Id, string Nombre, int PPublicoP, int PProveedorP, string
c, int s)
    {
        this.Id_Producto = Id;
        this.Nombre_Producto = Nombre;
        this.Precio_Publico_Producto = PPublicoP;
        this.Precio_Proveedor_Producto = PProveedorP;
        this.Color = c;
        this.Stock = s;
    }
}

```

```

/*
    Cliente: Esta clase representa un cliente y tiene las siguientes propiedades:
        Id_Cliente: Identificador del cliente (tipo long).
        Nombre_Cliente: Nombre del cliente (tipo string).
        Correo_Electronico: Correo electrónico del cliente (tipo string).
        tarjeta: Objeto de tipo Tarjeta que representa la tarjeta asociada
*/

public class Cliente
{
    public long Id_Cliente { get; set; }
    public string Nombre_Cliente { get; set; }
    public string Correo_Electronico { get; set; }
    public Tarjeta tarjeta { get; set; }

    public Cliente() { }
    public Cliente(long Id, string Nombre, string Correo, Tarjeta tarjeta)
    {
        this.Id_Cliente = Id;
        this.Nombre_Cliente = Nombre;
        this.Correo_Electronico = Correo;
        this.tarjeta = tarjeta;
    }
}

/*
    Proveedor: Esta clase representa un proveedor y contiene las siguientes propiedades:
        Id_Proveedor: Identificador del proveedor (tipo long).
        Nombre_Proveedor: Nombre del proveedor (tipo string).
        Correo_Electronico: Correo electrónico del proveedor (tipo string).
        Domicilio_Fiscal: Domicilio fiscal del proveedor (tipo string).
    La clase Proveedor también tiene dos constructores, uno sin parámetros y otro
    que acepta los valores de todas las propiedades.
*/

public class Proveedor
{
    public long Id_Proveedor { get; set; }
    public string Nombre_Proveedor { get; set; }
    public string Correo_Electronico { get; set; }
    public string Domicilio_Fiscal { get; set; }

    public Proveedor() { }
    public Proveedor(long Id, string Nombre, string Correo, string Domicilio)
    {
        this.Id_Proveedor = Id;
        this.Nombre_Proveedor = Nombre;
        this.Correo_Electronico = Correo;
        this.Domicilio_Fiscal = Domicilio;
    }
}

/*
    Tarjeta: Esta clase representa una tarjeta de crédito y tiene las siguientes
    propiedades:
        Id_Tarjeta: Identificador de la tarjeta (tipo long).
        Id_Cliente: Identificador del cliente asociado a la tarjeta (tipo long).
        Numero_Tarjeta: Número de la tarjeta (tipo long).
        CVV: Código de seguridad de la tarjeta (tipo string).
        Banco: Nombre del banco emisor de la tarjeta (tipo string).
        Fecha: Fecha asociada a la tarjeta (tipo DateTime).
    La clase Tarjeta también tiene dos constructores, uno sin parámetros y otro que
    acepta los valores de todas las propiedades.
*/

public class Tarjeta
{
    public long Id_Tarjeta { get; set; }
    public long Id_Cliente { get; set; }
    public long Numero_Tarjeta { get; set; }
    public string CVV { get; set; }
    public string Banco { get; set; }
    public DateTime Fecha { get; set; }

    public Tarjeta() { }

```



```
        public Tarjeta(long id_Tarjeta, long id_Cliente, long numero_Tarjeta, string
cVV, string banco, DateTime fecha)
        {
            Id_Tarjeta = id_Tarjeta;
            Id_Cliente = id_Cliente;
            Numero_Tarjeta = numero_Tarjeta;
            CVV = cVV;
            Banco = banco;
            Fecha = fecha;
        }
    }
}
```

## Cliente

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

namespace ProyectoSQLServer
{
    public partial class ClienteForm : Form
    {
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        int idA = 0; //Variable para id de Cliente para cuando se modificara o eliminara

        public ClienteForm()
        {
            InitializeComponent();
            conectaBD();
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                return -1;
            }
        }

        public void muestra()
        {
            string query = string.Concat("SELECT * FROM Usuario.Cliente"); //Sentencia
            para mostrar todas la tuplas

            SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
            de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
            SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewCliente.DataSource = null;
            dataGridViewCliente.DataSource = dataTable; //Llenamos el dataGrid con los
            valores de las tuplas
            dataGridViewCliente.Columns[0].HeaderText = "Id Cliente";
            dataGridViewCliente.Columns[1].HeaderText = "Nombre Cliente";
            dataGridViewCliente.Columns[2].HeaderText = "Correo Electrónico";
            //comboBoxCarrera.
            limpia(); //Funcion para limpiar los textbox
        }

        public void limpia()
        {
            textBoxNombre.Clear(); //Limpiamos textbox de Nombre
            textBoxCorreo.Clear(); //Limpiamos textbox de Correo
        }

        private void dataGridViewCliente_CellClick(object sender,
            DataGridViewCellEventArgs e)
        {
            idA = Convert.ToInt32(dataGridViewCliente.CurrentRow.Cells[0].Value);
            //Indicamos el id de la tupla que seleccionamos
            textBoxNombre.Text =
            dataGridViewCliente.CurrentRow.Cells[1].Value.ToString(); //Llenamos el textbox de
            nombre con los datos del datagridview
            textBoxCorreo.Text =
            dataGridViewCliente.CurrentRow.Cells[2].Value.ToString(); //Llenamos el textbox de correo
            con los datos del datagridview
        }
    }
}
```

```

//-----
private void buttonAgregar_Click(object sender, EventArgs e)
{
    insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void insertaRegistro()
{
    try
    {
        conexion.Open(); //Abre la conexion
        string consulta = "INSERT INTO Usuario.Cliente" + "(Nombre_Cliente,
Correo_Electronico) " +
        "VALUES('" + textBoxNombre.Text + "','" + textBoxCorreo.Text +
        "')";
        //Sentencia para insertar alumnos con los datos de las textbox
        SqlCommand command = new SqlCommand(consulta, conexion);
//Declaramos variable SqlDataAdapter
        command.ExecuteNonQuery(); //Ejecutamos el comando
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}

//-----

private void buttonModificar_Click(object sender, EventArgs e)
{
    modificaRegistro(); //Mandamos a llamar funcion para modificar la tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void modificaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "UPDATE Usuario.Cliente SET Nombre_Cliente='" +
textBoxNombre.Text + "','" + Correo_Electronico='" +
        + textBoxCorreo.Text + "' WHERE Id_Cliente=" + idA;
        //Sentencia en SQL para modificar la tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable de tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos la sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos los textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}

//-----

```

```

private void buttonEliminar_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

public void eliminaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "DELETE Usuario.Cliente WHERE Id_Cliente=" + idA;
//Sentencia en SQL para eliminar tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cerramos conexxion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}

private void buttonTelefonoCliente_Click(object sender, EventArgs e)
{
    TelefonoClienteForm telefonoClienteForm = new TelefonoClienteForm();
    telefonoClienteForm.ShowDialog();
}

private void buttonTarjeta_Click(object sender, EventArgs e)
{
    Tarjeta_Cliente tarjeta_Cliente = new Tarjeta_Cliente();
    tarjeta_Cliente.ShowDialog();
}
}
}

```

## DetalleCarrito

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

namespace ProyectoSQLServer
{
    public partial class DetalleCarrito : Form
    {
        List<Producto> productoList = new List<Producto>();
        List<Carrito> carritoList = new List<Carrito>();
        List<Cliente> clienteList = new List<Cliente>();
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        int idA = 0; //Variable para id de Cliente para cuando se modificara o eliminara
        long idP = 0; //Variable para id de Producto para cuando se modificara o
        eliminara

        public DetalleCarrito()
        {
            InitializeComponent();
            conectaBD(); //Funcion para hacer la conexion con la base de datos
            //Inicializamos herramientas para el buen uso de nuestro programa
            comboBoxCarrito.SelectedIndex = -1;
            comboBoxProducto.SelectedIndex = -1;
            textBoxSubtotal.Text = "0";
            textBoxCantidad.Clear();
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                llenalista();
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                return -1;
            }
        }

        public void muestra()
        {
            string query1 = string.Concat("SELECT CONCAT(C.Id_Carrito, ' | ',
U.Nombre_Cliente), D.Cantidad, cast(D.SubTotal AS DECIMAL(10,2)), CONCAT(
P.Nombre_Producto, " +
            "' | ', cast(P.Precio_Publico_Producto AS DECIMAL(10,2)) )" +
            "FROM Operaciones.Detalle_Carrito D, Articulo.Producto
P, Operaciones.Carrito_Venta C, Usuario.Cliente U, Datos.Tarjeta_Cliente T " +
            "WHERE C.Id_Carrito = D.Id_Carrito AND D.Id_Producto = P.Id_Producto AND
U.Id_Cliente = T.Id_Cliente AND T.Id_Tarjeta = C.Id_Tarjeta_Cliente ");

