29-4-2016

Proyecto Cristal

Profesor: Arturo Rojas López

Alumnos:

González Reyes Idania

Rojas Quintero Irving

4”A” SISTEMAS

# Redacción del problema

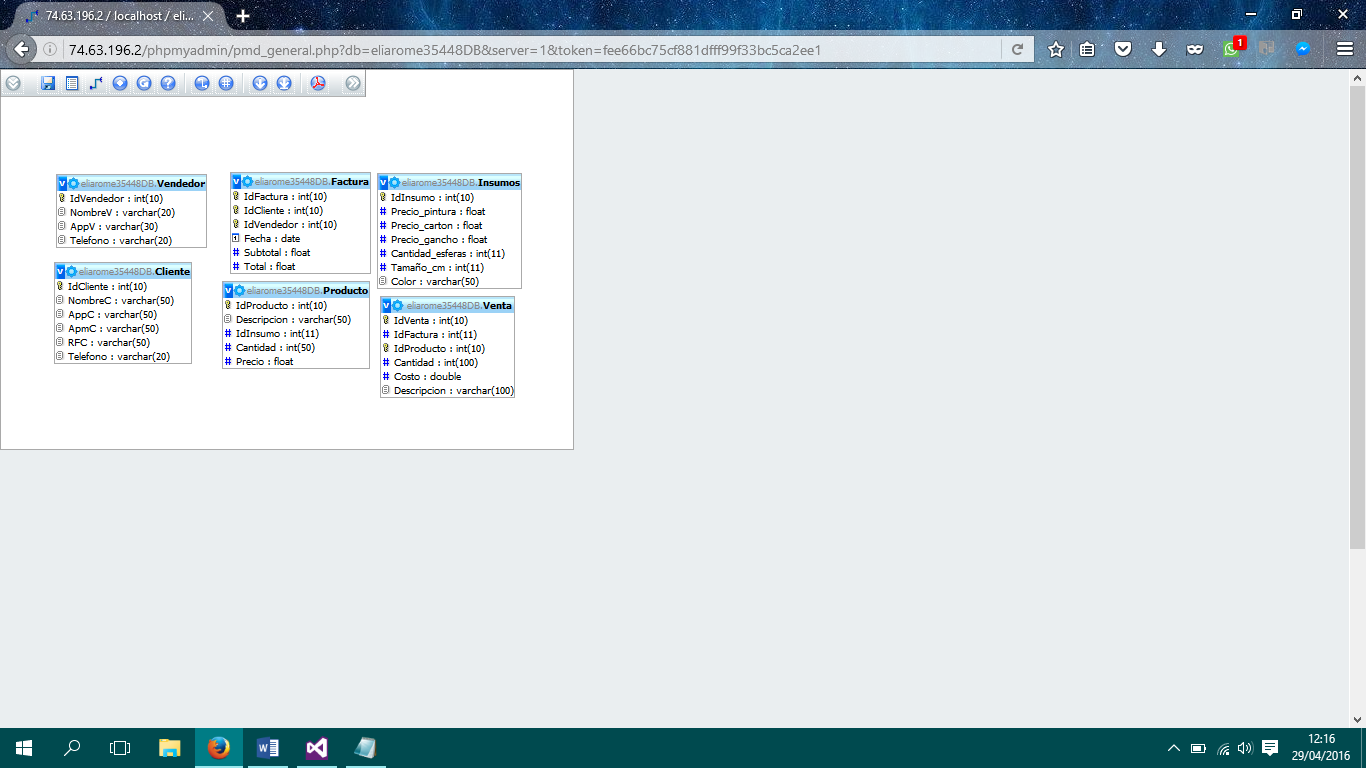
Cristal helado

La empresa Cristal helado ubicada en el municipio de Chignahuapan Puebla ha diseñado y fabricado esferas navideñas por más de 25 años, su dueño Rafael Noel Pérez Olmos inicio en la elaboración del producto junto con sus dos hijos (Pánfilo y Ximena) y su esposa Victoria, en la actualidad la empresa está constituida por 5 empleados coordinados por el Sr. Noel. El éxito comercial radica en la especialización del tipo de producto que elabora: esfera dorada, plateada y roja, en dos tamaños: 3 y 5 centímetros de diámetro. La venta la realiza por unidad o por caja, la cual puede ser de 6, 12 o 24 unidades. El costo de producción está determinado entre otras cosas por el precio de los insumos, tales como: el litro de pintura, el kilo de cartón para el embalaje según las unidades a empacar y el kilo del gancho usado para colgar la esfera que es el mismo para los dos tamaños. Un litro de pintura rinde para 300 unidades de 3 centímetros de diámetro, mientras para las de 5 centímetros rinde para 265 unidades. Un kilo de cartón para almacenar 6 unidades rinde para armar 30 cajas, para almacenar 12 unidades el kilo rinde para armar 20 cajas y finalmente para almacenar 24 unidades rinde para armar 15 cajas. Un kilo de ganchos alcanza para 700 esferas. Una vez que obtiene los precios de producción calcula el costo de fabricar una esfera y entonces para la venta aumenta un 30 % el costo de producción y obtiene el precio de esfera al público. Por ejemplo, si el litro de pintura cuesta $200, y el kilo de gancho cuesta $250, para producir una esfera dorada de 5 centímetros requiere de  $1.11 (0.75-pintura + 0.36-gancho), entonces al aumentarle $0.33, nos genera el precio de venta de $1.44. Un segundo ejemplo para obtener el precio de venta de una caja de esfera roja de 3 centímetros de diámetro de 12 unidades es el siguiente: si el litro de pintura cuesta $250, el kilo de cartón $80 y el kilo de gancho cuesta $240, para producir la caja requiere de $18.04 (9.96-pintura + 4-cartón + 4.08-gancho), entonces al aumentarle $5.41, nos genera el precio de venta $23.45.

Crear un sistema que permita al Sr. Noel realizar la venta de su producto al público de una forma rápida considerando la compra de los insumos que permiten calcular los precios de la esfera. El software elaborará una factura al cliente que lo requiera, agregando el iva respectivo al precio de venta o bien una nota que describa la compra realizada. La aplicación llevará un control de las ventas realizadas para poder realizar consultas de aquellas que solicitaron factura o nota y cuando se requiera ver el detalle de la venta respectiva, es decir, poder visualizar las ventas y cuando se seleccione una en particular poder ver el detalle de la misma (fecha, vendedor, producto(s), cantidades de la venta, etc.)

Utilizar una base de datos para gestionar la información de la empresa, conectar la aplicación a la base de datos y ofrecer las consultas de ventas indicadas además de la funcionalidad de la venta misma.

