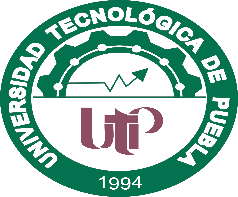
****

**UNIVERISDAD TECNOLOGICA DE PUEBLA**

**TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN TSU S.I**

**DESARROLLO DE APLICACIONES 1**

**DOCUMENTACION PROYECTO**

**ROMERO ROMAN EDGAR**

**FLORES RAMIREZ EDUARDO**

**ESPINOZA CASTRO GUSTAVO**

**3”F”**

**Planteamiento para la BASE DE DATOS de un centro distribuidor de productos (CEDIS)**

Se requiere saber el número de artículos que se envían a cada cliente, así como las entradas de los productos al centro distribuidor, a su vez también es necesario conocer cuál es el total de las ventas diarias, el precio de venta de los artículos, por cada cliente.

Se desea que al momento de la compra se otorgue una factura o un ticket al cliente, donde se almacene la fecha y así poder realizar los inventarios.

Es necesario llevar un control de los diferentes tipos de pago para dar razón a la empresa que envía los productos.

Se deb0e tomar en cuenta que algunos productos se les aplica descuentos el cual varía de acuerdo a la temporada del año.

Para los productos se debe almacenar:

* Id producto
* Nombre
* Marca
* Piezas
* Gramos
* Sabor

Para los clientes:

* Id cliente
* Nombre (Empresas y micro-empresa)
* Dirección
* Teléfono

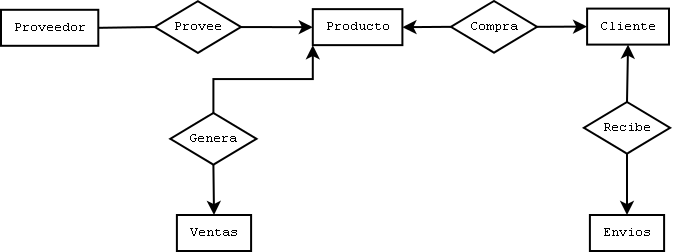
Para el proveedor almacenar:

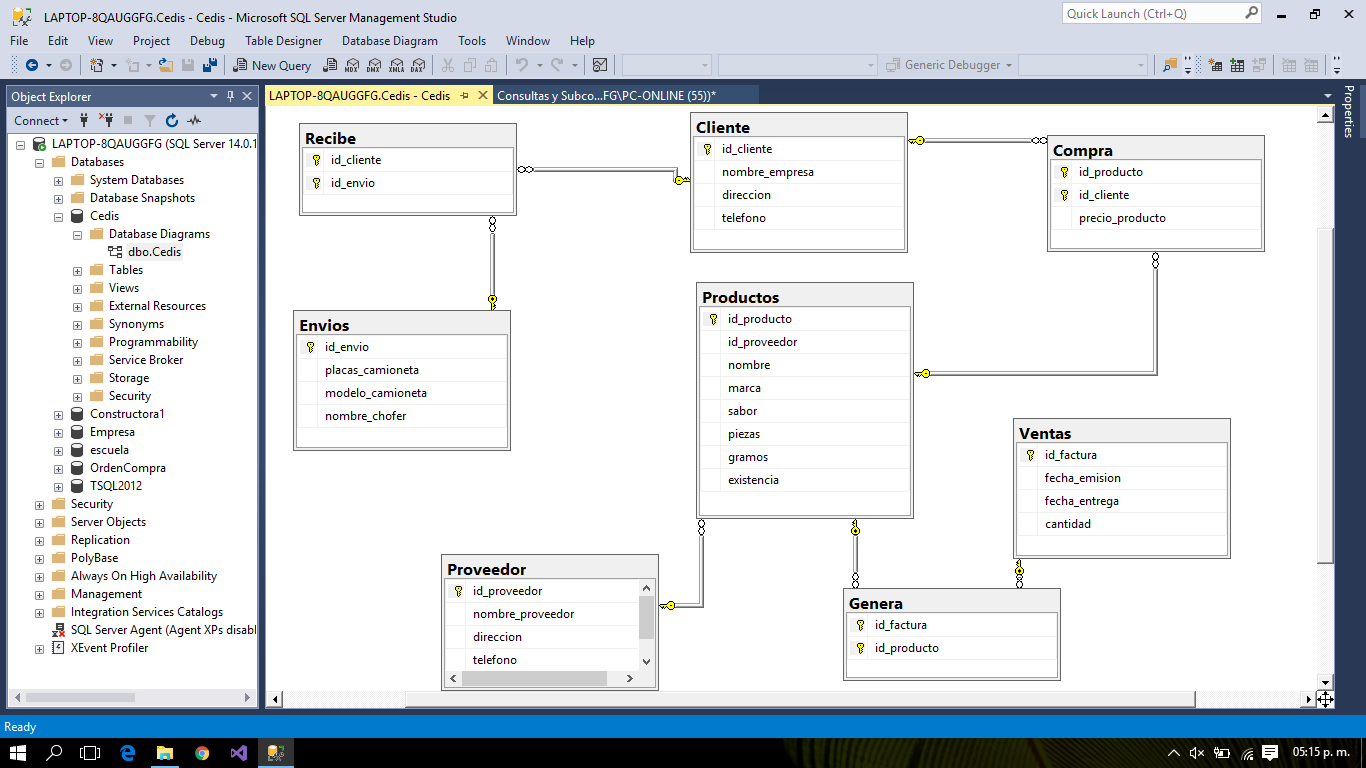
* Id proveedor
* Nombre
* Dirección
* Teléfono