            SqlCommand command = new SqlCommand(query1, conexion); //Declaramos variable
            de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
            SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewDetalleCarrito.DataSource = null;
            dataGridViewDetalleCarrito.DataSource = dataTable; //Llenamos el dataGrid
            con los valores de las tuplas
            dataGridViewDetalleCarrito.Columns[0].HeaderText = "Carrito | Cliente";
            dataGridViewDetalleCarrito.Columns[1].HeaderText = "Cantidad";
            dataGridViewDetalleCarrito.Columns[2].HeaderText = "SubTotal";
            dataGridViewDetalleCarrito.Columns[3].HeaderText = "Nombre Producto | Precio
            Publico Producto";
            //dataGridViewDetalleCarrito.Columns[4].HeaderText = "Precio Publico
            Producto";
        }
    }
}
```

```

        CarritoForm carritoForm =
Application.OpenForms.OfType<CarritoForm>().FirstOrDefault();
        if (carritoForm != null)
            carritoForm.muestra();
        limpia(); //Funcion para limpiar los textbox
    }

    public void limpia()
    {
        //Limpiamos herramientas de form
        comboBoxCarrito.SelectedIndex = -1;
        comboBoxProducto.SelectedIndex = -1;
        textBoxSubtotal.Text = "0";
        textBoxCantidad.Clear();
    }

    public void llenalista()
    {
        string query = string.Concat("SELECT * FROM Articulo.Producto"); //Sentencia
para mostrar todas la tuplas

        SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
        SqlDataReader sqlDataReader = command.ExecuteReader();
        while (sqlDataReader.Read())
        {
            //Llenamos la lista de Producto que tenemos en la base
            Producto producto = new Producto();
            producto.Id_Producto = sqlDataReader.GetInt64(0);
            producto.Nombre_Producto = sqlDataReader.GetString(1) + " | " +
Math.Round(sqlDataReader.GetDecimal(2), 2);
            //producto.Precio_Publico_Producto = sqlDataReader.GetDecimal(2);
            producto.Precio_Proveedor_Producto = sqlDataReader.GetDecimal(3);
            producto.Color = sqlDataReader.GetString(4);
            //producto.Stock = sqlDataReader.GetInt32(5);

            productoList.Add(producto);
        }
        comboBoxProducto.DataSource = productoList; //Le damos valor al combobox
Producto
        comboBoxProducto.DisplayMember = "Nombre_Producto";
        comboBoxProducto.ValueMember = "Id_Producto";

        query = string.Concat("SELECT C.Id_Carrito, C.Id_Tarjeta_Cliente,
C.Forma_De_Pago, C.Fecha_Venta, C.Total_Carrito, U.Nombre_Cliente" +
            " FROM Operaciones.Carrito_Venta C, Datos.Tarjeta_Cliente T,
Usuario.Cliente U " +
            "WHERE T.Id_Cliente = U.Id_Cliente AND T.Id_Tarjeta =
C.Id_Tarjeta_Cliente"); //Sentencia para mostrar todas la tuplas

        sqlDataReader.Close();
        command = new SqlCommand(query, conexion); //Declaramos variable de
SqlCommand que lleva la sentencia y la conexion
        SqlDataReader = command.ExecuteReader();
        while (sqlDataReader.Read())
        {
            //Llenamos la lista de Carrito que tenemos en la base
            Carrito carrito = new Carrito();
            carrito.Id_Carrito = sqlDataReader.GetInt64(0);
            carrito.Id_Tarjeta = sqlDataReader.GetInt64(1);
            carrito.FormaPago = sqlDataReader.GetString(2);
            carrito.FechaVenta = sqlDataReader.GetDateTime(3);
            carrito.Total = sqlDataReader.GetDecimal(4);
            carrito.Cliente = sqlDataReader.GetString(5);
            carrito.idynombre = carrito.Id_Carrito + " - " + carrito.Cliente;

            carritoList.Add(carrito);
        }
        comboBoxCarrito.DataSource = carritoList; //Le damos valor al combobox
Carrito
        comboBoxCarrito.DisplayMember = "idynombre";
        comboBoxCarrito.ValueMember = "idynombre";
    }

```

```

        private void dataGridViewDetalleCarrito_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            string g = dataGridViewDetalleCarrito.CurrentRow.Cells[0].Value.ToString();
            string idCa = ""; //El siguiente ciclo es para extraer solo el id del Carrito
            for (int i = 0; i < g.Length; i++)
            {
                if (g[i] != ' ')
                {
                    idCa += g[i];
                }
                else
                {
                    break;
                }
            }

            idA = int.Parse(idCa); //Convertirmos el id a int
            for (int i = 0; i < productoList.Count; i++)
            {
                if (productoList[i].Nombre_Producto ==
dataGridViewDetalleCarrito.CurrentRow.Cells[3].Value.ToString())
                {
                    idP = productoList[i].Id_Producto; //Damos valor al idP
                    break;
                }
            }
            textBoxCantidad.Text =
dataGridViewDetalleCarrito.CurrentRow.Cells[1].Value.ToString(); //Llenamos el textbox
de cantidad con los datos del datagridview
            textBoxSubtotal.Text =
dataGridViewDetalleCarrito.CurrentRow.Cells[2].Value.ToString(); //Llenamos el textbox
de cantidad con los datos del datagridview
            for (int i = 0; i < carritoList.Count; i++)
            {
                if (carritoList[i].Id_Carrito == idA)
                {
                    int id =
comboBoxCarrito.FindString(carritoList[i].Id_Carrito.ToString());
                    comboBoxCarrito.SelectedIndex = id; //Seleccionamos el el combobox
el elegido
                    break;
                }
            }
            for (int i = 0; i < productoList.Count; i++)
            {
                if (productoList[i].Id_Producto == idP)
                {
                    int id =
comboBoxProducto.FindStringExact(productoList[i].Nombre_Producto);
                    comboBoxProducto.SelectedIndex = id; //Seleccionamos en el combobox
el elegido
                    break;
                }
            }
        }

        private void buttonAgregar_Click(object sender, EventArgs e)
        {
            insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
            conectaBD(); //Conectamos a la Base de datos
        }

```

```

public void insertaRegistro()
{
    try
    {
        if (textBoxCantidad.Text == "0")
        {
            MessageBox.Show("Cantidad invalida");
        }
        else
        {
            conexion.Open(); //Abre la conexion
            string consulta = "INSERT INTO Operaciones.Detalle_Carrito
(Id_Carrito, Id_Producto,Cantidad,SubTotal) " +
            "VALUES(" +
            carritoList[comboBoxCarrito.SelectedIndex].Id_Carrito + "," +
            productoList[comboBoxProducto.SelectedIndex].Id_Producto + "','" + textBoxCantidad.Text +
            "','" + textBoxSubtotal.Text + "')";
            //Sentencia para insertar detalles carrito con los datos de las
            textbox
            SqlCommand command = new SqlCommand(consulta, conexion);
            //Declaramos variable SqlDataAdapter
            command.ExecuteNonQuery(); //Ejecutamos el comando
            conexion.Close(); //Cerramos conexion
            limpia(); //Limpiamos textbox
        }
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error: " + ex.Message); //Error en la conexion
    }
}

private void buttonModificar_Click(object sender, EventArgs e)
{
    modificaRegistro(); //Mandamos a llamar funcion para modificar la tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void modificaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "UPDATE Operaciones.Detalle_Carrito SET Id_Carrito=" +
        carritoList[comboBoxCarrito.SelectedIndex].Id_Carrito + ",Id_Producto="
        + productoList[comboBoxProducto.SelectedIndex].Id_Producto +
        ",Cantidad=" + textBoxCantidad.Text + ", SubTotal=" + textBoxSubtotal.Text +
        " WHERE Id_Carrito=" + idA + " AND Id_Producto=" + idP;
        //Sentencia en SQL para modificar la tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
        variable de tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos la sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos los textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error: Numero Duplicado de Tarjeta"); //Error en la
        conexion
    }
}
}

```