Clases



Clase datos

Para el acceso a la base de datos, creamos una clase llamada datos, que nos genera la conexión a la base de datos.

public class Datos

{

MySqlConnection carretera = new MySqlConnection();

public MySqlCommand cadena\_Acc;

public MySqlDataReader dr;

private MySqlDataAdapter adapt;

public bool AbrirConexion()

{

carretera.ConnectionString = "Server=74.63.196.2; Database=eliarome35448DB; uid=eliarome35448; pwd=4lnS6eodR3Ei;";

try

{

carretera.Open();

return true;

}

catch (Exception s)

{

MessageBox.Show("Error: " + s.Message);

carretera = null;

return false;

}

}

public void desconectar()

{

carretera.Close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*METODOS DE CONSTRUCCION DE COMANDOS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public MySqlCommand construye\_command(string cadena)

{

cadena\_Acc = new MySqlCommand(cadena, carretera);

return (cadena\_Acc);

}

public int ejecutanonquery()

{

int afectados;

try

{

afectados = cadena\_Acc.ExecuteNonQuery();

return (afectados);

}

catch (Exception oEx)

{

MessageBox.Show(oEx.Message);

return (0);

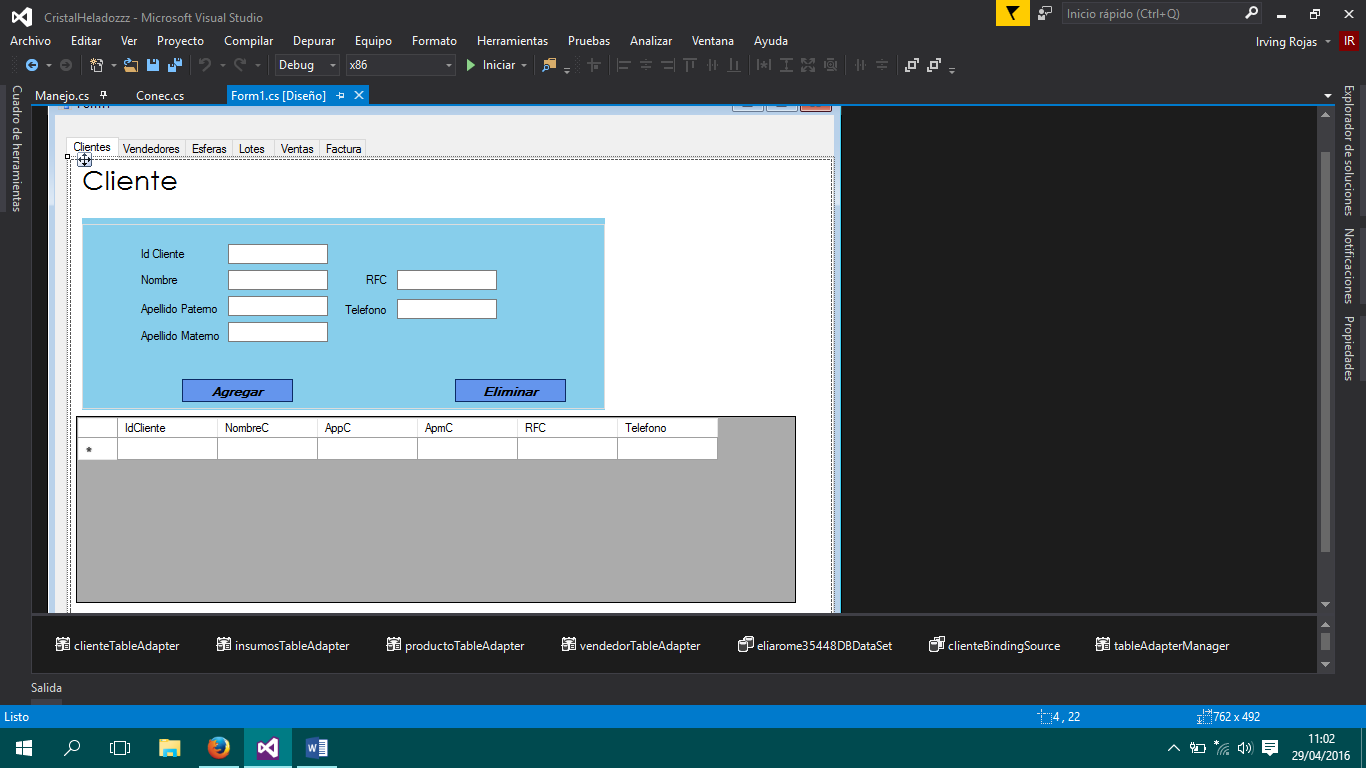
}

}

}

Cliente

Se presenta la página de cliente, la cual tiene como funcionalidad dar de alta un nuevo cliente



En la clase Cliente se encuentran el método de alta y eliminar

public class Cliente

{

public string NombreC { set; get; }

public string AppC { set; get; }

public string ApmC { set; get; }

public string RFC { set; get; }

public string Telefono { set; get; }

public int IdCliente { set; get; }

//CONSTRUCTOR~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\*\*\*\*\*\*\*\*\*\*\*\*\*\*+

public Cliente()

{

IdCliente = 0;

NombreC = "";

AppC = "";

ApmC = "";

RFC = "";

Telefono = "";

}

public Cliente(int ID, string nom, string ap, string am, string rf, string te)

{

IdCliente = ID;

NombreC = nom;

AppC = ap;

ApmC = am;

RFC = rf;

Telefono = te;

}

//\*\*\*\*\*\*metodo para manipular la base de datos ~~~~~~~~~~~~~~~~~~~\*\*\*\*\*\*\*\*\*\*

public int alta()

{

MySqlCommand Commando;

Datos Insertar = new Datos();

int regresa = 0;

if (Insertar.AbrirConexion())

{

string cad\_comando = "Insert into Cliente(IdCliente,NombreC,AppC,ApmC,RFC,Telefono)" + "values(" + IdCliente + ",'" + NombreC + "','" + AppC + "','" + ApmC + "','" + RFC + "','" + Telefono + "')";

Commando = Insertar.construye\_command(cad\_comando);

if (Insertar.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Commando.Connection.Close();

Insertar.desconectar();

}

else

regresa = -1;

return regresa;

}//fin alta

public int eliminar()

{

MySqlCommand Commando;

Datos Insertar = new Datos();

int regresa = 0;

if (Insertar.AbrirConexion())

{

string cad\_comando = "delete FROM `Cliente` WHERE `IdCliente` = " + IdCliente;

Commando = Insertar.construye\_command(cad\_comando);

if (Insertar.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Commando.Connection.Close();

Insertar.desconectar();