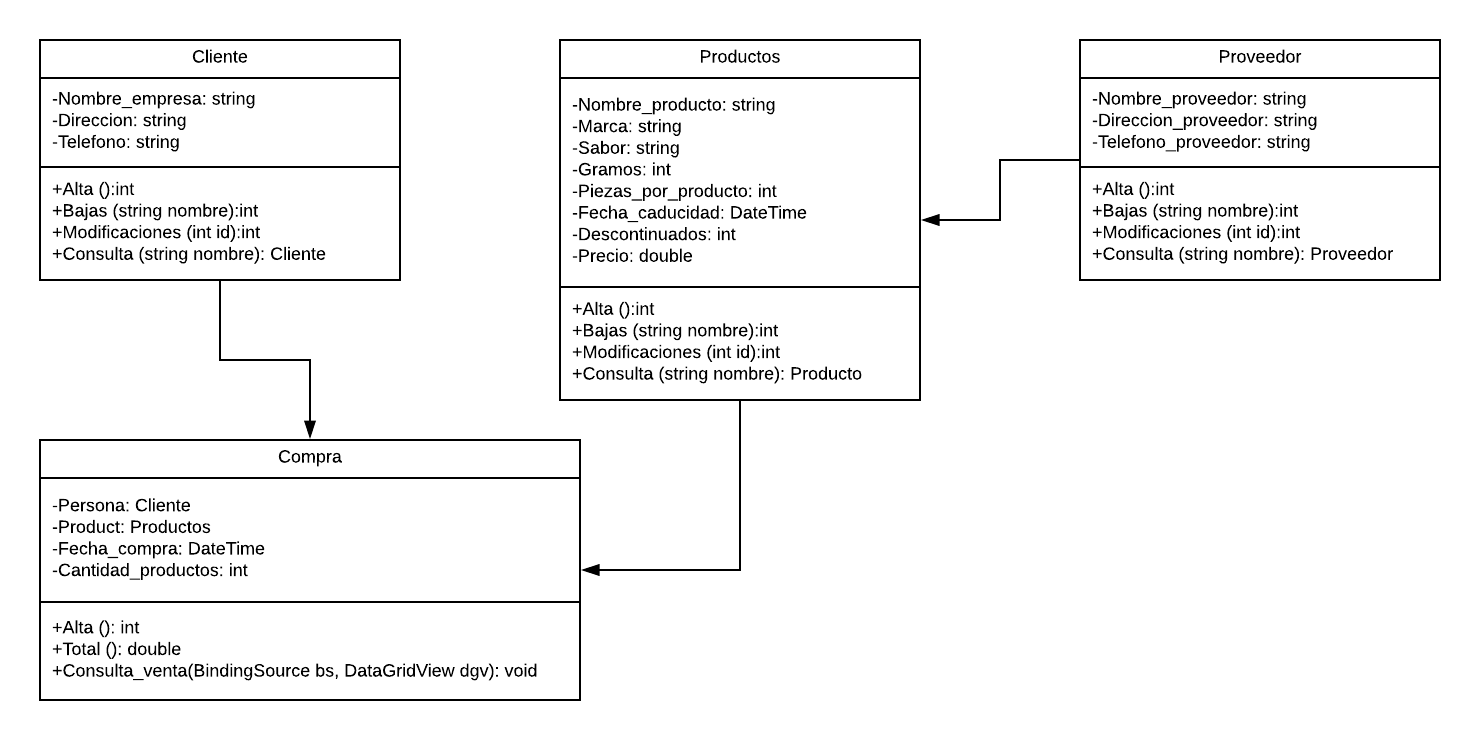
Para las ventas se requiere:

