



UNIVERSIDAD TECNOLÓGICA DE PUEBLA

Materia: Desarrollo de Aplicaciones 2

Maestra: Rojas López Arturo

Proyecto Desarrollo de Aplicaciones 2

Alumno:

- María Leticia Osorio Ventura
- Marcos Flores Osorio
- Juan Elías Romero Sánchez
- Azael León Castillo

Grupo/Grado: 4°A

Carrera: TICS

ÍNDICE

ÍNDICE.....	2
CÓDIGO BASE DE DATOS.....	3
CÓDIGO CLASE.....	14
CODIGO FORMULARIO	34
VENTANAS	49
REQUISITOS DE LA APLICACIÓN	58

CÓDIGO BASE DE DATOS

-- phpMyAdmin SQL Dump

-- version 4.0.5deb1

-- http://www.phpmyadmin.net

--

-- Servidor: localhost

-- Tiempo de generación: 18-04-2016 a las 12:14:19

-- Versión del servidor: 5.5.31-1

-- Versión de PHP: 5.5.1-2

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

SET time_zone = "+00:00";

/*!40101 SET

@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET

@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET

@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

```

/*!40101 SET NAMES utf8 */;

--

-- Base de datos: `eliarome33807DB`

--

-----

--

-- Estructura de tabla para la tabla `Carton`

--

CREATE TABLE IF NOT EXISTS `Carton` (
  `id_carton` int(11) NOT NULL AUTO_INCREMENT,
  `Precio` float DEFAULT NULL,
  `Kilos` float DEFAULT NULL,
  `Tamaño` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_carton`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

--

-- Volcado de datos para la tabla `Carton`

--

INSERT INTO `Carton` (`id_carton`, `Precio`, `Kilos`, `Tamaño`) VALUES

```

```
(1, 120, 2, 12),  
(2, 15.5, 20, 6),  
(3, 14, 20, 12);
```

```
-- -----
```

```
--
```

```
-- Estructura de tabla para la tabla `cliente`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `cliente` (  
  `Id_cliente` int(11) NOT NULL AUTO_INCREMENT,  
  `Nombre` varchar(15) DEFAULT NULL,  
  `Apellidos_P` varchar(25) DEFAULT NULL,  
  `Apellidos_M` varchar(25) DEFAULT NULL,  
  `Telefono` int(11) DEFAULT NULL,  
  PRIMARY KEY (`Id_cliente`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
```

```
--
```

```
-- Volcado de datos para la tabla `cliente`
```

```
--
```

```
INSERT INTO `cliente` (`Id_cliente`, `Nombre`, `Apellidos_P`, `Apellidos_M`,  
`Telefono`) VALUES
```

```
(2, 'Marcos', 'Osorio', 'Flores', 2759186),  
(3, 'Juan Elias', 'Sanchez', 'Romero', 1110372),  
(4, 'Azael', 'Leon', 'Castillo', 2369512),  
(5, 'Rafael', 'Moreno', 'Valle', 2459683),  
(6, 'Ingrid', 'Sanchez', 'Lopez', 3459621);
```

```
-- -----
```

```
--
```

```
-- Estructura de tabla para la tabla `Empleado`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `Empleado` (  
  `Id_Empleado` int(11) NOT NULL AUTO_INCREMENT,  
  `Nombre` varchar(15) DEFAULT NULL,  
  `Apellidos_P` varchar(25) DEFAULT NULL,  
  `Apellidos_M` varchar(25) DEFAULT NULL,  
  `Telefono` int(11) DEFAULT NULL,  
  PRIMARY KEY (`Id_Empleado`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

```
--
```

```
-- Volcado de datos para la tabla `Empleado`
```

```
--
```

```
INSERT INTO `Empleado` (`Id_Empleado`, `Nombre`, `Apellidos_P`,  
`Apellidos_M`, `Telefono`) VALUES
```

```
(1, 'Leticia', 'Osorio', 'Ventura', 3451296),  
(2, 'Alejandro', 'Osorio', 'Lopez', 1234578),  
(3, 'Jose Maria', 'Atenco', 'Jimenez', 3546258),  
(4, 'Abraham', 'Loranca', 'Rosas', 2459789);
```

```
-- -----
```

```
--
```

```
-- Estructura de tabla para la tabla `Esferas`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `Esferas` (  
  `Id_esferas` int(11) NOT NULL AUTO_INCREMENT,  
  `Existencia` int(11) DEFAULT NULL,  
  `Id_lote` int(11) DEFAULT NULL,  
  `precio_unitario` float DEFAULT NULL,  
  `caja` float NOT NULL,  
  PRIMARY KEY (`Id_esferas`),  
  KEY `Id_lote` (`Id_lote`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

```
--
```

```
-- Volcado de datos para la tabla `Esferas`
```