}

else

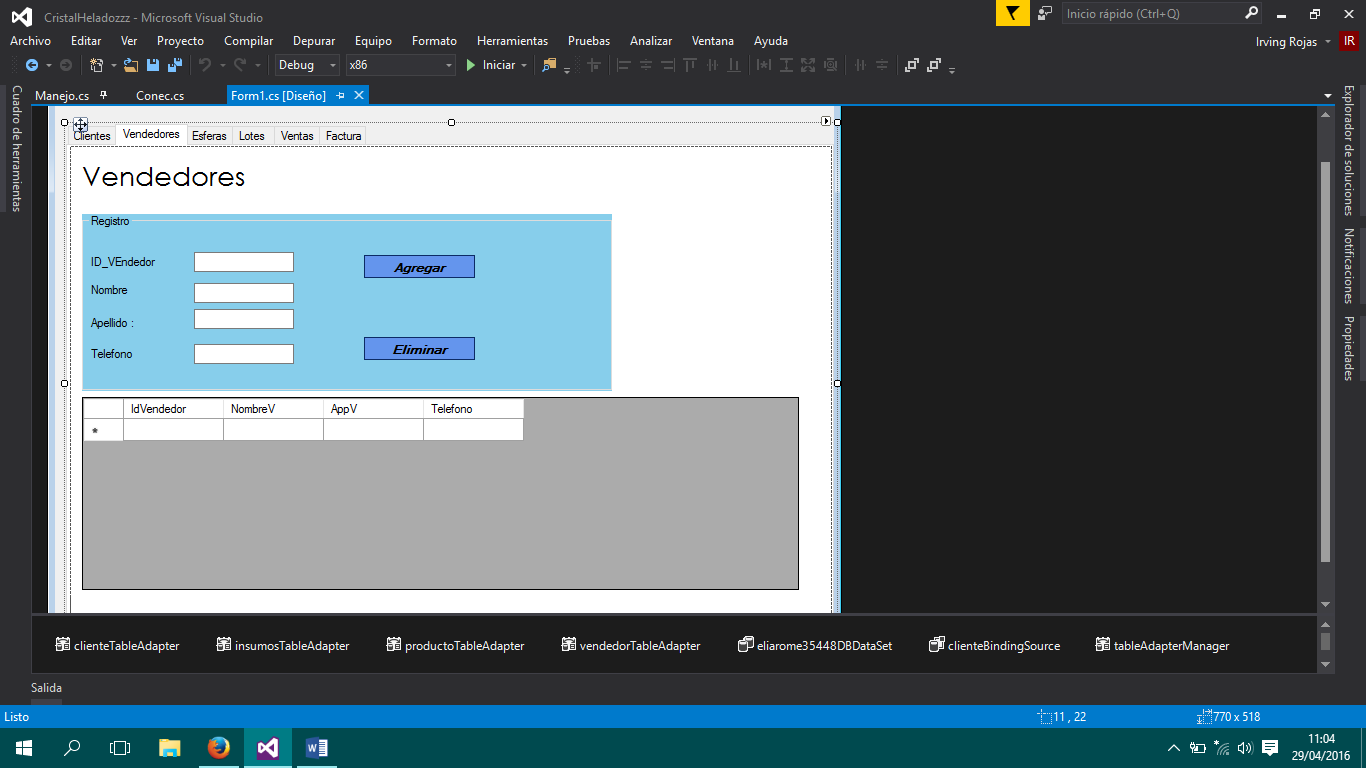
regresa = -1;

return regresa;

}

} // fin class Cliente

# Vendedores



public class Vendedor

{

public int IdVendedor { set; get; }

public string NombreV { set; get; }

public string AppV { set; get; }

public string Telefono { set; get; }

public int alta()

{

MySqlCommand Commando;

Datos inserta = new Datos();

int regresa = 0;

if (inserta.AbrirConexion())

{

string cad\_comando = "Insert into Vendedor(IdVendedor,NombreV,AppV,Telefono)" + "values(" + IdVendedor + ",'" + NombreV + "','" + AppV + "','" + Telefono + "')";

Commando = inserta.construye\_command(cad\_comando);

if (inserta.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Commando.Connection.Close();

inserta.desconectar();

}

else

regresa = -1;

return regresa;

}

public int Eliminar()

{

MySqlCommand Commando;

Datos inserta = new Datos();

int regresa = 0;

if (inserta.AbrirConexion())

{

string cad\_comando = "delete FROM `Vendedor` WHERE `IdVendedor` = " + IdVendedor;

Commando = inserta.construye\_command(cad\_comando);

if (inserta.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Commando.Connection.Close();

inserta.desconectar();

}

else

regresa = -1;

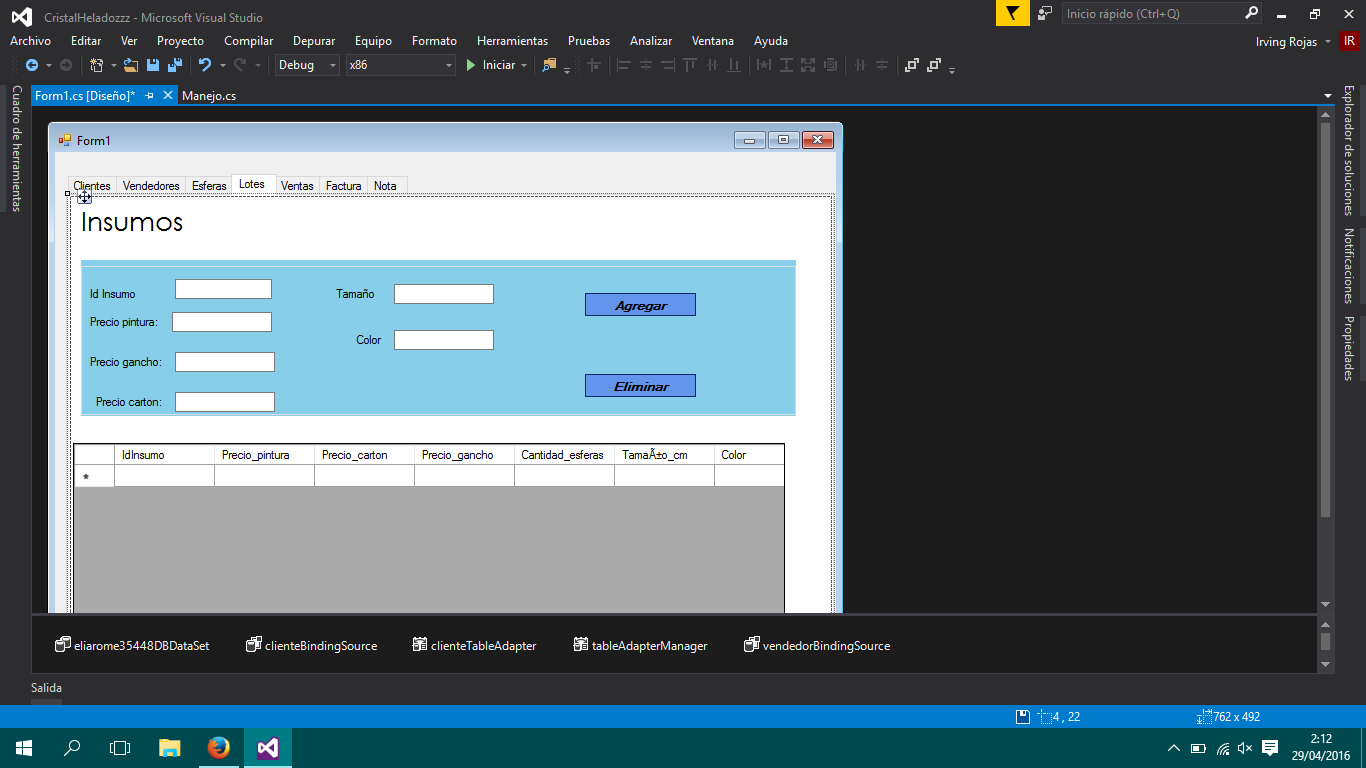
return regresa;