```

private void buttonEliminar_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

public void eliminaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "DELETE Operaciones.Detalle_Carrito WHERE Id_Carrito="
+ idA + "AND Id_Producto=" + idP + "AND Cantidad=" + textBoxCantidad.Text +
        "AND Subtotal=" + textBoxSubtotal.Text ; //Sentencia en SQL para
eliminar tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cerramos conexxion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}
}
}

```

## DetalleOrden

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

namespace ProyectoSQLServer
{
    public partial class DetalleOrdenForm : Form
    {
        List<Producto> productoList = new List<Producto>();
        List<Orden> ordenList = new List<Orden>();
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        int idA = 0; //Variable para id de Cliente para cuando se modificara o eliminara
        long idP = 0;
        int idO = 0;

        public DetalleOrdenForm()
        {
            InitializeComponent();
            conexion.Open(); //Abrimos conexion con la base de datos
            llenalista(); //Funcion que muestra en el datagridview las tuplas
            conexion.Close();
            conectaBD();
            //Inicializamos herramientas para el buen uso de nuestro programa
            comboBoxOrden.SelectedIndex = -1;
            comboBoxProducto.SelectedIndex = -1;
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                return -1;
            }
        }

        public void muestra()
        {
            string query1 = string.Concat("SELECT O.Num_Orden,
D.Cantidad,cast(D.SubTotal AS DECIMAL(10,2)),CONCAT( P.Nombre_Producto, ' | '
,cast(P.Precio_Publico_Proveedor AS DECIMAL(10,2))) " +
"FROM Operaciones.Orden O,Operaciones.Detalle_Orden D,Articulo.Producto
P " +
"WHERE O.Num_Orden = D.Num_Orden AND D.Id_Producto = P.Id_Producto");

            SqlCommand command = new SqlCommand(query1, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewDetalleOrden.DataSource = null;
            dataGridViewDetalleOrden.DataSource = dataTable; //Llenamos el dataGrid con
los valores de las tuplas
            dataGridViewDetalleOrden.Columns[0].HeaderText = "Número Orden";
            dataGridViewDetalleOrden.Columns[1].HeaderText = "Cantidad";
            dataGridViewDetalleOrden.Columns[2].HeaderText = "SubTotal";
        }
    }
}
```

```

        dataGridViewDetalleOrden.Columns[3].HeaderText = "Nombre Producto | Precio
Proveedor";
        //dataGridViewDetalleOrden.Columns[4].HeaderText = "Precio Proveedor";
        OrdenForm ordenForm =
Application.OpenForms.OfType<OrdenForm>().FirstOrDefault();
        if(ordenForm != null)
            ordenForm.muestra();
        limpia(); //Funcion para limpiar los textbox
    }

    public void limpia()
    {
        //Limpiamos herramientas para el buen uso de nuestro programa
        textBoxCantidad.Clear();
        comboBoxOrden.SelectedIndex = -1;
        comboBoxProducto.SelectedIndex = -1;
    }

    public void llenalista()
    {
        string query = string.Concat("SELECT * FROM Articulo.Producto"); //Sentencia
para mostrar todas la tuplas

        SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
        SqlDataReader sqlDataReader = command.ExecuteReader();
        //Llenamos lista de Productos
        while (sqlDataReader.Read())
        {
            Producto producto = new Producto();
            producto.Id_Producto = sqlDataReader.GetInt64(0);
            producto.Nombre_Producto = sqlDataReader.GetString(1) + " | " +
Math.Round(sqlDataReader.GetDecimal(3), 2);
            producto.Precio_Publico_Producto = sqlDataReader.GetDecimal(2);
            //producto.Precio_Proveedor_Producto = sqlDataReader.GetDecimal(3);
            producto.Color = sqlDataReader.GetString(4);

            productList.Add(producto);
        }
        comboBoxProducto.DataSource = productList;
        comboBoxProducto.DisplayMember = "Nombre_Producto";
        comboBoxProducto.ValueMember = "Id_Producto";

        query = string.Concat("SELECT * FROM Operaciones.Orden"); //Sentencia para
mostrar todas la tuplas

        sqlDataReader.Close();
        command = new SqlCommand(query, conexion); //Declaramos variable de
SqlCommand que lleva la sentencia y la conexion
        sqlDataReader = command.ExecuteReader();
        //Llenamos lista de ordenes
        while (sqlDataReader.Read())
        {
            Orden orden = new Orden();
            orden.Num_Orden = sqlDataReader.GetInt64(0);
            orden.Id_Producto = sqlDataReader.GetInt64(1);
            orden.Fecha_Orden = sqlDataReader.GetDateTime(2);
            orden.Total = sqlDataReader.GetDecimal(3);

            ordenList.Add(orden);
        }
        comboBoxOrden.DataSource = ordenList;
        comboBoxOrden.DisplayMember = "Num_Orden";
        comboBoxOrden.ValueMember = "Num_Orden";
    }
}

```

```

        private void dataGridViewDetalleOrden_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            idA = Convert.ToInt32(dataGridViewDetalleOrden.CurrentRow.Cells[0].Value);
//Indicamos el id de la tupla que seleccionamos
            for (int i = 0; i < productList.Count; i++)
            {
                if (productoList[i].Nombre_Producto ==
dataGridViewDetalleOrden.CurrentRow.Cells[3].Value.ToString())
                {
                    idP = productoList[i].Id_Producto; //Agregamos id de producto
                    break;
                }
            }
            textBoxCantidad.Text =
dataGridViewDetalleOrden.CurrentRow.Cells[1].Value.ToString(); //Llenamos el textbox de
nombre con los datos del datagridview
            for (int i = 0; i < ordenList.Count; i++)
            {
                if (ordenList[i].Num_Orden == idA)
                {
                    int id =
comboBoxOrden.FindStringExact(ordenList[i].Num_Orden.ToString());
                    comboBoxOrden.SelectedIndex = id; //Seleccionamos el valor en el
comboBox
                    break;
                }
            }
            for (int i = 0; i < productList.Count; i++)
            {
                if (productoList[i].Id_Producto == idP)
                {
                    int id =
comboBoxProducto.FindStringExact(productoList[i].Nombre_Producto);
                    comboBoxProducto.SelectedIndex = id; //Seleccionamos el valor en el
comboBox
                    break;
                }
            }
        }

        private void buttonAgregar_Click(object sender, EventArgs e)
        {
            insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
            conectaBD(); //Conectamos a la Base de datos
        }

        public void insertaRegistro()
        {
            try
            {
                conexion.Open(); //Abre la conexion
                string consulta = "INSERT INTO Operaciones.Detalle_Orden" + "(Num_Orden,
Id_Producto,Cantidad) " +
                "VALUES(" + ordenList[comboBoxOrden.SelectedIndex].Num_Orden + "," +
                productoList[comboBoxProducto.SelectedIndex].Id_Producto + "," + textBoxCantidad.Text +
                ")";
                //Sentencia para insertar detalles con los datos de las textbox
                SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
                variable SqlDataAdapter
                command.ExecuteNonQuery(); //Ejecutamos el comando
                conexion.Close(); //Cerramos conexion
                limpia(); //Limpiamos textbox
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cierra conexion
                MessageBox.Show("Error: " + ex.Message ); //Error en la conexion
            }
        }
    }

```

```

private void buttonModificar_Click(object sender, EventArgs e)
{
    modificaRegistro();//Mandamos a llamar funcion para modificar la tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void modificaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "UPDATE Operaciones.Detalle_Orden SET Num_Orden=" +
ordenList[comboBoxOrden.SelectedIndex].Num_Orden + ",Id_Producto="
        + productList[comboBoxProducto.SelectedIndex].Id_Producto +
",Cantidad=" + textBoxCantidad.Text + " WHERE Num_Orden=" + idA + " AND Id_Producto=" +
idP;

        //Sentencia en SQL para modificar la tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable de tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos la sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos los textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error: Numero Duplicado de Tarjeta"); //Error en la
conexion

    }

}

private void buttonEliminar_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

public void eliminaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "DELETE Operaciones.Detalle_Orden WHERE Num_Orden=" +
idA + "AND Id_Producto=" + idP; //Sentencia en SQL para eliminar tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cerramos conexxion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion

    }

}
}

```

## DevolucionCarrito

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

namespace ProyectoSQLServer
{
    public partial class DevolucionCarrito : Form
    {
        List<Producto> productoList = new List<Producto>();
        List<Carrito> carritoList = new List<Carrito>();
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        int idA = 0; //Variable para id de Cliente para cuando se modificara o eliminara
        long idP = 0;
        int conecta = 0;

        string query ; //Sentencia para mostrar todas la tuplas

        public DevolucionCarrito()
        {
            InitializeComponent();
            conectaBD();
            comboBoxProducto.SelectedIndex = -1;
            dateTimePickerFecha.Value = DateTime.Now;
            textBoxCantidad.Clear();
            //comboBoxProducto.Enabled = false;
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                llenalista();
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                return -1;
            }
        }

        public void muestra()
        {
            string query1 = string.Concat("SELECT C.Id_Carrito, D.Cantidad_Producto,
D.Fecha, CONCAT( P.Nombre_Producto, ' | ',cast(P.Precio_Publico_Producto AS
DECIMAL(10,2))) " +
"FROM Artículo.Producto P,Operaciones.Carrito_Venta C,
Operaciones.Devolucion_Carrito D " +
"WHERE C.Id_Carrito = D.Id_Carrito AND D.Id_Producto = P.Id_Producto");