--

```
INSERT INTO `Esferas` (`Id_esferas`, `Existencia`, `Id_lote`, `precio_unitario`,  
`caja`) VALUES
```

```
(1, 300, 100, 4, 0);
```

-- -----

--

-- Estructura de tabla para la tabla `Factura`

--

```
CREATE TABLE IF NOT EXISTS `Factura` (
```

```
  `Id_Fatura` int(11) NOT NULL,
```

```
  `Id_Empleado` int(11) DEFAULT NULL,
```

```
  `Id_cliente` int(11) DEFAULT NULL,
```

```
  `Id_esferas` int(11) DEFAULT NULL,
```

```
  `Fecha` date DEFAULT NULL,
```

```
  `Cantidad` int(11) DEFAULT NULL,
```

```
  `Subtotal` float DEFAULT NULL,
```

```
  `Total` float DEFAULT NULL,
```

```
  PRIMARY KEY (`Id_Fatura`),
```

```
  KEY `Id_cliente` (`Id_cliente`),
```

```
  KEY `Id_esferas` (`Id_esferas`),
```

```
  KEY `Id_Empleado` (`Id_Empleado`)
```



```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Volcado de datos para la tabla `Factura`
```

```
--
```

```
INSERT INTO `Factura` (`Id_Fatura`, `Id_Empleado`, `Id_cliente`, `Id_esferas`,  
`Fecha`, `Cantidad`, `Subtotal`, `Total`) VALUES
```

```
(0, 1, 4, 1, '2016-04-13', 50, 2000, 2150),
```

```
(1, 1, 4, 1, '2016-04-13', 50, 2000, 2150);
```

```
-- -----
```

```
--
```

```
-- Estructura de tabla para la tabla `Gancho`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `Gancho` (
```

```
  `Id_gancho` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `Precio_unidad` float DEFAULT NULL,
```

```
  `Existencia` int(11) DEFAULT NULL,
```

```
  PRIMARY KEY (`Id_gancho`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

```
--
```

-- Volcado de datos para la tabla `Gancho`

--

```
INSERT INTO `Gancho` (`Id_gancho`, `Precio_unidad`, `Existencia`) VALUES
```

```
(1, 0.33, 700),
```

```
(2, 3, 50),
```

```
(3, 250, 100);
```

--

-- Estructura de tabla para la tabla `Lote`

--

```
CREATE TABLE IF NOT EXISTS `Lote` (
```

```
  `Id_lote` int(11) NOT NULL DEFAULT '0',
```

```
  `Id_pintura` int(11) DEFAULT NULL,
```

```
  `id_carton` int(11) DEFAULT NULL,
```

```
  `Id_gancho` int(11) DEFAULT NULL,
```

```
  `Cantidad_esferas` int(11) DEFAULT NULL,
```

```
  `Tamaño_cm` int(3) DEFAULT NULL,
```

```
  `Invercion` float DEFAULT NULL,
```

```
  PRIMARY KEY (`Id_lote`),
```

```
  KEY `id_carton` (`id_carton`),
```

```
  KEY `Id_pintura` (`Id_pintura`),
```

```

KEY `Id_gancho` (`Id_gancho`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Volcado de datos para la tabla `Lote`

--

INSERT INTO `Lote` (`Id_lote`, `Id_pintura`, `id_carton`, `Id_gancho`,
`Cantidad_esferas`, `Tamaño_cm`, `Invercion`) VALUES
(100, 1, 1, 1, 300, 6, 1200);

-----

--

-- Estructura de tabla para la tabla `Pintura`

--

CREATE TABLE IF NOT EXISTS `Pintura` (
  `Id_pintura` int(11) NOT NULL AUTO_INCREMENT,
  `Color` varchar(10) DEFAULT NULL,
  `Precio` float DEFAULT NULL,
  `litros` float DEFAULT NULL,
  PRIMARY KEY (`Id_pintura`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;

```

--

-- Volcado de datos para la tabla `Pintura`

--

```
INSERT INTO `Pintura` (`Id_pintura`, `Color`, `Precio`, `litros`) VALUES
```

```
(1, 'rojo', 100, 1),
```

```
(2, 'verde', 100, 1),
```

```
(3, 'Rojo', 100, 2),
```

```
(4, 'Amarillo', 100, 3),
```

```
(5, 'Amarillo', 15, 4),
```

```
(6, 'Amarillo', 200, 2),
```

```
(7, 'Azul', 250, 4);
```

--

-- Restricciones para tablas volcadas

--

--

-- Filtros para la tabla `Esferas`

--