}

}//fin clas Vendedor

# Lotes

La parte de lotes representa también los insumos que se utilizan al elaborar un lote de esferas, se piden las especificaciones del nuevo lote de esferas, se agregan y se eliminan los lotes desde un método en la clase manejo.



public class Insumos

{

public int IdInsumo { set; get; }

public double PrePintura { get; set; }

public double PreGancho { get; set; }

public double PreCarton { get; set; }

public int tamaño { get; set; }

public String color { get; set; }

public int alta()

{

MySqlCommand Commando;

Datos inserta = new Datos();

int regresa = 0;

if (inserta.AbrirConexion())

{

string cad\_comando = "INSERT INTO `eliarome35448DB`.`Insumos` (`IdInsumo`, `Precio\_pintura`, `Precio\_carton`, `Precio\_gancho, `Tamaño\_cm`, `Color`) VALUES ('" + IdInsumo + "','" + PrePintura + "','" + PreCarton + "','" + PreGancho + "','" + tamaño + "','" + color + "')";

Commando = inserta.construye\_command(cad\_comando);

if (inserta.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Commando.Connection.Close();

inserta.desconectar();

}

else

regresa = -1;

return regresa;

}

public int eliminar()

{

MySqlCommand Commando;

Datos inserta = new Datos();

int regresa = 0;

if (inserta.AbrirConexion())

{

string cad\_comando = "delete FROM `Insumos` WHERE `IdInsumo` = " + IdInsumo;

Commando = inserta.construye\_command(cad\_comando);

if (inserta.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Commando.Connection.Close();

inserta.desconectar();

}

else

regresa = -1;

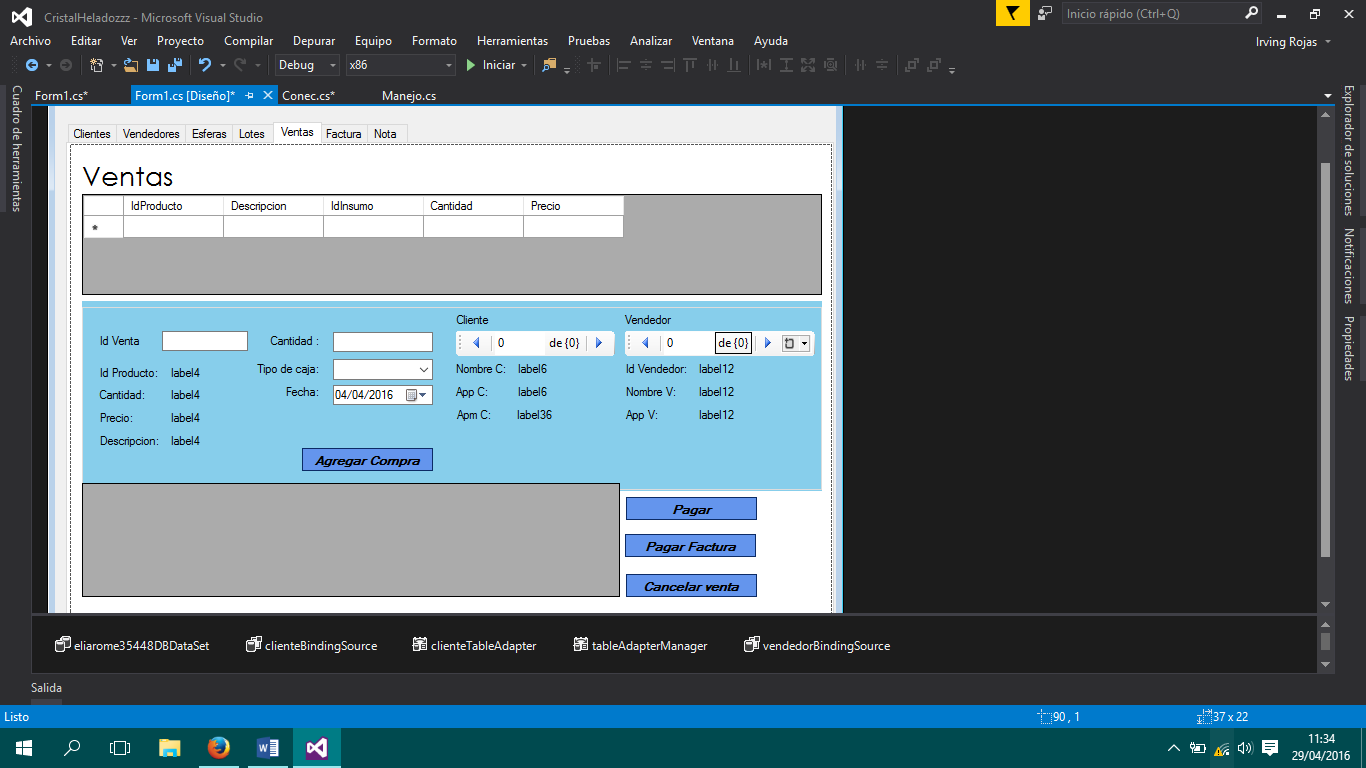
return regresa;

}

}//fin clas Insumos

# Ventas

En ventas ingresas el id de la venta, En el dataGridView se muestran los productos existentes, al seleccionar uno se muestra en los label4 Mostrando las especificaciones del producto, se ingresa la cantidad de productos a comprar, el tipo de caja especifica si es de 6, 12, 24 y unitaria, la fecha de la compra se secciona el id del cliente bindingNavigator y el Id\_Vendedor, en el botón agregar compra se ingresa al carrito de compras y se van almacenando hasta que el usuario presione el botón pagar que solo muestra la nota de pago, o en otro caso pagar factura que muestra la factura tomando datos desde ventas, y también se puede cancelar la venta.