            SqlCommand command = new SqlCommand(query1, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewDevolucion.DataSource = null;
            dataGridViewDevolucion.DataSource = dataTable; //Llenamos el dataGrid con
los valores de las tuplas
            dataGridViewDevolucion.Columns[0].HeaderText = "Id Carrito";
            dataGridViewDevolucion.Columns[1].HeaderText = "Cantidad";
            dataGridViewDevolucion.Columns[2].HeaderText = "Fecha";
        }
    }
}
```

```

        dataGridViewDevolucion.Columns[3].HeaderText = "Nombre Producto | Precio Publico Producto";
        //dataGridViewDevolucion.Columns[4].HeaderText = "Precio Publico Producto";
        CarritoForm carritoForm =
Application.OpenForms.OfType<CarritoForm>().FirstOrDefault();
        if (carritoForm != null)
            carritoForm.muestra();
        limpia(); //Funcion para limpiar los textbox
    }

    public void limpia()
    {
        comboBoxProducto.SelectedIndex = -1;
        dateTimePickerFecha.Value = DateTime.Now;
        textBoxCantidad.Clear();
    }

    public void llenalista()
    {
        SqlCommand command; //Declaramos variable de SqlCommand que lleva la
        sentencia y la conexion
        SqlDataReader sqlDataReader;
        query = string.Concat("SELECT C.Id_Carrito, C.Id_Tarjeta_Cliente,
C.Forma_De_Pago, C.Fecha_Venta, C.Total_Carrito, U.Nombre_Cliente" +
            " FROM Operaciones.Carrito_Venta C, Datos.Tarjeta_Cliente T,
Usuario.Cliente U " +
            "WHERE T.Id_Cliente = U.Id_Cliente AND T.Id_Tarjeta =
C.Id_Tarjeta_Cliente"); //Sentencia para mostrar todas la tuplas

        command = new SqlCommand(query, conexion); //Declaramos variable de
        SqlCommand que lleva la sentencia y la conexion
        sqlDataReader = command.ExecuteReader();
        //Llenamos lista de carrito
        while (sqlDataReader.Read())
        {
            Carrito carrito = new Carrito();
            carrito.Id_Carrito = sqlDataReader.GetInt64(0);
            carrito.Id_Tarjeta = sqlDataReader.GetInt64(1);
            carrito.FormaPago = sqlDataReader.GetString(2);
            carrito.FechaVenta = sqlDataReader.GetDateTime(3);
            carrito.Total = sqlDataReader.GetDecimal(4);
            carrito.Cliente = sqlDataReader.GetString(5);
            carrito.idynombre = carrito.Id_Carrito + " - " + carrito.Cliente;

            carritoList.Add(carrito);
        }
        sqlDataReader.Close();
        comboBoxCarrito.DataSource = carritoList;
        comboBoxCarrito.DisplayMember = "idynombre";
        comboBoxCarrito.ValueMember = "idynombre";
    }

    private void dataGridViewDevolucion_CellClick(object sender,
DataGridViewCellEventArgs e)
    {
        string g = dataGridViewDevolucion.CurrentRow.Cells[0].Value.ToString();
        string idCa = "";
        for (int i = 0; i < g.Length; i++)
        {
            if (g[i] != ' ')
            {
                idCa += g[i]; //Separamos el id de todo el string
            }
            else
                break;
        }

        idA = int.Parse(idCa); //Indicamos el id de la tupla que seleccionamos
        conexion.Open();
        query = string.Concat("SELECT * "
            + " FROM Artículo.Producto P, Operaciones.Detalle_Carrito O"
            + " WHERE P.Id_Producto = O.Id_Producto AND O.Id_Carrito = " + idA);
        //Sentencia para mostrar todas la tuplas
    }

```

```

        SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
        SqlDataReader sqlDataReader1 = command.ExecuteReader();
        productoList.Clear();
        comboBoxProducto.DataSource = null;
        string q = sqlDataReader1.ToString();

        while (sqlDataReader1.Read())
        {
            Producto producto = new Producto();
            producto.Id_Producto = sqlDataReader1.GetInt64(0);
            producto.Nombre_Producto = sqlDataReader1.GetString(1) + " | " +
Math.Round(sqlDataReader1.GetDecimal(2), 2);

            productoList.Add(producto);
        }

        sqlDataReader1.Close();
        conexion.Close();

        for (int i = 0; i < productoList.Count; i++)
        {
            if (productoList[i].Nombre_Producto ==
dataGridViewDevolucion.CurrentRow.Cells[3].Value.ToString())
            {
                idP = productoList[i].Id_Producto; //Indicamos el di del producto
seleccionado
                break;
            }
        }
        textBoxCantidad.Text =
dataGridViewDevolucion.CurrentRow.Cells[1].Value.ToString(); //Llenamos el textbox de
cantidad con los datos del datagridview
        dateTimePickerFecha.Value =
DateTime.Parse(dataGridViewDevolucion.CurrentRow.Cells[2].Value.ToString());
        for (int i = 0; i < carritoList.Count; i++)
        {
            if (carritoList[i].Id_Carrito == idA)
            {
                int id =
comboBoxCarrito.FindString(carritoList[i].Id_Carrito.ToString());
                comboBoxCarrito.SelectedIndex = id; //Damos valor a el combobox
                break;
            }
        }
        for (int i = 0; i < productoList.Count; i++)
        {
            if (productoList[i].Id_Producto == idP)
            {
                int id =
comboBoxProducto.FindStringExact(productoList[i].Nombre_Producto);
                comboBoxProducto.SelectedIndex = id; //Damos el valor al combobox
                break;
            }
        }
    }

    private void buttonAgregar_Click(object sender, EventArgs e)
    {
        insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
        conectaBD(); //Conectamos a la Base de datos
    }

```



```

public void insertaRegistro()
{
    try
    {
        conexion.Open(); //Abre la conexion
        //"(SELECT Producto FROM Operaciones.Detalle_Carrito DC,
Operaciones.Devolucion_Carrito CD WHERE DC.Id_Carrito = CD.Id_Carrito)"
        string fecha = dateTimePickerFecha.Value.ToString("yyyy-MM-dd");
        string consulta = "INSERT INTO Operaciones.Devolucion_Carrito
(Id_Carrito, Id_Producto, Fecha, Cantidad_Producto) " +
        "VALUES(" + carritoList[comboBoxCarrito.SelectedIndex].Id_Carrito +
",," + productoList[comboBoxProducto.SelectedIndex].Id_Producto + "," + fecha +
",," + textBoxCantidad.Text + ")";
        //Sentencia para insertar alumnos con los datos de las textbox
SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable SqlDataAdapter
command.ExecuteNonQuery(); //Ejecutamos el comando
conexion.Close(); //Cerramos conexion
limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error: " + ex.Message); //Error en la conexion
    }
}

private void buttonModificar_Click(object sender, EventArgs e)
{
    modificaRegistro(); //Mandamos a llamar funcion para modificar la tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void modificaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string fecha = dateTimePickerFecha.Value.ToString("yyyy-MM-dd");
        string consulta = "UPDATE Operaciones.Devolucion_Carrito SET
Id_Carrito=" + carritoList[comboBoxCarrito.SelectedIndex].Id_Carrito + ",Id_Producto="
+ productoList[comboBoxProducto.SelectedIndex].Id_Producto +
",Cantidad_Producto=" + textBoxCantidad.Text + ", Fecha=" + fecha +
" WHERE Id_Carrito=" + idA + " AND Id_Producto=" + idP;
        //Sentencia en SQL para modificar la tupla
SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable de tipo SqlCommand
command.ExecuteNonQuery(); //Ejecutamos la sentencia
conexion.Close(); //Cerramos conexion
limpia(); //Limpiamos los textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error: Numero Duplicado de Tarjeta"); //Error en la
conexion
    }
}

private void buttonEliminar_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

public void eliminaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string fecha = dateTimePickerFecha.Value.ToString("yyyy-MM-dd");
        string consulta = "DELETE Operaciones.Devolucion_Carrito WHERE
Id_Carrito=" + idA + "AND Id_Producto=" + idP + "AND Cantidad_Producto=" +

```