* Id factura
* Fecha de emisión
* Fecha de entrega
* Pecio de venta
* Cantidad del producto

Para envíos guardar:

* Id envió
* Placas de la camioneta repartidora
* Modelo de la camioneta
* Nombre del chofer

DIAGRAMA DE CLASES

****

CLASES

public class OperacionesSQL

{

private ConexionDB conexion = new ConexionDB();

SqlDataReader leer;

DataTable tabla = new DataTable();

SqlCommand comando = new SqlCommand();

SqlCommand comando2 = new SqlCommand();

//--------CLIENTE----------------------------------------------//

public DataTable Cliente\_Mostrar()

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "select \* from Cliente";

comando.CommandType = CommandType.Text;

leer = comando.ExecuteReader();

tabla.Load(leer);

conexion.cerrarConexion();

return tabla;

}

public void Insertar\_cliente(string nombre)

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "insert into Cliente values('" + nombre + "')";

comando.CommandType = CommandType.Text;

comando.ExecuteNonQuery();

}

public void Modificar\_cliente(string nombre,int id)

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "updateCliente2";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.AddWithValue("@nombre", nombre);

comando.Parameters.AddWithValue("id",id);

comando.ExecuteNonQuery();

comando.Parameters.Clear();

}

public void Baja\_cliente(int id)

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "BajaCliente";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.AddWithValue("@id", id);

comando.ExecuteNonQuery();

//comando.ExecuteNonQuery

comando.Parameters.Clear();

}

//--------PROVEEDOR----------------------------------------------//

public DataTable Proveedor\_Mostrar()

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "select \* from Proveedor";

comando.CommandType = CommandType.Text;

leer = comando.ExecuteReader();

tabla.Load(leer);

conexion.cerrarConexion();

return tabla;

}

public void Insertar\_proveedor(string nombre)

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "insert into Proveedor values('" + nombre + "')";

comando.CommandType = CommandType.Text;

comando.ExecuteNonQuery();

}

public void Modificar\_proveedor(string nombre, int id)

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "UpdateProveedor";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.AddWithValue("@nombre", nombre);

comando.Parameters.AddWithValue("@id", id);

comando.ExecuteNonQuery();

comando.Parameters.Clear();

}

public void Baja\_proveedor(int id)

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "BajaProveedor";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.AddWithValue("@id", id);

comando.ExecuteNonQuery();

comando.Parameters.Clear();

}

//--------PRODUCTO----------------------------------------------//

public DataTable Producto\_Mostrar()

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "select \* from Producto";

comando.CommandType = CommandType.Text;

leer = comando.ExecuteReader();

tabla.Load(leer);

conexion.cerrarConexion();

return tabla;

}

public void Insertar\_producto(int idproveedor,string nombre,string marca,string sabor,int gramos,int piezas,DateTime fecha\_Caducidad,int descontinuados,double precio)

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "insert into Producto values("+idproveedor+",'"+nombre+ "','" + marca + "','" + sabor + "'," + gramos + "," + piezas + ",'" + fecha\_Caducidad+ "'," + descontinuados+ "," + precio+")";

comando.CommandType = CommandType.Text;

comando.ExecuteNonQuery();

conexion.cerrarConexion();

}

public ComboBox cargaIDproveedor()

{

ComboBox comboid = new ComboBox();

comando.Connection = conexion.abrirConexion();

comando.CommandText = "select Proveedor.idProveedor from Proveedor";

comando.CommandType = CommandType.Text;

leer = comando.ExecuteReader();

comboid.Items.Add(leer);

conexion.cerrarConexion();

return comboid;

}

public void Modificar\_producto(int idproveedor, string nombre, string marca, string sabor, int gramos, int piezas, DateTime fecha\_Caducidad, float precio,int idproducto)

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "ModificacionesProducto";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.AddWithValue("@idprove", idproveedor);

comando.Parameters.AddWithValue("@nombre",nombre) ;

comando.Parameters.AddWithValue("@marca", marca);

comando.Parameters.AddWithValue("@sabor", sabor);

comando.Parameters.AddWithValue("@gramos", gramos);

comando.Parameters.AddWithValue("@piezas", piezas);

comando.Parameters.AddWithValue("@@fecha\_Caducidad", fecha\_Caducidad);

comando.Parameters.AddWithValue("@precio", precio);

comando.Parameters.AddWithValue("@idproducto", idproducto);

comando.ExecuteNonQuery();

comando.Parameters.Clear();