public class Venta

{

public int IdVenta { get; set; }

public int IdEsfera { get; set; }

public int idFactura { get; set; }

public String Descripcion { get; set; }

public double Subtotal { get; set; }

public int Cantidad { get; set; }

public Venta()

{

}

public Venta(int idv, int ide, int idf, String des, double sub, int cant)

{

IdVenta = idv;

IdEsfera = ide;

idFactura = idf;

Descripcion = des;

Subtotal = sub;

Cantidad = cant;

}

public int alta()

{

MySqlCommand Commando;

Datos inserta = new Datos();

int regresa = 0;

if (inserta.AbrirConexion())

{

string cad\_comando = "INSERT INTO `eliarome35448DB`.`Venta` (`IdFactura`, `IdProducto`, `Cantidad`, `Costo`, `Descripcion`) VALUES ('" + idFactura + "','" + IdEsfera + "','" + Cantidad + "','" + Subtotal + "','" + Descripcion + "')";

Commando = inserta.construye\_command(cad\_comando);

if (inserta.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Commando.Connection.Close();

inserta.desconectar();

}

else

regresa = -1;

return regresa;

}

public int Eliminar()

{

MySqlCommand Commando;

Datos inserta = new Datos();

int regresa = 0;

if (inserta.AbrirConexion())

{

string cad\_comando = "delete FROM `Venta` WHERE `IdVenta` =" + IdVenta;

Commando = inserta.construye\_command(cad\_comando);

if (inserta.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Commando.Connection.Close();

inserta.desconectar();

}

else

regresa = -1;

return regresa;

}

public static MySqlConnection ObtenerConexion()

{

MySqlConnection conectar = new MySqlConnection("Server=74.63.196.2; Database=eliarome35448DB; uid=eliarome35448; pwd=4lnS6eodR3Ei;");

conectar.Open();

return conectar;

}

public void person(TextBox fa, Label l1)

{

string a = "", b = "", c = "";

MySqlCommand \_comando = new MySqlCommand(String.Format(

"SELECT `Name\_c`,`App\_c`,`Apm\_c` FROM `Cliente` WHERE `Id\_Cliente` = {0}", fa.Text), ObtenerConexion());

MySqlDataReader \_reader = \_comando.ExecuteReader();

while (\_reader.Read())

{

a = \_reader.GetString(0);

b = \_reader.GetString(1);

c = \_reader.GetString(2);

}

l1.Text = a + " " + b + " " + c;

}

public void suma(TextBox fa, Label l1, Label l2, Label l3)

{

double sum = 0;

MySqlCommand \_comando = new MySqlCommand(String.Format(

"SELECT sum(`Costo`) FROM `Venta` WHERE `IdFactura` = {0}", fa.Text), ObtenerConexion());

MySqlDataReader \_reader = \_comando.ExecuteReader();

while (\_reader.Read())

{

sum = \_reader.GetDouble(0);

}

l1.Text = Convert.ToString(sum);

double st = sum \* .16;

l2.Text = Convert.ToString(st);

l3.Text = Convert.ToString(sum + st);

}

public void Cancelar(TextBox fa)

{

MySqlCommand \_comando = new MySqlCommand(String.Format(

"delete FROM `Venta` WHERE `IdFactura` = " + fa.Text), ObtenerConexion());

MySqlDataReader \_reader = \_comando.ExecuteReader();

\_reader.Read();

}

public static List<Venta> carrito(TextBox id\_ven)

{

List<Venta> \_lista = new List<Venta>();

MySqlCommand \_comando = new MySqlCommand(String.Format("SELECT \*FROM `Venta` WHERE `IdFactura` = {0}", id\_ven.Text), ObtenerConexion());

MySqlDataReader \_reader = \_comando.ExecuteReader();

while (\_reader.Read())

{

Venta pin = new Venta();

pin.IdVenta = \_reader.GetInt32(0);

pin.idFactura = \_reader.GetInt32(1);

pin.IdEsfera = \_reader.GetInt32(2);

pin.Cantidad = \_reader.GetInt32(3);

pin.Subtotal = \_reader.GetDouble(4);

pin.Descripcion = \_reader.GetString(5);

\_lista.Add(pin);

}

return \_lista;