```

        textBoxCantidad.Text + "AND Fecha =" + fecha + "'"; //Sentencia en
        SQL para eliminar tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
        variable tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cerramos conexxion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
        conexion
    }
}

private void comboBoxCarrito_SelectedIndexChanged(object sender, EventArgs e)
{
    if (conecta == 0)
    {
        conecta++;
    }
    else
    {
        comboBoxProducto.Enabled = true;
        conexion.Open();
        string g = comboBoxCarrito.Text;
        string idCa = "";
        for (int i = 0; i < g.Length; i++)
        {
            if (g[i] != ' ')
            {
                idCa += g[i];
            }
            else
            {
                break;
            }
        }

        idA = int.Parse(idCa);
        SqlCommand command; //Declaramos variable de SqlCommand que lleva la
        sentencia y la conexion
        comboBoxProducto.Enabled = true;

        query = string.Concat("SELECT * "
        + " FROM Artículo.Producto P, Operaciones.Detalle_Carrito O"
        + " WHERE P.Id_Producto = O.Id_Producto AND O.Id_Carrito = " + idA);
        //Sentencia para mostrar todas la tuplas

        command = new SqlCommand(query, conexion); //Declaramos variable de
        SqlCommand que lleva la sentencia y la conexion
        SqlDataReader sqlDataReader1 = command.ExecuteReader();
        productoList.Clear();
        comboBoxProducto.DataSource = null;
        string q = sqlDataReader1.ToString();
        while (sqlDataReader1.Read())
        {
            Producto producto = new Producto();
            producto.Id_Producto = sqlDataReader1.GetInt64(0);
            producto.Nombre_Producto = sqlDataReader1.GetString(1) + " | " +
            Math.Round(sqlDataReader1.GetDecimal(2), 2);

            productoList.Add(producto);
        }
        comboBoxProducto.DataSource = productoList;
        comboBoxProducto.DisplayMember = "Nombre_Producto";
        comboBoxProducto.ValueMember = "Id_Producto";

        sqlDataReader1.Close();
        conexion.Close();
    }
}
}
}

```

## Orden

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

namespace ProyectoSQLServer
{
    public partial class OrdenForm : Form
    {
        List<Proveedor> proveedorList = new List<Proveedor>();
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        int idA = 0; //Variable para id de Cliente para cuando se modificara o eliminara
        int idP = 0;

        public OrdenForm()
        {
            InitializeComponent();
            textBoxTotal.Text = "0";
            conexion.Open(); //Abrimos conexion con la base de datos
            llenalista(); //Funcion que muestra en el datagridview las tuplas
            conexion.Close();
            conectaBD();
            comboBoxProveedor.SelectedIndex = -1;
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                return -1;
            }
        }

        public void muestra()
        {
            string query1 = string.Concat("SELECT O.Num_Orden,
O.Fecha_Orden,cast(O.Total AS DECIMAL(10,2)), CONCAT(P.Nombre_Proveedor, ' | ' " +
" ,P.Correo_Electronico) FROM Usuario.Proveedor P, Operaciones.Orden O
WHERE P.Id_Proveedor=O.Id_Proveedor");

            SqlCommand command = new SqlCommand(query1, conexion); //Declaramos variable
            de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
            SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewOrden.DataSource = null;
            dataGridViewOrden.DataSource = dataTable; //Llenamos el dataGrid con los
            valores de las tuplas
            dataGridViewOrden.Columns[0].HeaderText = "Número de Orden";
            dataGridViewOrden.Columns[1].HeaderText = "Fecha de Orden";
            dataGridViewOrden.Columns[2].HeaderText = "Total";
            dataGridViewOrden.Columns[3].HeaderText = "Nombre Proveedor | Correo
Electrónico";
            //dataGridViewOrden.Columns[4].HeaderText = "Correo Electrónico";
            limpia(); //Funcion para limpiar los textbox
        }
    }
}
```

```

public void limpia()
{
    dateTimePickerFechaOrden.Value = DateTime.Now; //
    textBoxTotal.Text = "0";
    comboBoxProveedor.SelectedIndex = -1; //
}

public void llenalista()
{
    string query = string.Concat("SELECT * FROM Usuario.Proveedor"); //Sentencia
para mostrar todas la tuplas

    SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
    SqlDataReader sqlDataReader = command.ExecuteReader();
    //Llenamos la lista de proveedor
    while (sqlDataReader.Read())
    {
        Proveedor proveedor = new Proveedor();
        proveedor.Id_Proveedor = sqlDataReader.GetInt64(0);
        proveedor.Nombre_Proveedor = sqlDataReader.GetString(2) + " | " +
sqlDataReader.GetString(1);
        //proveedor.Correo_Electronico = sqlDataReader.GetString(1) + " | " ;
        proveedor.Domicilio_Fiscal = sqlDataReader.GetString(3);

        proveedorList.Add(proveedor);
    }
    comboBoxProveedor.DataSource = proveedorList;
    comboBoxProveedor.DisplayMember = "Nombre_Proveedor";
    comboBoxProveedor.ValueMember = "Id_Proveedor";
}

private void dataGridViewOrden_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    idA = Convert.ToInt32(dataGridViewOrden.CurrentRow.Cells[0].Value);
//Indicamos el id de la tupla que seleccionamos
    if
(decimal.Round(Convert.ToDecimal(dataGridViewOrden.CurrentRow.Cells[2].Value),
2).ToString() != "BDNull")
        textBoxTotal.Text =
decimal.Round(Convert.ToDecimal(dataGridViewOrden.CurrentRow.Cells[2].Value),
2).ToString();
    else
        textBoxTotal.Text = ""; //Llenamos el textbox de total con los datos del
datagridview
        dateTimePickerFechaOrden.Value =
DateTime.Parse(dataGridViewOrden.CurrentRow.Cells[1].Value.ToString());
        for (int i = 0; i < proveedorList.Count; i++)
        {
            if (dataGridViewOrden.CurrentRow.Cells[3].Value.ToString() ==
proveedorList[i].Nombre_Proveedor)
            {
                idP =
comboBoxProveedor.FindStringExact(proveedorList[i].Nombre_Proveedor);
                comboBoxProveedor.SelectedIndex = idP;
                break;
            }
        }
}

private void buttonAgregar_Click(object sender, EventArgs e)
{
    insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
    conectaBD(); //Conectamos a la Base de datos
}

```

```

public void insertaRegistro()
{
    try
    {
        conexion.Open(); //Abre la conexion
        string fecha = dateTimePickerFechaOrden.Value.ToString("yyyy-MM-dd");
        string consulta = "INSERT INTO Operaciones.Orden" + "(Id_Proveedor,
Fecha_Orden, Total) " +
            "VALUES (" +
proveedorList[comboBoxProveedor.SelectedIndex].Id_Proveedor + "','" + fecha + "','" +
int.Parse(textBoxTotal.Text) + ")";
        //Sentencia para insertar alumnos con los datos de las textbox
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable SqlDataAdapter
        command.ExecuteNonQuery(); //Ejecutamos el comando
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}

private void buttonModificar_Click(object sender, EventArgs e)
{
    modificaRegistro(); //Mandamos a llamar funcion para modificar la tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void modificaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string fecha = dateTimePickerFechaOrden.Value.ToString("yyyy-MM-dd");
        string consulta = "UPDATE Operaciones.Orden SET Id_Proveedor=" +
proveedorList[comboBoxProveedor.SelectedIndex].Id_Proveedor + ",Total=" +
        + textBoxTotal.Text + ",Fecha_Orden='" + fecha + "' WHERE
Num_Orden=" + idA;
        //Sentencia en SQL para modificar la tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable de tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos la sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos los textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}

private void buttonEliminar_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

```

```

        public void eliminaRegistro()
        {
            try
            {
                conexion.Open(); //Abrimos conexion
                string consulta = "DELETE Operaciones.Orden WHERE Num_Orden=" + idA;
                //Sentencia en SQL para eliminar tupla
                SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
                variable tipo SqlCommand
                command.ExecuteNonQuery(); //Ejecutamos sentencia
                conexion.Close(); //Cerramos conexion
                limpia(); //Limpiamos textbox
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos conexxion
                MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
                conexion
            }
        }

        private void buttonDetalle_Click(object sender, EventArgs e)
        {
            DetalleOrdenForm detalleOrdenForm = new DetalleOrdenForm();
            detalleOrdenForm.ShowDialog();
        }
    }
}

```

## Producto

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

namespace ProyectoSQLServer
{
    public partial class ProductoForm : Form
    {
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        int idA = 0; //Variable para id de Producto para cuando se modificara o
eliminarla

        public ProductoForm()
        {
            InitializeComponent();
            conectaBD();
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                return -1;
            }
        }

        public void muestra()
        {
            string query = string.Concat("SELECT
Id_Producto,Nombre_Producto,cast(Precio_Publico_Producto AS DECIMAL(10,2)),
cast(Precio_Publico_Proveedor AS DECIMAL(10,2)), " +
"Color, Stock FROM Articulo.Producto"); //Sentencia para mostrar todas
la tuplas

            SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewProducto.DataSource = null;
            dataGridViewProducto.DataSource = dataTable; //Llenamos el dataGrid con los
valores de las tuplasdataGridViewProveedor.Columns[0].HeaderText = "Id Proveedor";
            dataGridViewProducto.Columns[0].HeaderText = "Id Producto";
            dataGridViewProducto.Columns[1].HeaderText = "Nombre Producto";
            dataGridViewProducto.Columns[2].HeaderText = "Precio Público Producto";
            dataGridViewProducto.Columns[3].HeaderText = "Precio Público Proveedor";
            dataGridViewProducto.Columns[4].HeaderText = "Color";
            dataGridViewProducto.Columns[5].HeaderText = "Stock";
            //comboBoxCarrera.
            limpiar(); //Funcion para limpiar los textbox
        }
    }
}
```