}

public void Baja\_producto(int id)

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "delete from Producto where idProducto="+id;

comando.CommandType = CommandType.Text;

comando.ExecuteNonQuery();

}

//--------COMPRA----------------------------------------------//

public DataTable Compra\_Mostrar()

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "select \* from Compra";

comando.CommandType = CommandType.Text;

leer = comando.ExecuteReader();

tabla.Load(leer);

conexion.cerrarConexion();

return tabla;

}

public void Modificar\_compra(int cantidad, int id)

{

comando.Connection = conexion.abrirConexion();

comando.CommandText = "UpdateProveedor";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.AddWithValue("@cantidad", cantidad);

comando.Parameters.AddWithValue("@id", id);

comando.ExecuteNonQuery();

comando.Parameters.Clear();

}

}

FORMULARIOS

public partial class Principal : Form

{

public Principal()

{

InitializeComponent();

}

private void timer1\_Tick(object sender, EventArgs e)

{

lblHora.Text = DateTime.Now.ToString("HH:mm: ss");

}

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

{

Form1 abrir = new Form1();

abrir.Show();

}

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

{

Proveedor abrir = new Proveedor();

abrir.Show();

}

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

{

Productos abrir = new Productos();

abrir. Show();

}

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

{

Compras abrir = new Compras();

abrir.Show();

}

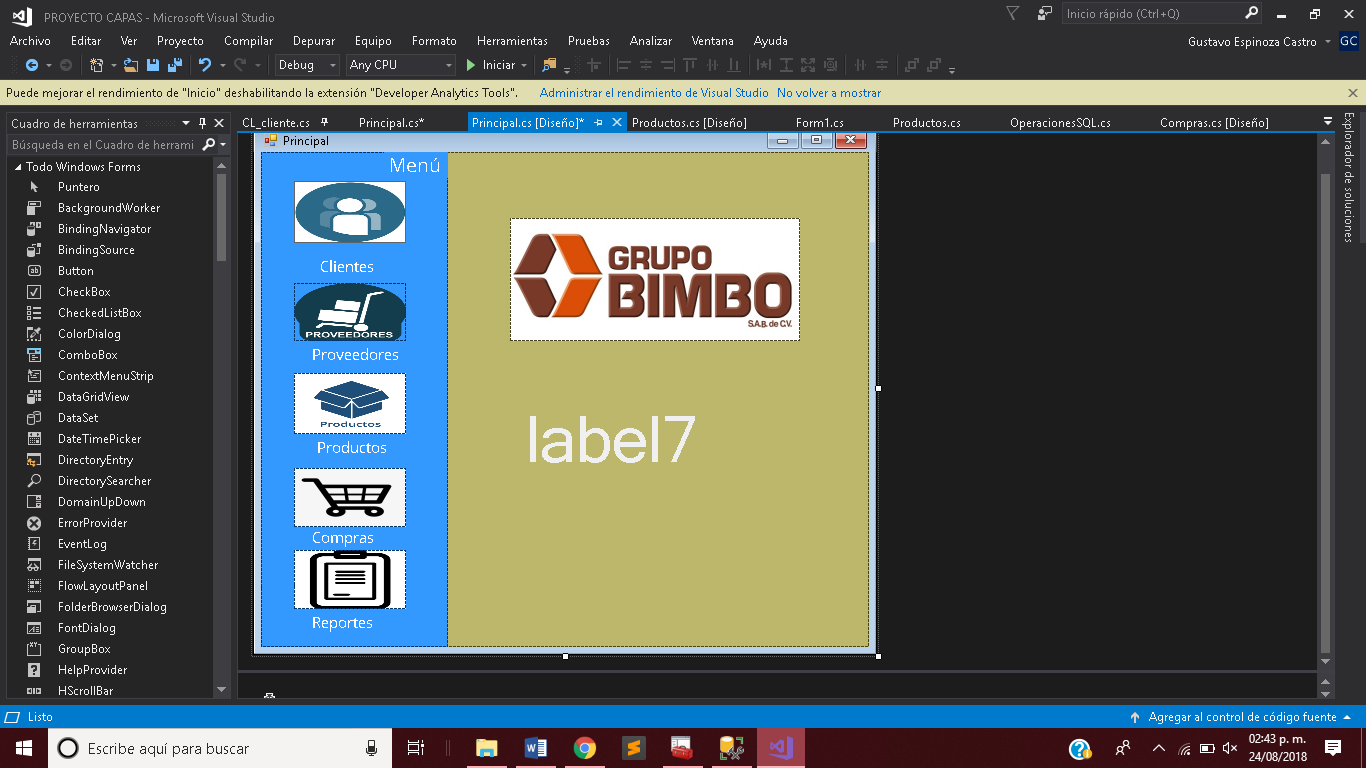
private void ptbReportes\_Click(object sender, EventArgs e)