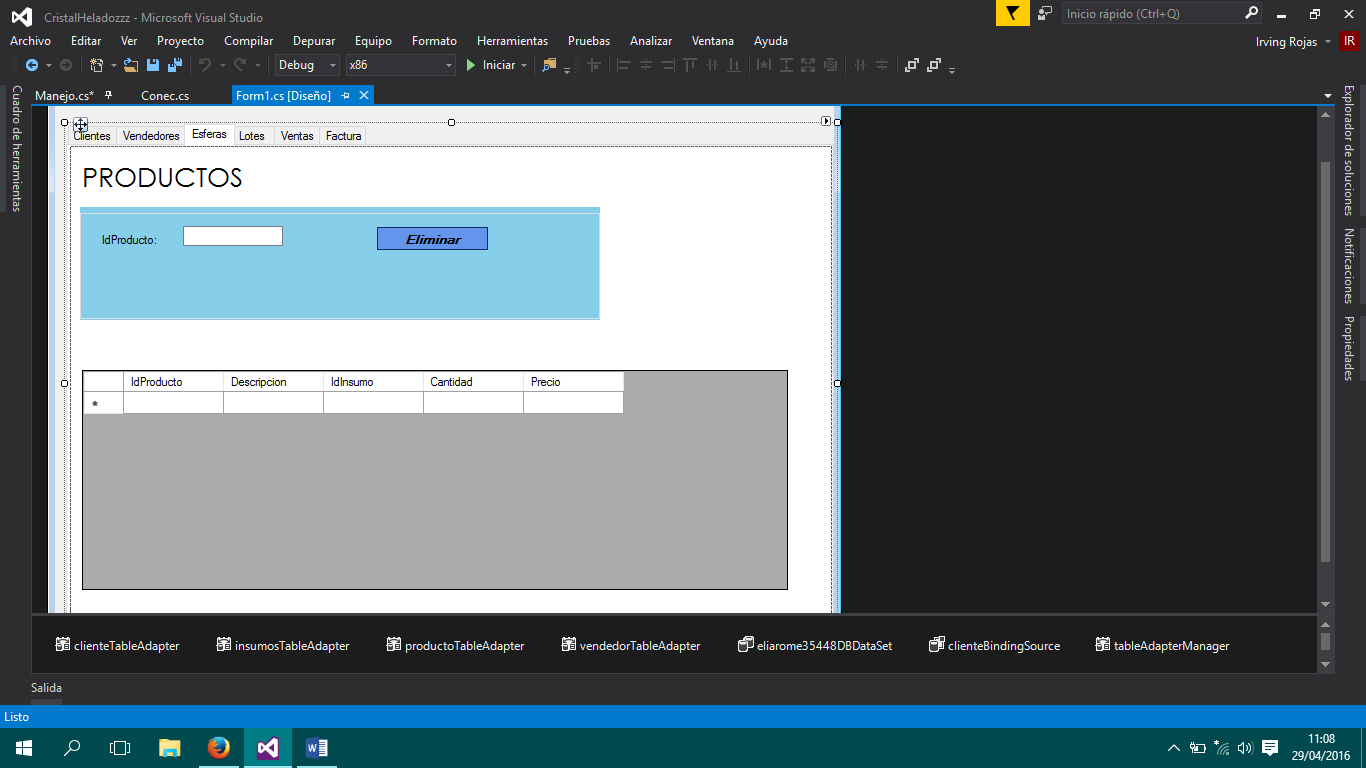
}

}

}

# Esferas

En este apartado solo se puede eliminar un producto(esferas), ya que al ingresar el lote se va llenando la tabla en la base de datos.



public class Producto

{

public int Idesfera { set; get; }

public int IdInsumo { set; get; }

public int cantidad { set; get; }

public double precio { set; get; }

public string descripcion { set; get; }

public int alta()

{

MySqlCommand Commando;

Datos inserta = new Datos();

int regresa = 0;

if (inserta.AbrirConexion())

{

string cad\_comando = "INSERT INTO `eliarome35448DB`.`Producto` (`Descripcion`, `IdInsumo`, `Cantidad`, `Precio`) VALUES ('" + descripcion + "','" + IdInsumo + "','" + cantidad + "','" + precio + "')";

Commando = inserta.construye\_command(cad\_comando);

if (inserta.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Commando.Connection.Close();

inserta.desconectar();

}

else

regresa = -1;

return regresa;

}

public int eliminar()

{

MySqlCommand Commando;

Datos inserta = new Datos();

int regresa = 0;

if (inserta.AbrirConexion())

{

string cad\_comando = "delete FROM `Producto` WHERE `IdProducto` =" + Idesfera;

Commando = inserta.construye\_command(cad\_comando);

if (inserta.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Commando.Connection.Close();

inserta.desconectar();

}

else

regresa = -1;

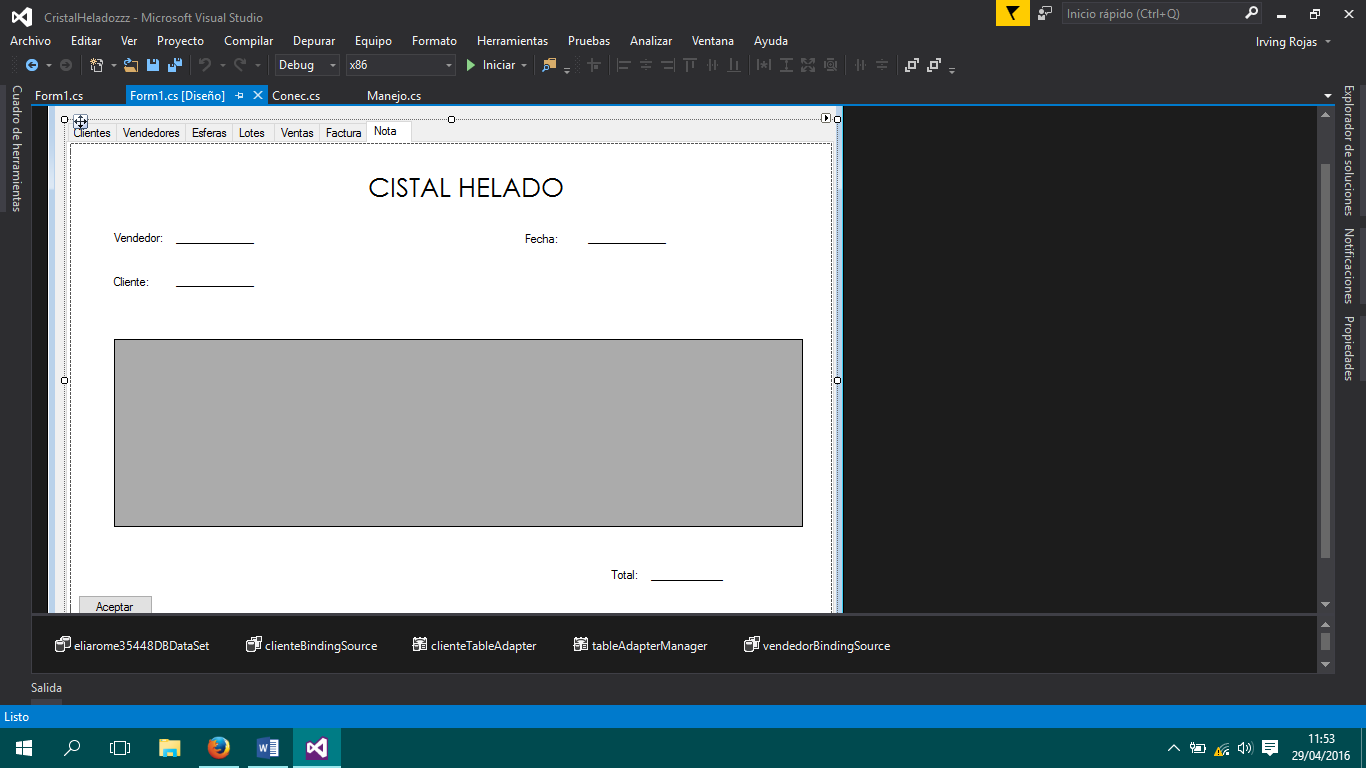
return regresa;