```
ALTER TABLE `Esferas`
```

```
ADD CONSTRAINT `Esferas_ibfk_1` FOREIGN KEY (`Id_lote`) REFERENCES  
`Lote` (`Id_lote`);
```

--

-- Filtros para la tabla `Factura`

--

ALTER TABLE `Factura`

ADD CONSTRAINT `Factura_ibfk_1` FOREIGN KEY (`Id_cliente`) REFERENCES `cliente` (`Id_cliente`),

ADD CONSTRAINT `Factura_ibfk_2` FOREIGN KEY (`Id_esferas`) REFERENCES `Esferas` (`Id_esferas`),

ADD CONSTRAINT `Factura_ibfk_3` FOREIGN KEY (`Id_Empleado`) REFERENCES `Empleado` (`Id_Empleado`);

--

-- Filtros para la tabla `Lote`

--

ALTER TABLE `Lote`

ADD CONSTRAINT `Lote_ibfk_1` FOREIGN KEY (`id_carton`) REFERENCES `Carton` (`id_carton`),

ADD CONSTRAINT `Lote_ibfk_2` FOREIGN KEY (`Id_pintura`) REFERENCES `Pintura` (`Id_pintura`),

ADD CONSTRAINT `Lote_ibfk_3` FOREIGN KEY (`Id_gancho`) REFERENCES `Gancho` (`Id_gancho`);

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

CÓDIGO CLASE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MySql.Data.MySqlClient;
using System.Windows.Forms;

namespace class_esferas
{
    public class esferas
    {
        public int Id_esferas{ get; set; }
        public int Existencia { get; set; }
        public int id_lote { get; set; }
        public double Precio_unitario { get; set; }
    }
    public class lote
    {
        public int id_lote { get; set; }
        public int id_pint { get; set; }
        public int id_carton { get; set; }
        public int id_gancho { get; set; }
        public int cantidad { get; set; }
        public int tamaño { get; set; }
        public double inversion { get; set; }
    }
    public class cliente
    {
        public int id { get; set; }
        public string Nombre { get; set; }
        public string Apellido_P { get; set; }
        public string Apellido_M { get; set; }
        public int Telefono { get; set; }
    }
    public class pint
    {
        public int id {get;set;}
        public string Color {get;set;}
        public float precio {get;set;}
        public float litro {get;set;}
    }
}
```

```

    }
    public class gan
    {
        public int id { get; set; }
        public float precio_unit { get; set; }
        public int Existencia { get; set; }
    }
    public class cart
    {
        public int id { get; set; }
        public float prec { get; set; }
        public float kilos { get; set; }
        public int tam { get; set; }
    }
    public class conec
    {
        public static MySqlConnection ObtenerConexion()
        {
            MySqlConnection conectar = new MySqlConnection("server=69.162.95.98;
            database=eliarome33807DB; Uid=eliarome33807; pwd=x6aCKX20nTDb;");

            conectar.Open();
            return conectar;
        }
        public static List<pin> conPintura()
        {
            List<pin> _lista = new List<pin>();
            MySqlCommand _comando = new MySqlCommand(String.Format(
            "SELECT `Id_pintura`, `Color`, `Precio`, `litros` FROM `Pintura`"),
            ObtenerConexion());
            MySqlDataReader _reader = _comando.ExecuteReader();
            while (_reader.Read())
            {
                pin pin = new pin();

                pin.id = _reader.GetInt32(0);
                pin.Color = _reader.GetString(1);
                pin.precio = _reader.GetFloat(2);
                pin.litro = _reader.GetFloat(3);

                _lista.Add(pin);
            }
            return _lista;
        }

        public static List<cart> Carton()
        {
            List<cart> _lista = new List<cart>();
            MySqlCommand _comando = new MySqlCommand(String.Format(
            "SELECT `id_carton`, `Precio`, `Kilos`, `Tamaño` FROM `Carton`"),
            ObtenerConexion());
            MySqlDataReader _reader = _comando.ExecuteReader();
            while (_reader.Read())
            {
                cart pin = new cart();