{

Reportes abrir = new Reportes();

abrir.Show();

}



public partial class Cliente : Form

{

CL\_cliente objeto = new CL\_cliente();

public int idCliente = 0;

public bool Editar = false;

public Form1()

{

InitializeComponent();

}

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

{

MostarClientes();

}

private void MostarClientes()

{

dgvCliente.DataSource = objeto.mostrarCliente();

}

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

{

//INSERTAR//

if (Editar == false)

{

try

{

objeto.Insertarcliente(txtNombre.Text);

MessageBox.Show("Registro insertado correctamente");

Limpiar();

MostarClientes();

}

catch (Exception ex)

{

MessageBox.Show("Operacion fallida por: " + ex);

}

}

//MODIFICAR//

if (Editar == true)

{

try

{

objeto.Modificarcliente(txtNombre.Text, idCliente);

MessageBox.Show("Modificacion exitosa");

Editar = false;

Limpiar();

MostarClientes();

}

catch (Exception ex)

{

MessageBox.Show("Operacion fallida por: " + ex);

}

}

}

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

{

if (dgvCliente.SelectedRows.Count > 0)

{

Editar = true;

txtNombre.Text = dgvCliente.CurrentRow.Cells["nombre\_empresa"].Value.ToString();

idCliente = Convert.ToInt32(dgvCliente.CurrentRow.Cells["idCliente"].Value);

}

else

MessageBox.Show("Selecciona la fila a editar");

}

private void Limpiar()

{

txtNombre.Clear();

}

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

{

if (dgvCliente.SelectedRows.Count > 0)

{

idCliente = Convert.ToInt32(dgvCliente.CurrentRow.Cells["idCliente"].Value);

objeto.Bajacliente(idCliente);

MessageBox.Show("Operacion exitosa");

MostarClientes();