}

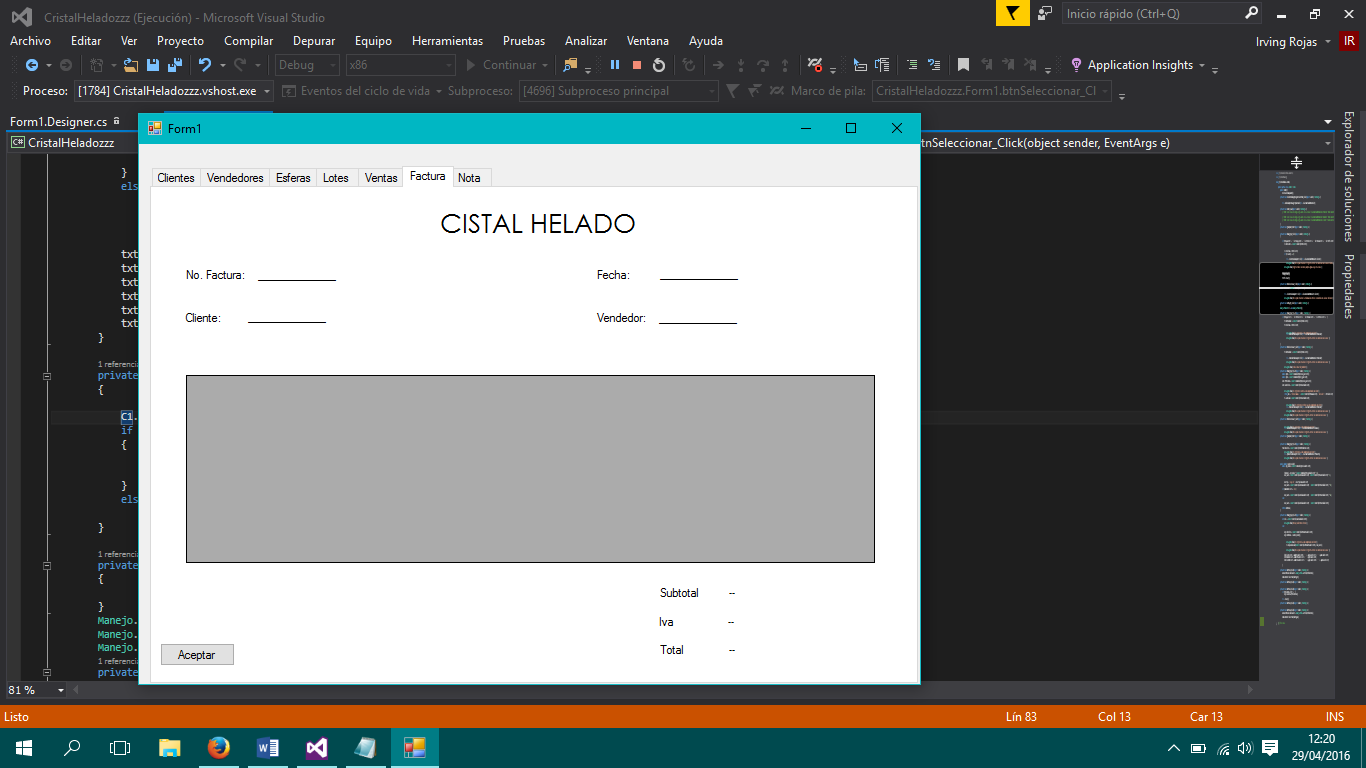
}//fin clas Producto

# Nota

Para generar la nota mostrara los datos según lo que ingreso en factura, llenando los campos para la nota, solo muestra lo siguiente y el dataGridView muestra los productos comprados con sus especifiaciones.



# Factura



# Clase Formulario

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

this.Validate();

this.clienteBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.eliarome35448DBDataSet);

}

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

{

// TODO: esta línea de código carga datos en la tabla 'eliarome35448DBDataSet.Producto' Puede moverla o quitarla según sea necesario.

this.productoTableAdapter.Fill(this.eliarome35448DBDataSet.Producto);

// TODO: esta línea de código carga datos en la tabla 'eliarome35448DBDataSet.Insumos' Puede moverla o quitarla según sea necesario.

this.insumosTableAdapter.Fill(this.eliarome35448DBDataSet.Insumos);

// TODO: esta línea de código carga datos en la tabla 'eliarome35448DBDataSet.Vendedor' Puede moverla o quitarla según sea necesario.

this.vendedorTableAdapter.Fill(this.eliarome35448DBDataSet.Vendedor);

// TODO: esta línea de código carga datos en la tabla 'eliarome35448DBDataSet.Cliente' Puede moverla o quitarla según sea necesario.

this.clienteTableAdapter.Fill(this.eliarome35448DBDataSet.Cliente);

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

Manejo.Cliente C1 = new Manejo.Cliente();

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

{

if (txtapm.Text != "" && txtapp.Text != "" && txtId.Text != "" && txtnom.Text != "" && txtrfc.Text != "" && txttel.Text != "")

{

C1.IdCliente = +Convert.ToInt32(txtId.Text);

C1.NombreC = txtnom.Text;

C1.AppC = txtapp.Text;

C1.ApmC = txtapm.Text;

C1.RFC = txtrfc.Text;

C1.Telefono = txttel.Text;

if (C1.alta() == 1)

{

MessageBox.Show("El registro ha sido completado con éxito");

this.clienteTableAdapter.Fill(this.eliarome35448DBDataSet.Cliente);

}

else

MessageBox.Show("No se pudo realizar el registro. Revise su conexión con la base de datos");

}

else

MessageBox.Show("Ingrese todos los datos porque alguna caja esta vacia");

txtapm.Clear();

txtapp.Clear();

txtId.Clear();

txtnom.Clear();

txtrfc.Clear();

txttel.Clear();

}

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

{

C1.IdCliente = +Convert.ToInt32(txtId.Text);

if (C1.eliminar() == 1)

{

MessageBox.Show("La Eliminacion ha sido completado con éxito");

this.clienteTableAdapter.Fill(this.eliarome35448DBDataSet.Cliente);

}

else

MessageBox.Show("No se pudo realizar la Eliminacion. Revise su conexión con la base de datos");

}

private void tabPage1\_Click(object sender, EventArgs e)

{

}

Manejo.Producto ES = new Manejo.Producto();

Manejo.Vendedor P1 = new Manejo.Vendedor();

Manejo.Insumos lote = new Manejo.Insumos();

private void btnAgregar5\_Click(object sender, EventArgs e)

{

if (txtapp2.Text != "" && txtId2.Text != "" && txtnom2.Text != "" && txttel2.Text != "")

{

P1.IdVendedor = +Convert.ToInt32(txtId2.Text);

P1.NombreV = txtnom2.Text;

P1.AppV = txtapp2.Text;

P1.Telefono = txttel2.Text;

if (P1.alta() == 1)

{

MessageBox.Show("El registro ha sido completado con éxito");

this.vendedorTableAdapter.Fill(this.eliarome35448DBDataSet.Vendedor);

}

else

MessageBox.Show("No se pudo realizar el registro. Revise su conexión con la base ");

}

else MessageBox.Show("Llene las casillas");

}

private void btnSeleccionar5\_Click(object sender, EventArgs e)

{

if (txtId2.Text != "")

{

P1.IdVendedor = +Convert.ToInt32(txtId2.Text);

if (P1.Eliminar() == 1)

{

MessageBox.Show("El registro ha sido completado con éxito");

this.vendedorTableAdapter.Fill(this.eliarome35448DBDataSet.Vendedor);

}

else

MessageBox.Show("No se pudo realizar el registro. Revise su conexión con la base ");

}

else

MessageBox.Show("Falta llenar id\_vendedor");

}

private void btnAgregar2\_Click(object sender, EventArgs e)