```

```

        pin.id = _reader.GetInt32(0);
        pin.prec = _reader.GetFloat(1);
        pin.kilos = _reader.GetFloat(2);
        pin.tam = _reader.GetInt32(3);

        _lista.Add(pin);
    }
    return _lista;
}

public static List<gan> Ganchos()
{
    List<gan> _lista = new List<gan>();
    MySqlCommand _comando = new MySqlCommand(String.Format(
        "SELECT `Id_gancho`,`Precio_unidad`,`Existencia` FROM `Gancho`"),
ObtenerConexion());
    MySqlDataReader _reader = _comando.ExecuteReader();
    while (_reader.Read())
    {
        gan pin = new gan();

        pin.id = _reader.GetInt32(0);
        pin.precio_unit = _reader.GetFloat(1);
        pin.Existencia = _reader.GetInt32(2);

        _lista.Add(pin);
    }
    return _lista;
}

public static List<cliente> Cliente()
{
    List<cliente> _lista = new List<cliente>();
    MySqlCommand _comando = new MySqlCommand(String.Format(
        "SELECT `Id_cliente`,`Nombre`,`Apellidos_P`,`Apellidos_M`,`Telefono`
FROM `cliente`"), ObtenerConexion());
    MySqlDataReader _reader = _comando.ExecuteReader();
    while (_reader.Read())
    {
        cliente pin = new cliente();

        pin.id = _reader.GetInt32(0);
        pin.Nombre = _reader.GetString(1);
        pin.Apellido_P = _reader.GetString(2);
        pin.Apellido_M = _reader.GetString(3);
        pin.Telefono = _reader.GetInt32(4);

        _lista.Add(pin);
    }
    return _lista;
}

public static List<lote> lote()
{
    List<lote> _lista = new List<lote>();
    MySqlCommand _comando = new MySqlCommand(String.Format(

```



```

        "SELECT `Id_lote`,`Id_pintura`,`id_carton` ,
        `Id_gancho`,`Cantidad_esferas`,`Tamaño_cm`,`Invercion` FROM `Lote`"),
ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {

            lote pin = new lote();

            pin.id_lote = _reader.GetInt32(0);
            pin.id_pint = _reader.GetInt32(1);
            pin.id_carton = _reader.GetInt32(2);
            pin.id_gancho = _reader.GetInt32(3);
            pin.cantidad = _reader.GetInt32(4);
            pin.tamaño = _reader.GetInt32(5);
            pin.inversion = _reader.GetDouble(6);
            _lista.Add(pin);
        }
        return _lista;
    }

    public static List<esferas> esferas()
    {
        List<esferas> _lista = new List<esferas>();
        MySqlCommand _comando = new MySqlCommand(String.Format(
            "SELECT * FROM `Esferas`"), ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {

            esferas pin = new esferas();

            pin.Id_esferas = _reader.GetInt32(0);
            pin.Existencia = _reader.GetInt32(1);
            pin.id_lote = _reader.GetInt32(2);
            pin.Precio_unitario = _reader.GetDouble(3);
            _lista.Add(pin);
        }
        return _lista;
    }

    public static List<cliente> buscar(int fa)
    {
        List<cliente> _lista = new List<cliente>();
        MySqlCommand _comando = new MySqlCommand(String.Format(
            "SELECT * FROM `cliente` WHERE `Id_cliente` = {0}", fa),
ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {

            cliente pin = new cliente();

            pin.id = _reader.GetInt32(0);
            pin.Nombre = _reader.GetString(1);
            pin.Apellido_P = _reader.GetString(2);
            pin.Apellido_M = _reader.GetString(3);
            pin.Telefono = _reader.GetInt32(4);
        }
    }

```

```

        _lista.Add(pin);
    }
    return _lista;
}
public static List<cliente> id_clie(ComboBox c)
{
    List<cliente> _lista = new List<cliente>();
    MySqlCommand _comando = new MySqlCommand(String.Format(
        "SELECT `Id_cliente` FROM `cliente`"), ObtenerConexion());
    MySqlDataReader _reader = _comando.ExecuteReader();
    while (_reader.Read())
    {
        cliente pin = new cliente();

        c.Items.Add(_reader.GetInt32(0));

        _lista.Add(pin);
    }
    return _lista;
}

}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MySql.Data.MySqlClient;
using System.Windows.Forms;

namespace class_esferas
{
    public class pintura
    {
        public double precio { get; set; }
    }
    public class infor_unit
    {
        public void pre_Pintura(int pin, Label l, Label lc)
        {
            MySqlCommand _comando = new MySqlCommand(String.Format("SELECT
`Precio`,`Color` FROM `Pintura` WHERE `Id_pintura` = '{0}'", pin),
conec.ObtenerConexion());
            MySqlDataReader _reader = _comando.ExecuteReader();
            while (_reader.Read())
            {
                l.Text = _reader.GetString(0);
                lc.Text = _reader.GetString(1);
            }
        }
        public void carto(int car, Label l)
        {