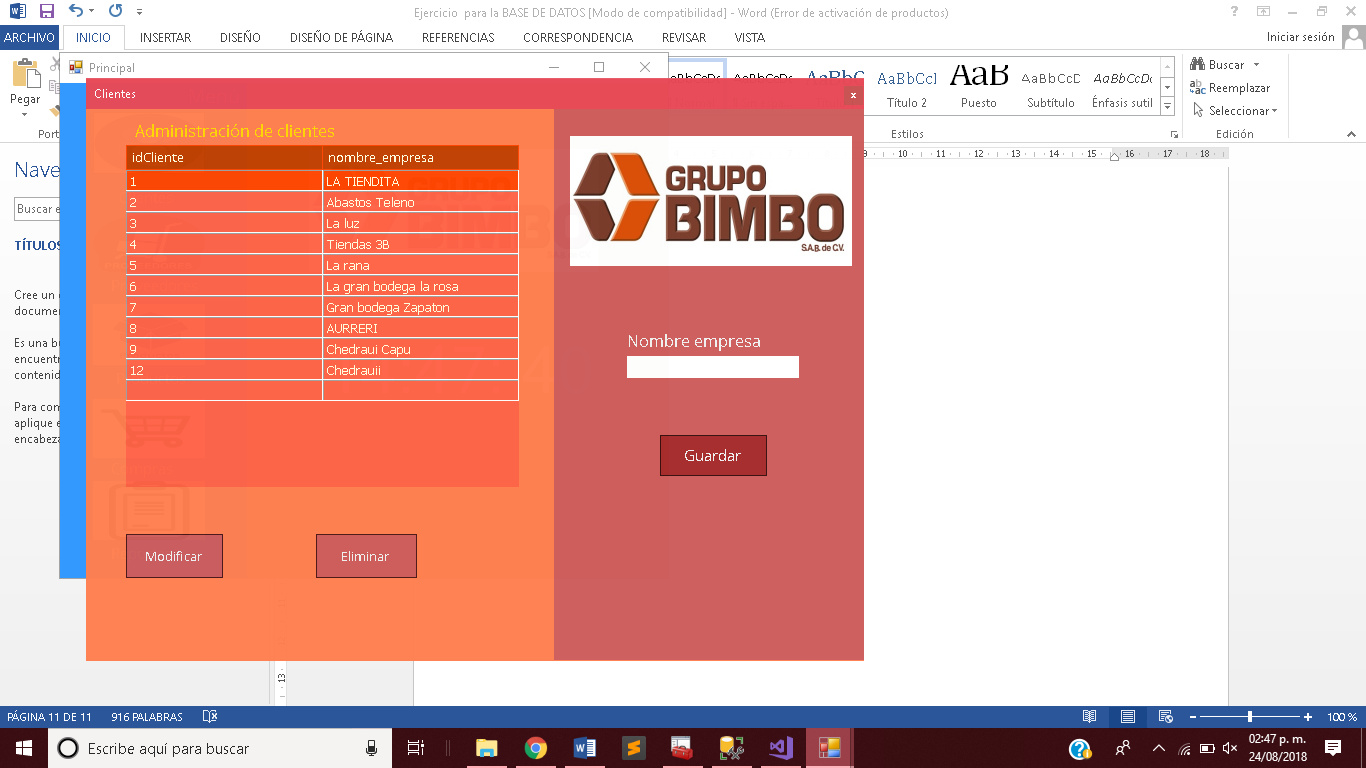
}

else

MessageBox.Show("Error en la operacion");

}

}



public partial class Proveedor : Form

{

CL\_cliente.CL\_proveedor objeto = new CL\_cliente.CL\_proveedor();

public int idProveedor = 0;

public bool Editar = false;

public Proveedor()

{

InitializeComponent();

}

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

{

MostarProveedor();

}

private void MostarProveedor()

{

dgvProveedor.DataSource = objeto.mostrarProveedor();

}

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

{

//INSERTAR//

if (Editar == false)

{

try

{

objeto.Insertarproveedor(txtNombre.Text,idProveedor);

MessageBox.Show("Registro insertado correctamente");

MostarProveedor();

}

catch (Exception ex)

{

MessageBox.Show("Operacion fallida por: " + ex);

}

}

//MODIFICAR//

if (Editar == true)

{

try

{

objeto.Modificar\_proveedor(txtNombre.Text,idProveedor);

MessageBox.Show("Modificacion exitosa");

Editar = false;

;

MostarProveedor();

}

catch (Exception ex)

{

MessageBox.Show("Operacion fallida por: " + ex);

}

}

}

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

{

if (dgvProveedor.SelectedRows.Count > 0)

{

idProveedor= Convert.ToInt32(dgvProveedor.CurrentRow.Cells["idProveedor"].Value);

objeto.Bajaproveedor(idProveedor);

MessageBox.Show("Operacion exitosa");

MostarProveedor();

}

else

MessageBox.Show("Error en la operacion");

}

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

{

if (dgvProveedor.SelectedRows.Count > 0)

{

Editar = true;

txtNombre.Text = dgvProveedor.CurrentRow.Cells["nombre\_proveedor"].Value.ToString();

idProveedor = Convert.ToInt32(dgvProveedor.CurrentRow.Cells["idProveedor"].Value);