```

public void limpia()
{
    textBoxNombre.Clear(); //Limpiamos textbox de Nombre
    textBoxPrecioPublico.Clear(); //Limpiamos textbox
    textBoxPrecioProveedor.Clear();
    textBoxColor.Clear();
    textBoxStock.Clear();
}

private void dataGridViewProducto_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    idA = Convert.ToInt32(dataGridViewProducto.CurrentRow.Cells[0].Value);
    //Indicamos el id de la tupla que seleccionamos
    textBoxNombre.Text =
dataGridViewProducto.CurrentRow.Cells[1].Value.ToString(); //Llenamos el textbox de
nombre con los datos del datagridview
    textBoxPrecioPublico.Text =
(decimal.Round(Convert.ToDecimal(dataGridViewProducto.CurrentRow.Cells[2].Value),
2)).ToString();
    textBoxPrecioProveedor.Text =
(decimal.Round(Convert.ToDecimal(dataGridViewProducto.CurrentRow.Cells[3].Value),
2)).ToString();
    textBoxColor.Text =
dataGridViewProducto.CurrentRow.Cells[4].Value.ToString();
    textBoxStock.Text =
dataGridViewProducto.CurrentRow.Cells[5].Value.ToString();
}

private void buttonAgregar_Click(object sender, EventArgs e)
{
    insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void insertaRegistro()
{
    try
    {
        conexion.Open(); //Abre la conexion
        string consulta = "INSERT INTO Articulo.Producto" +
            "(Nombre_Producto, Precio_Publico_Producto,
Precio_Publico_Proveedor,Color,Stock) " +
            "VALUES('" + textBoxNombre.Text + "','" + textBoxPrecioPublico.Text
+ "','" + textBoxPrecioProveedor.Text
            + "','" + textBoxColor.Text + "','" + textBoxStock.Text + ")";
        //Sentencia para insertar alumnos con los datos de las textbox
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable SqlDataAdapter
        command.ExecuteNonQuery(); //Ejecutamos el comando
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}

private void buttonModificar_Click(object sender, EventArgs e)
{
    modificaRegistro(); //Mandamos a llamar funcion para modificar la tupla
    conectaBD(); //Conectamos a la Base de datos
}

```



```

public void modificaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string s = textBoxStock.Text;
        string consulta = "UPDATE Artículo.Producto SET Nombre_Producto='" +
textBoxNombre.Text + "',Precio_Publico_Producto='" +
        + textBoxPrecioPublico.Text + "',Precio_Publico_Proveedor='" +
textBoxPrecioProveedor.Text
        + "',Color='" + textBoxColor.Text + "',Stock='" + textBoxStock.Text +
" WHERE Id_Producto=" + idA;
        //Sentencia en SQL para modificar la tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable de tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos la sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos los textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}

private void buttonEliminar_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

public void eliminaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "DELETE Artículo.Producto WHERE Id_Producto=" + idA;
//Sentencia en SQL para eliminar tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cerramos conexxion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}
}
}

```

## Proveedor

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

namespace ProyectoSQLServer
{
    public partial class ProveedorForm : Form
    {
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        int idA = 0; //Variable para id de Proveedor para cuando se modificara o
eliminar

        public ProveedorForm()
        {
            InitializeComponent();
            conectaBD();
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                return -1;
            }
        }

        public void muestra()
        {
            string query = string.Concat("SELECT * FROM Usuario.Proveedor"); //Sentencia
para mostrar todas la tuplas

            SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewProveedor.DataSource = null;
            dataGridViewProveedor.DataSource = dataTable; //Llenamos el dataGrid con los
valores de las tuplas
            dataGridViewProveedor.Columns[0].HeaderText = "Id Proveedor";
            dataGridViewProveedor.Columns[1].HeaderText = "Correo Electrónico";
            dataGridViewProveedor.Columns[2].HeaderText = "Nombre Proveedor";
            dataGridViewProveedor.Columns[3].HeaderText = "Domicilio Fiscal";
            limpia(); //Funcion para limpiar los textbox
        }

        public void limpia()
        {
            textBoxNombre.Clear(); //Limpiamos textbox de Nombre
            textBoxCorreo.Clear(); //Limpiamos textbox de Correo
            textBoxDomicilio.Clear();
        }
    }
}
```

```

        private void dataGridViewProveedor_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            idA = Convert.ToInt32(dataGridViewProveedor.CurrentRow.Cells[0].Value);
            //Indicamos el id de la tupla que seleccionamos
            textBoxNombre.Text =
            dataGridViewProveedor.CurrentRow.Cells[2].Value.ToString(); //Llenamos el textbox de
            nombre con los datos del datagridview
            textBoxCorreo.Text =
            dataGridViewProveedor.CurrentRow.Cells[1].Value.ToString(); //Llenamos el textbox de
            correo con los datos del datagridview
            textBoxDomicilio.Text =
            dataGridViewProveedor.CurrentRow.Cells[3].Value.ToString();
        }

        private void buttonAgregar_Click(object sender, EventArgs e)
        {
            insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
            conectaBD(); //Conectamos a la Base de datos
        }

        public void insertaRegistro()
        {
            try
            {
                conexion.Open(); //Abre la conexion
                string consulta = "INSERT INTO Usuario.Proveedor" +
                "(Correo_Electronico, Nombre_Proveedor, Domicilio_Fiscal) " +
                "VALUES(' " + textBoxCorreo.Text + "', " + textBoxNombre.Text +
                "', '" + textBoxDomicilio.Text + "')";
                //Sentencia para insertar alumnos con los datos de las textbox
                SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
                variable SqlDataAdapter
                command.ExecuteNonQuery(); //Ejecutamos el comando
                conexion.Close(); //Cerramos conexion
                limpia(); //Limpiamos textbox
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cierra conexion
                MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
                conexion
            }
        }

        private void buttonModificar_Click(object sender, EventArgs e)
        {
            modificaRegistro(); //Mandamos a llamar funcion para modificar la tupla
            conectaBD(); //Conectamos a la Base de datos
        }

        public void modificaRegistro()
        {
            try
            {
                conexion.Open(); //Abrimos conexion
                string consulta = "UPDATE Usuario.Proveedor SET Correo_Electronico=' " +
                textBoxCorreo.Text + "', Nombre_Proveedor=' " +
                textBoxNombre.Text + "', Domicilio_Fiscal=' " +
                textBoxDomicilio.Text + "' WHERE Id_Proveedor=" + idA;
                //Sentencia en SQL para modificar la tupla
                SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
                variable de tipo SqlCommand
                command.ExecuteNonQuery(); //Ejecutamos la sentencia
                conexion.Close(); //Cerramos conexion
                limpia(); //Limpiamos los textbox
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cierra conexion
                MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
                conexion
            }
        }
    }

```

```

private void buttonEliminar_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

public void eliminaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "DELETE Usuario.Proveedor WHERE Id_Proveedor=" + idA;
//Sentencia en SQL para eliminar tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cerramos conexxion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}

private void buttonTelefono_Click(object sender, EventArgs e)
{
    TelefonoProveedorForm telefonoProveedorForm = new TelefonoProveedorForm();
    telefonoProveedorForm.ShowDialog();
}
}

```

## TarjetaCliente

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

namespace ProyectoSQLServer
{
    public partial class Tarjeta_Cliente : Form
    {
        List<Cliente> clienteList = new List<Cliente>();
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        int idA = 0; //Variable para id de Cliente para cuando se modificara o eliminara

        public Tarjeta_Cliente()
        {
            InitializeComponent();
            conectaBD();
            comboBoxCliente.SelectedIndex = -1;
            dateTimePickerFecha.MinDate = new DateTime(2023, 6, 1);
            dateTimePickerFecha.Value = new DateTime(2024, 1, 1);
            dateTimePickerFecha.Format = DateTimePickerFormat.Custom;
            dateTimePickerFecha.CustomFormat = "MM/yy";
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                llenalista();
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                MessageBox.Show(ex.Message);
                return -1;
            }
        }

        public void muestra()
        {
            string query1 = string.Concat("SELECT T.Id_Cliente,
T.Numero_Tarjeta,T.CVV,T.Banco, FORMAT(Fecha,'MM/yy'),CONCAT(C.Nombre_Cliente, ' | ',
C.Correo_Electronico) FROM Usuario.Cliente C, " +
"Datos.Tarjeta_Cliente T WHERE C.Id_Cliente=T.Id_Cliente");