{

double pre1 = Convert.ToDouble(txtprecio\_pint.Text);

double pre2 = Convert.ToDouble(txtpre\_carto.Text);

double pre3 = Convert.ToDouble(txtpre\_ganc.Text);

lote.IdInsumo = Convert.ToInt32(textBox1.Text);

lote.tamaño = Convert.ToInt32(txttamaño.Text);

lote.PrePintura = Convert.ToDouble(txtprecio\_pint.Text);

lote.PreCarton = Convert.ToDouble(txtpre\_carto.Text);

lote.PreGancho = Convert.ToDouble(txtpre\_ganc.Text);

lote.CanEsferas = Convert.ToInt32(txtcantidad1.Text);

lote.color = txtcolor.Text;

if (lote.alta() == 1)

{

MessageBox.Show("El registro de Lote ha sido completado con éxito");

this.insumosTableAdapter.Fill(this.eliarome35448DBDataSet.Insumos);

string des = "Esfera Tamaño " + Convert.ToInt32(txttamaño.Text) + "cm Color " + txtcolor.Text;

double precio = (pre1 + pre2 + pre3) / Convert.ToDouble(txtcantidad1.Text);

ES.IdInsumo = Convert.ToInt32(textBox1.Text);

ES.cantidad = Convert.ToInt32(txtcantidad1.Text);

ES.descripcion = des;

ES.precio = precio;

if (ES.alta() == 1)

{

MessageBox.Show("El registro de Esfera ha sido completado con éxito");

this.vendedorTableAdapter.Fill(this.eliarome35448DBDataSet.Vendedor);

}

else

MessageBox.Show("No se pudo realizar el registro. Revise su conexión con la base ");

}

else

MessageBox.Show("No se pudo realizar el registro. Revise su conexión con la base ");

}

private void btnSeleccionar2\_Click(object sender, EventArgs e)

{

lote.IdInsumo = Convert.ToInt32(textBox1.Text);

if (lote.eliminar() == 1)

{

MessageBox.Show("El registro ha sido completado con éxito");

this.insumosTableAdapter.Fill(this.eliarome35448DBDataSet.Insumos);

}

else

MessageBox.Show("No se pudo realizar el registro. Revise su conexión con la base ");

}

private void groupBox3\_Enter(object sender, EventArgs e)

{

}

private void btnAgregar1\_Click(object sender, EventArgs e)

{

Manejo.Producto Prod = new Manejo.Producto();

Prod.Idesfera = Convert.ToInt32(txtIdProducto1.Text);

if (Prod.eliminar() == 1)

{

MessageBox.Show("El registro ha sido completado con éxito");

this.productoTableAdapter.Fill(this.eliarome35448DBDataSet.Producto);

}

else

MessageBox.Show("No se pudo realizar el registro. Revise su conexión con la base ");

}

string descrip = "";

int nue\_canti=0;

public double Cantid\_valid()

{

double pre\_unita = Convert.ToDouble(precioLabel1.Text);

double Subtotal = 0;

if (comboBox1.Text=="6")

{

descrip = "Caja 6 " + descripcionLabel1.Text;

Subtotal = pre\_unita \* (Convert.ToDouble(txtcantidad2.Text) \* 6);

nue\_canti = Convert.ToInt32(cantidadLabel1.Text)- Convert.ToInt32(txtcantidad2.Text) \* 6;

}

else

if (comboBox1.Text == "12")

{

descrip = "Caja 12 " + descripcionLabel1.Text;

Subtotal = pre\_unita \* (Convert.ToDouble(txtcantidad2.Text) \* 12);

nue\_canti = Convert.ToInt32(cantidadLabel1.Text) - Convert.ToInt32(txtcantidad2.Text) \* 12;

}

else

if (comboBox1.Text == "24")

{

descrip = "Caja 24 " + descripcionLabel1.Text;

Subtotal = pre\_unita \* (Convert.ToDouble(txtcantidad2.Text) \* 24);

nue\_canti = Convert.ToInt32(cantidadLabel1.Text) - Convert.ToInt32(txtcantidad2.Text) \* 24;

}

else

{

descrip = descripcionLabel1.Text;

Subtotal = pre\_unita \* (Convert.ToDouble(txtcantidad2.Text));

nue\_canti = Convert.ToInt32(cantidadLabel1.Text) - Convert.ToInt32(txtcantidad2.Text);

}

return Subtotal;

}

Manejo.Venta obj1 = new Manejo.Venta();

private void btnAgregar4\_Click(object sender, EventArgs e)

{

int cant = Convert.ToInt32(txtcantidad2.Text);

int exis = Convert.ToInt32(cantidadLabel1.Text);

if (cant > exis)

MessageBox.Show("No hay suficentes esferas");

else

{

obj1.idFactura = Convert.ToInt32(txtidventa.Text);

obj1.IdEsfera = Convert.ToInt32(idProductoLabel1.Text);

obj1.Cantidad = Convert.ToInt32(txtcantidad2.Text);

obj1.Subtotal = Cantid\_valid();

obj1.Descripcion = descrip;

if (obj1.alta() == 1)

{

MessageBox.Show("El registro ha sido completado con éxito");

dataGridView1.DataSource = Manejo.Venta.carrito(txtidventa);

ES.disponibilidad(Convert.ToInt32(idProductoLabel1.Text), nue\_canti);

this.productoTableAdapter.Fill(this.eliarome35448DBDataSet.Producto);

}

else

MessageBox.Show("No se pudo realizar el registro. Revise su conexión con la base ");

lblFactura.Text = txtidventa.Text;

lblCliente.Text = nombreCLabel1.Text + " " + appCLabel1.Text + " " + apmCLabel1.Text;

lblFecha.Text = dateTimePicker2.Value.ToShortDateString();

lblVendedor.Text = nombreVLabel1.Text + " " + appVLabel1.Text;

lblClienteN.Text = nombreCLabel1.Text + " " + appCLabel1.Text + " " + apmCLabel1.Text;

lblFechaN.Text = dateTimePicker2.Value.ToShortDateString();

lblVendedorN.Text = nombreVLabel1.Text + " " + appVLabel1.Text;

}

}

private void button3\_Click(object sender, EventArgs e)

{

dataGridView2.DataSource = Manejo.Venta.carrito(txtidventa);

obj1.suma(txtidventa,txtsubtotal,txtiva,txttotal);

tabControl1.SelectTab(tabPage7);

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

if (txtidventa.Text != "")

{

obj1.Cancelar(txtidventa);

}

this.Close();

}

private void button5\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button4\_Click(object sender, EventArgs e)

{

dataGridView3.DataSource = Manejo.Venta.carrito(txtidventa);

label32.Text = txtsubtotal.Text;

tabControl1.SelectTab(tabPage4);

}

}// fin class

# Corridas de escritorio