}

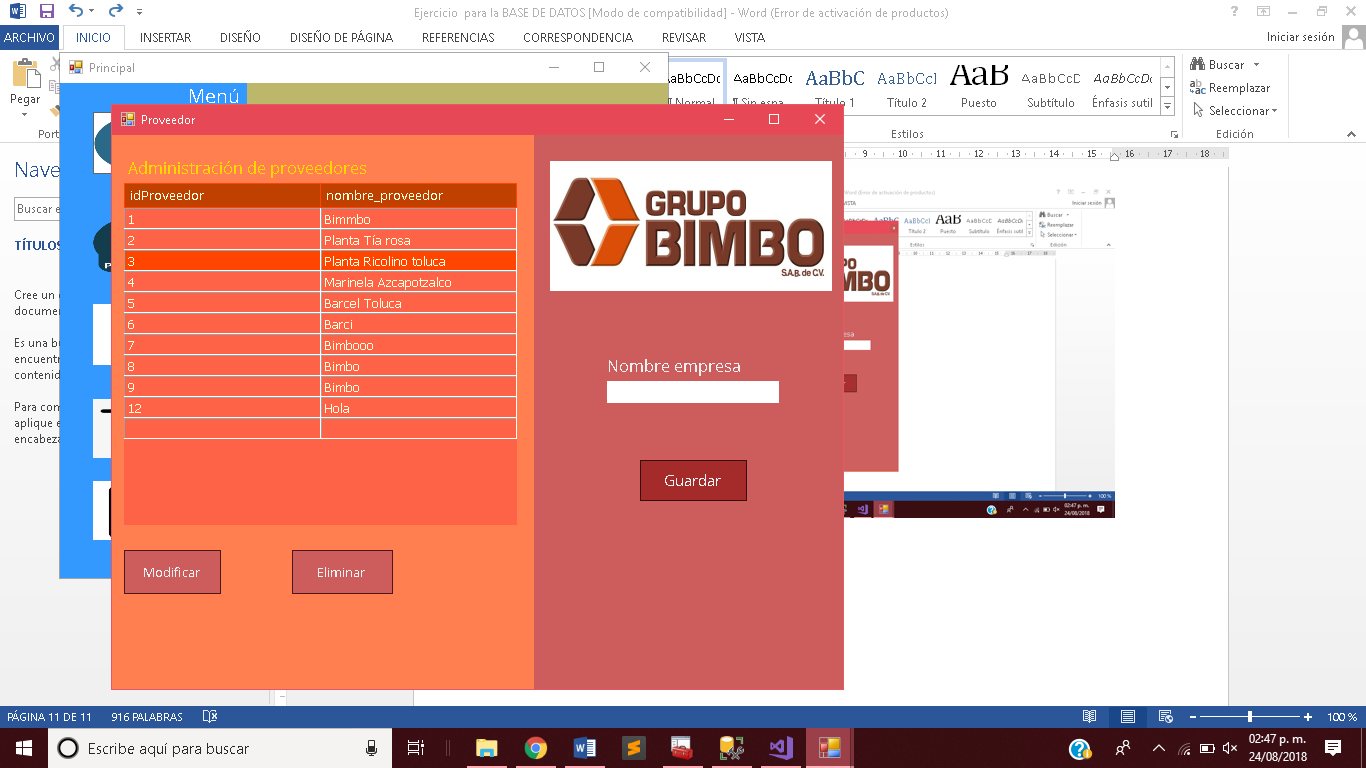
else

MessageBox.Show("Selecciona la fila a editar");

}

}

}



public partial class Productos : Form

{

CL\_cliente.CL\_producto objeto = new CL\_cliente.CL\_producto();

public int descontinuado = 0;

public bool Editar = false;

public int idproducto = 0;

public Productos()

{

InitializeComponent();

}

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

{

//INSERTAR//

//MODIFICAR//

if (Editar == true)

{

try

{

objeto.Modificar\_producto(Convert.ToInt32(cmbIdproveedor.Text), textNombreproducto.Text, cmbMarca.Text, cmbSabor.Text, Convert.ToInt32(cmbGramos.Text), Convert.ToInt32(cmbPiezas.Text), Convert.ToDateTime(dtpFecha.Value.ToShortDateString()), Convert.ToSingle(txtPrecio), Convert.ToInt32(txtIdproducto));

MessageBox.Show("Modificacion exitosa");

Editar = false;

MostarProductos();

}

catch (Exception ex)

{

MessageBox.Show("Operacion fallida por: " + ex);

}

}

}

private void MostarProductos()

{

dgvProducto.DataSource = objeto.Producto\_Mostrar();

}

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

{

for (int x = 1; x < 11; x++)

{

cmbPiezas.Items.Add(x);

}

for (int x = 10; x < 101; x++)

{

cmbGramos.Items.Add(x);

}

MostarProductos();

cmbIdproveedor.Items.Add(objeto.cargaIDproveedor());

}

private void cmbIdproveedor\_SelectedIndexChanged(object sender, EventArgs e)

{

}

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

{

if (dgvProducto.SelectedRows.Count > 0)

{

Editar = true;

cmbIdproveedor.Text =dgvProducto.CurrentRow.Cells["idProveedor"].Value.ToString();

textNombreproducto.Text = dgvProducto.CurrentRow.Cells["nombre"].Value.ToString();

cmbMarca.Text = dgvProducto.CurrentRow.Cells["marca"].Value.ToString();

cmbSabor.Text = dgvProducto.CurrentRow.Cells["sabor"].Value.ToString();

cmbGramos.Text = dgvProducto.CurrentRow.Cells["gramos"].Value.ToString();

cmbPiezas.Text = dgvProducto.CurrentRow.Cells["pieza\_x\_producto"].Value.ToString();

dtpFecha.Text = dgvProducto.CurrentRow.Cells["fecha\_caducidad"].Value.ToString();

txtPrecio.Text = dgvProducto.CurrentRow.Cells["precio"].Value.ToString();

idproducto = Convert.ToInt32(dgvProducto.CurrentRow.Cells["idProducto"].Value);