            SqlCommand command = new SqlCommand(query1, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewTarjetaCliente.DataSource = null;
            //dataTable.Rows.RemoveAt(-1);
            dataGridViewTarjetaCliente.DataSource = dataTable; //Llenamos el dataGrid
con los valores de las tuplas
            dataGridViewTarjetaCliente.Columns[0].HeaderText = "Id Cliente";
            dataGridViewTarjetaCliente.Columns[1].HeaderText = "Número Tarjeta";
            dataGridViewTarjetaCliente.Columns[2].HeaderText = "CVV";
            dataGridViewTarjetaCliente.Columns[3].HeaderText = "Banco";
            dataGridViewTarjetaCliente.Columns[4].HeaderText = "Fecha";
            dataGridViewTarjetaCliente.Columns[5].HeaderText = "Nombre Cliente | Correo
Electrónico";
        }
    }
}
```

```

        //dataGridViewTarjetaCliente.Columns[6].HeaderText = "Correo Electrónico";
        limpia(); //Funcion para limpiar los textbox
    }

    public void limpia()
    {
        textBoxNumTarjeta.Clear();
        textBoxCVV.Clear();
        textBoxBanco.Clear();
        dateTimePickerFecha.Value = new DateTime(2024, 1, 1);
        comboBoxCliente.SelectedIndex = -1; //Limpiamos textbox de Correo
    }

    public void llenalista()
    {
        string query = string.Concat("SELECT * FROM Usuario.Cliente"); //Sentencia
        para mostrar todas la tuplas

        SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
        de SqlCommand que lleva la sentencia y la conexion
        SqlDataReader sqlDataReader = command.ExecuteReader();
        while (sqlDataReader.Read())
        {
            Cliente cliente = new Cliente();
            cliente.Id_Cliente = sqlDataReader.GetInt64(0);
            cliente.Nombre_Cliente = sqlDataReader.GetString(1) + " | " +
            sqlDataReader.GetString(2);
            //cliente.Correo_Electronico = sqlDataReader.GetString(2);

            clienteList.Add(cliente);
        }
        comboBoxCliente.DataSource = clienteList;
        comboBoxCliente.DisplayMember = "Nombre_Cliente";
        comboBoxCliente.ValueMember = "Id_Cliente";
    }

    private void dataGridViewTarjetaCliente_CellClick(object sender,
    DataGridViewCellEventArgs e)
    {
        idA = Convert.ToInt32(dataGridViewTarjetaCliente.CurrentRow.Cells[0].Value);
        //Indicamos el id de la tupla que seleccionamos
        textBoxNumTarjeta.Text =
        dataGridViewTarjetaCliente.CurrentRow.Cells[1].Value.ToString(); //Llenamos el textbox
        de nombre con los datos del datagridview
        textBoxCVV.Text =
        dataGridViewTarjetaCliente.CurrentRow.Cells[2].Value.ToString();
        textBoxBanco.Text =
        dataGridViewTarjetaCliente.CurrentRow.Cells[3].Value.ToString();
        dateTimePickerFecha.Value = DateTime.Parse("01/" +
        dataGridViewTarjetaCliente.CurrentRow.Cells[4].Value.ToString());
        for (int i = 0; i < clienteList.Count; i++)
        {
            if (clienteList[i].Id_Cliente == idA)
            {
                int id =
                comboBoxCliente.FindStringExact(clienteList[i].Nombre_Cliente);
                comboBoxCliente.SelectedIndex = id;
            }
        }
    }
}

```

```

private void buttonAgregar_Click(object sender, EventArgs e)
{
    insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void insertaRegistro()
{
    try
    {
        conexion.Open(); //Abre la conexion
        string fecha = dateTimePickerFecha.Value.ToString("yyyy-MM-dd");
        string consulta = "INSERT INTO Datos.Tarjeta_Cliente" + "(Id_Cliente,
Numero_Tarjeta, CVV, Banco, Fecha) " +
            "VALUES(" + clienteList[comboBoxCliente.SelectedIndex].Id_Cliente +
            ", '" + textBoxNumTarjeta.Text + "', '" + textBoxCVV.Text + "', '" +
            textBoxBanco.Text + "', '" + fecha + "')";
        //Sentencia para insertar alumnos con los datos de las textbox
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable SqlDataAdapter
        command.ExecuteNonQuery(); //Ejecutamos el comando
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error: Numero Duplicado de Tarjeta"); //Error en la
conexion
    }
}

private void buttonModificar_Click(object sender, EventArgs e)
{
    modificaRegistro(); //Mandamos a llamar funcion para modificar la tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void modificaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string fecha = dateTimePickerFecha.Value.ToString("yyyy-MM-dd");
        string consulta = "UPDATE Datos.Tarjeta_Cliente SET Id_Cliente=" +
clienteList[comboBoxCliente.SelectedIndex].Id_Cliente + ", Numero_Tarjeta=" +
        + textBoxNumTarjeta.Text + ", CVV=" + textBoxCVV.Text + ", Banco=" +
+ textBoxBanco.Text + ", Fecha=" + fecha + " WHERE Id_Cliente=" + idA;
        //Sentencia en SQL para modificar la tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable de tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos la sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos los textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error: Numero Duplicado de Tarjeta"); //Error en la
conexion
    }
}

```

```

private void buttonEliminar_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

public void eliminaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "DELETE Datos.Tarjeta_Cliente WHERE Id_Cliente=" + idA
+ "AND Numero_Tarjeta=" + textBoxNumTarjeta.Text; //Sentencia en SQL para eliminar tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
        variable tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cerramos conexxion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}
}
}

```



## TelefonoCliente

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

namespace ProyectoSQLServer
{
    public partial class TelefonoClienteForm : Form
    {
        List<Cliente> clienteList = new List<Cliente>();
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        int idA = 0; //Variable para id de Cliente para cuando se modificara o eliminara

        public TelefonoClienteForm()
        {
            InitializeComponent();
            conectaBD();
            comboBoxCliente.SelectedIndex = -1;
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                llenalista();
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                return -1;
            }
        }

        public void muestra()
        {
            string query = string.Concat("SELECT * FROM Datos.Telefono_Cliente");
            //Sentencia para mostrar todas la tuplas
            string query1 = string.Concat("SELECT T.Id_Cliente,
T.Numero,CONCAT(C.Nombre_Cliente, ' | ' ,C.Correo_Electronico) FROM Usuario.Cliente C, "
+
            "Datos.Telefono_Cliente T WHERE C.Id_Cliente=T.Id_Cliente");

            SqlCommand command = new SqlCommand(query1, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewTelefonoC.DataSource = null;
            //dataTable.Rows.RemoveAt(-1);
            dataGridViewTelefonoC.DataSource = dataTable; //Llenamos el dataGrid con
los valores de las tuplas
            dataGridViewTelefonoC.Columns[0].HeaderText = "Id Cliente";
            dataGridViewTelefonoC.Columns[1].HeaderText = "Número";
            dataGridViewTelefonoC.Columns[2].HeaderText = "Nombre Cliente | Correo
Electrónico";
            //dataGridViewTelefonoC.Columns[3].HeaderText = "Correo Electrónico";
            limpia(); //Funcion para limpiar los textbox
        }
    }
}
```

```

public void limpia()
{
    textBoxNumero.Clear(); //Limpiamos textbox de Nombre
    comboBoxCliente.SelectedIndex = -1; //Limpiamos textbox de Correo
}

public void llenalista()
{
    string query = string.Concat("SELECT * FROM Usuario.Cliente"); //Sentencia
para mostrar todas la tuplas

    SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
    SqlDataReader sqlDataReader = command.ExecuteReader();
    while (sqlDataReader.Read())
    {
        Cliente cliente = new Cliente();
        cliente.Id_Cliente = sqlDataReader.GetInt64(0);
        cliente.Nombre_Cliente= sqlDataReader.GetString(1) + " | " +
sqlDataReader.GetString(2);
        //cliente.Correo_Electronico = sqlDataReader.GetString(2);

        clienteList.Add(cliente);
    }
    comboBoxCliente.DataSource = clienteList;
    comboBoxCliente.DisplayMember = "Nombre_Cliente";
    comboBoxCliente.ValueMember = "Id_Cliente";
}

private void dataGridViewTelefonoC_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    idA = Convert.ToInt32(dataGridViewTelefonoC.CurrentRow.Cells[0].Value);
//Indicamos el id de la tupla que seleccionamos
    textBoxNumero.Text =
dataGridViewTelefonoC.CurrentRow.Cells[1].Value.ToString(); //Llenamos el textbox de
nombre con los datos del datagridview
    //comboBoxCliente.SelectedIndex =
dataGridViewTelefonoC.CurrentRow.Cells[3].Value.ToString(); //Llenamos el textbox de
correo con los datos del datagridview
    for (int i = 0; i < clienteList.Count;i++)
    {
        if (clienteList[i].Id_Cliente == idA)
        {
            int id =
comboBoxCliente.FindStringExact(clienteList[i].Nombre_Cliente);
            comboBoxCliente.SelectedIndex = id;
        }
    }
}

private void buttonAgregar_Click(object sender, EventArgs e)
{
    insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
conectaBD(); //Conectamos a la Base de datos
}

public void insertaRegistro()
{
    try
    {
        conexion.Open(); //Abre la conexion
        string consulta = "INSERT INTO Datos.Telefono_Cliente" + "(Id_Cliente,
Numero) " +
            "VALUES(" + clienteList[comboBoxCliente.SelectedIndex].Id_Cliente +
",'" + textBoxNumero.Text + "')";
        //Sentencia para insertar alumnos con los datos de las textbox
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable SqlDataAdapter
        command.ExecuteNonQuery(); //Ejecutamos el comando
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)