}

else

MessageBox.Show("Selecciona la fila a editar");

}

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

{

if (dgvProducto.SelectedRows.Count > 0)

{

idproducto = Convert.ToInt32(dgvProducto.CurrentRow.Cells["idProducto"].Value);

objeto.Baja\_producto(idproducto);

MessageBox.Show("Operacion exitosa");

MostarProductos();

}

else

MessageBox.Show("Error en la operacion");

}

}

}



public partial class Compras : Form

{

CL\_cliente.CL\_compra objeto = new CL\_cliente.CL\_compra();

public Compras()

{

InitializeComponent();

}

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

{

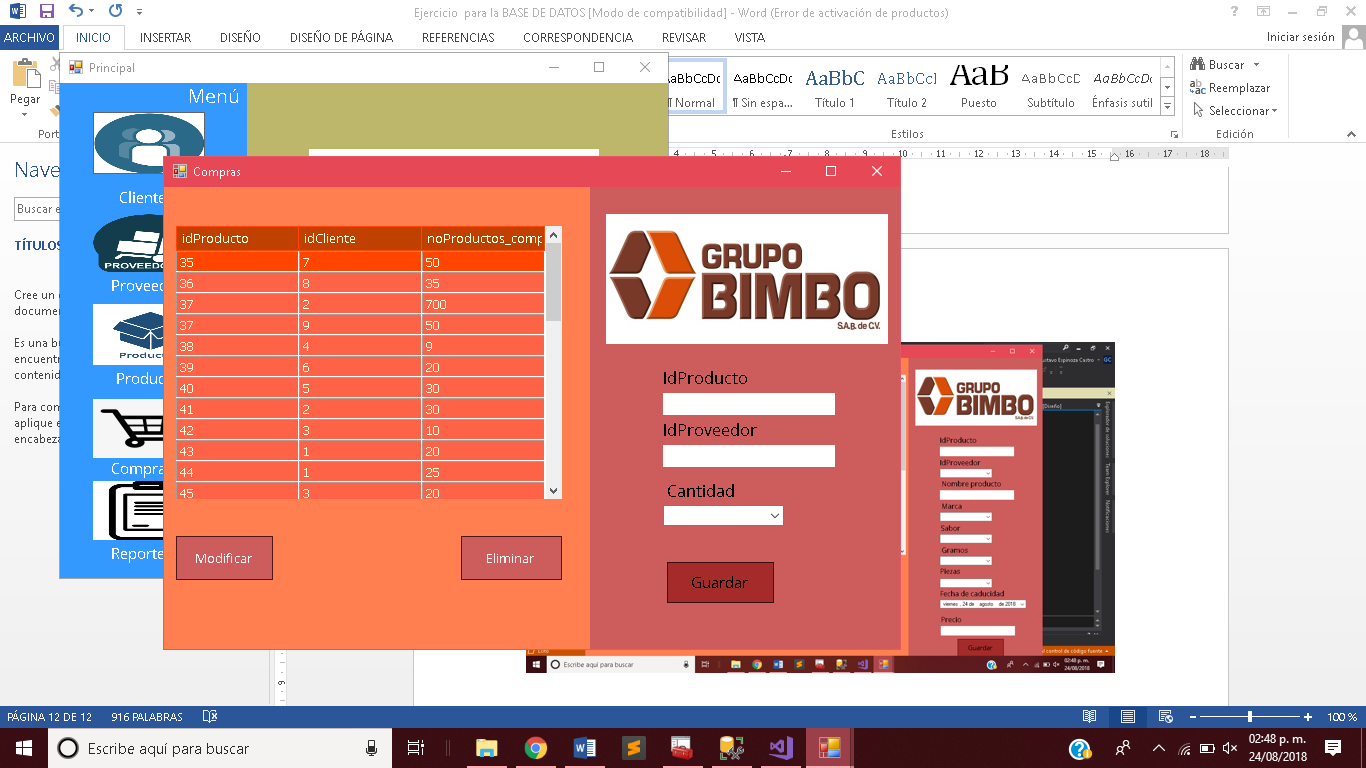
MostrarCompra();

}

private void MostrarCompra()

{

dgvCompra.DataSource = objeto.Compra\_Mostrar();

 }