```

```

        {
            conexion.Close(); //Cierra conexion
            MessageBox.Show("Error: Numero duplicado"); //Error en la conexion
        }
    }

private void buttonModificar_Click(object sender, EventArgs e)
{
    modificaRegistro(); //Mandamos a llamar funcion para modificar la tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void modificaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "UPDATE Datos.Telefono_Cliente SET Id_Cliente=" +
            clienteList[comboBoxCliente.SelectedIndex].Id_Cliente + ",Numero='" +
            + textBoxNumero.Text + "' WHERE Id_Cliente=" + idA;
        //Sentencia en SQL para modificar la tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
        variable de tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos la sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos los textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error: Numero duplicado"); //Error en la conexion
    }
}

private void buttonElimina_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

public void eliminaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "DELETE Datos.Telefono_Cliente WHERE Id_Cliente=" +
            idA + "AND Numero=" + textBoxNumero.Text; //Sentencia en SQL para eliminar tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
        variable tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cerramos conexxion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
        conexion
    }
}
}
}
}

```

## TelefonoProveedor

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

namespace ProyectoSQLServer
{
    public partial class TelefonoProveedorForm : Form
    {
        List<Proveedor> proveedores = new List<Proveedor>();
        SqlConnection conexion = new SqlConnection("Server=JETERMEDINA\\SQLEXPRESS;" +
"Database=ComprasEnLinea;" + "Integrated Security=true;");
        //Variable para la conexion con la base de datos
        int idA = 0; //Variable para id de Cliente para cuando se modificara o eliminara

        public TelefonoProveedorForm()
        {
            InitializeComponent();
            conectaBD();
            comboBoxProveedor.SelectedIndex = -1;
        }

        public int conectaBD()
        {
            try
            {
                conexion.Open(); //Abrimos conexion con la base de datos
                muestra(); //Funcion que muestra en el datagridview las tuplas
                llenalista();
                conexion.Close(); //Cerramos base de datos
                return 0;
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cerramos base de datos
                return -1;
            }
        }

        public void muestra()
        {
            string query1 = string.Concat("SELECT T.Id_Proveedor, T.Numero,
CONCAT(P.Nombre_Proveedor, ' | ' ,P.Correo_Electronico) FROM Usuario.Proveedor P, " +
"Datos.Telefono_Proveedor T WHERE P.Id_Proveedor=T.Id_Proveedor");

            SqlCommand command = new SqlCommand(query1, conexion); //Declaramos variable
de SqlCommand que lleva la sentencia y la conexion
            SqlDataAdapter adapter = new SqlDataAdapter(command); //Declaramos variable
SqlDataAdapter
            DataTable dataTable = new DataTable(); // Declaramos variable de DataTable
            adapter.Fill(dataTable); // Agrega renglones en determinado rango
            dataGridViewTelefonoP.DataSource = null;
            dataGridViewTelefonoP.DataSource = dataTable; //Llenamos el dataGrid con los
valores de las tuplas
            dataGridViewTelefonoP.Columns[0].HeaderText = "Id Proveedor";
            dataGridViewTelefonoP.Columns[1].HeaderText = "Número";
            dataGridViewTelefonoP.Columns[2].HeaderText = "Nombre Proveedor | Correo
Electrónico";
            //dataGridViewTelefonoP.Columns[3].HeaderText = "Correo Electrónico";
            limpia(); //Funcion para limpiar los textbox
        }

        public void limpia()
        {
            textBoxNumero.Clear(); //Limpiamos textbox de Nombre
            comboBoxProveedor.SelectedIndex = -1; //Limpiamos textbox de Correo
        }
    }
}
```

```

        public void llenalista()
        {
            string query = string.Concat("SELECT * FROM Usuario.Proveedor"); //Sentencia
            para mostrar todas la tuplas

            SqlCommand command = new SqlCommand(query, conexion); //Declaramos variable
            de SqlCommand que lleva la sentencia y la conexion
            SqlDataReader sqlDataReader = command.ExecuteReader();
            while (sqlDataReader.Read())
            {
                Proveedor proveedor = new Proveedor();
                proveedor.Id_Proveedor = sqlDataReader.GetInt64(0);
                proveedor.Nombre_Proveedor = sqlDataReader.GetString(2) + " | " +
                sqlDataReader.GetString(1);
                //proveedor.Correo_Electronico = sqlDataReader.GetString(1);
                proveedor.Domicilio_Fiscal = sqlDataReader.GetString(3);

                proveedores.Add(proveedor);
            }
            comboBoxProveedor.DataSource = proveedores;
            comboBoxProveedor.DisplayMember = "Nombre_Proveedor";
            comboBoxProveedor.ValueMember = "Id_Proveedor";
        }

        private void dataGridViewTelefonoP_CellClick(object sender,
        DataGridViewCellEventArgs e)
        {
            idA = Convert.ToInt32(dataGridViewTelefonoP.CurrentRow.Cells[0].Value);
            //Indicamos el id de la tupla que seleccionamos
            textBoxNumero.Text =
            dataGridViewTelefonoP.CurrentRow.Cells[1].Value.ToString(); //Llenamos el textbox de
            nombre con los datos del datagridview
            //comboBoxCliente.SelectedIndex =
            dataGridViewTelefonoC.CurrentRow.Cells[3].Value.ToString(); //Llenamos el textbox de
            correo con los datos del datagridview
            for (int i = 0; i < proveedores.Count; i++)
            {
                if (proveedores[i].Id_Proveedor == idA)
                {
                    int id =
                    comboBoxProveedor.FindStringExact(proveedores[i].Nombre_Proveedor);
                    comboBoxProveedor.SelectedIndex = id;
                    break;
                }
            }
        }

        private void buttonAgregar_Click(object sender, EventArgs e)
        {
            insertaRegistro(); //Mandamos a llamar funcion para insertar tupla
            conectaBD(); //Conectamos a la Base de datos
        }

        public void insertaRegistro()
        {
            try
            {
                conexion.Open(); //Abre la conexion
                string consulta = "INSERT INTO Datos.Telefono_Proveedor" +
                "(Id_Proveedor, Numero) " +
                "VALUES(" + proveedores[comboBoxProveedor.SelectedIndex].Id_Proveedor
                + "," + textBoxNumero.Text + ")";
                //Sentencia para insertar alumnos con los datos de las textbox
                SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
                variable SqlDataAdapter
                command.ExecuteNonQuery(); //Ejecutamos el comando
                conexion.Close(); //Cerramos conexion
                limpia(); //Limpiamos textbox
            }
            catch (Exception ex)
            {
                conexion.Close(); //Cierra conexion
                MessageBox.Show("Error: Numero duplicado"); //Error en la conexion
            }
        }
    }

```

```

private void buttonModificar_Click(object sender, EventArgs e)
{
    modificaRegistro();//Mandamos a llamar funcion para modificar la tupla
    conectaBD(); //Conectamos a la Base de datos
}

public void modificaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "UPDATE Datos.Telefono_Proveedor SET Id_Proveedor=" +
proveedores[comboBoxProveedor.SelectedIndex].Id_Proveedor + ",Numero=" +
        + textBoxNumero.Text + "' WHERE Id_Proveedor=" + idA;
        //Sentencia en SQL para modificar la tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable de tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos la sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos los textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cierra conexion
        MessageBox.Show("Error: Numero duplicado"); //Error en la conexion
    }
}

private void buttonEliminar_Click(object sender, EventArgs e)
{
    eliminaRegistro(); //Llamamos a la funcion que eliminar la tupla
    conectaBD(); //Conectamos a la Base de Datos
}

public void eliminaRegistro()
{
    try
    {
        conexion.Open(); //Abrimos conexion
        string consulta = "DELETE Datos.Telefono_Proveedor WHERE Id_Proveedor="
+ idA + "AND Numero=" + textBoxNumero.Text; //Sentencia en SQL para eliminar tupla
        SqlCommand command = new SqlCommand(consulta, conexion); //Declaramos
variable tipo SqlCommand
        command.ExecuteNonQuery(); //Ejecutamos sentencia
        conexion.Close(); //Cerramos conexion
        limpia(); //Limpiamos textbox
    }
    catch (Exception ex)
    {
        conexion.Close(); //Cerramos conexxion
        MessageBox.Show("Error de conexion: " + ex.Message); //Error en la
conexion
    }
}
}
}

```