```

```

        MySqlCommand _comando = new MySqlCommand(String.Format("SELECT `Precio`
FROM `Carton` WHERE `id_carton` = '{0}'", car), conec.ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {
            l.Text = _reader.GetString(0);
        }
    }
    public void Gancho(int pin, Label l)
    {
        MySqlCommand _comando = new MySqlCommand(String.Format("SELECT
`Precio_unidad` * `Existencia` FROM `Gancho` WHERE `Id_gancho` = '{0}'", pin),
conec.ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {
            l.Text = _reader.GetString(0);
        }
    }
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MySql.Data.MySqlClient;
namespace class_esferas
{
    public class empleado
    {
        public int id { get; set; }
        public string Nombre { get; set; }
        public string Apellido_P { get; set; }
        public string Apellido_M { get; set; }
        public int Telefono { get; set; }
    }

    public class Person
    {
        public static MySqlConnection ObtenerConexion()
        {
            MySqlConnection conectar = new MySqlConnection("server=69.162.95.98;
database=eliarome33807DB; Uid=eliarome33807; pwd=x6aCKX20nTDb;");

            conectar.Open();
            return conectar;
        }
        public static List<empleado> Empleado()
        {

```

```

        List<empleado> _lista = new List<empleado>();
        MySqlCommand _comando = new MySqlCommand(String.Format(
            "SELECT `Id_Empleado`, `Nombre`, `Apellidos_P`, `Apellidos_M`, `Telefono`
FROM `Empleado`"), ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {

            empleado pin = new empleado();

            pin.id = _reader.GetInt32(0);
            pin.Nombre = _reader.GetString(1);
            pin.Apellido_P = _reader.GetString(2);
            pin.Apellido_M = _reader.GetString(3);
            pin.Telefono = _reader.GetInt32(4);

            _lista.Add(pin);
        }
        return _lista;
    }
    public static List<empleado> Buscar_emp(int fa)
    {
        List<empleado> _lista = new List<empleado>();
        MySqlCommand _comando = new MySqlCommand(String.Format(
            "SELECT * FROM `Empleado` WHERE `Id_Empleado` = {0}", fa),
ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {

            empleado pin = new empleado();

            pin.id = _reader.GetInt32(0);
            pin.Nombre = _reader.GetString(1);
            pin.Apellido_P = _reader.GetString(2);
            pin.Apellido_M = _reader.GetString(3);
            pin.Telefono = _reader.GetInt32(4);

            _lista.Add(pin);
        }
        return _lista;
    }
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;
using System.Windows.Forms;

```

```

namespace Class_Esfera
{
    public class datos
    {
        public MySqlDataReader dr { get; set; }
        public MySqlCommand cadena_sql { set; get; }
        MySqlDataAdapter adapt;
        MySqlConnection conn;

        public bool Conectar()
        {
            conn = new MySqlConnection();
            conn.ConnectionString = "server=69.162.95.98; database=eliarome33807DB;
            Uid=eliarome33807; pwd=x6aCKX20nTDb";
            try
            {
                conn.Open();
                return true;
            }
            catch (Exception error)
            {
                MessageBox.Show("Error al conectarse a la base de datos:" +
                error.Message);
                return false;
            }
        }
        public void desconectar()
        {
            conn.Close();
        }
        public void construye_reader(string cadena)
        {
            cadena_sql = new MySqlCommand();
            cadena_sql.Connection = conn;
            cadena_sql.CommandText = cadena;
        }
        public MySqlDataReader ejecuta_reader()
        {
            try
            {
                dr = cadena_sql.ExecuteReader();
                return dr;
            }
            catch (Exception error)
            {
                MessageBox.Show("Error al ejecutar el reader" + error.Message);
                return null;
            }
        }
        public MySqlCommand construye_command(string cadena)
        {
            cadena_sql = new MySqlCommand(cadena, conn);
            return cadena_sql;
        }
        public int ejecutanonquery()
    }
}

```

```

{
    int afectados;
    try
    {
        afectados = cadena_sql.ExecuteNonQuery();
        return afectados;
    }
    catch (Exception error)
    {
        MessageBox.Show("Error al ejecutar el reader" + error.Message);
        return 0;
    }
}
public MySqlDataAdapter construye_adapter(string cadena)
{
    adapt = new MySqlDataAdapter(cadena, conn);
    return adapt;
}
public DataRow extrae_registro(MySqlDataAdapter adapter, string tabla)
{
    DataSet ds = new DataSet();
    DataRow fila;
    try
    {
        adapter.Fill(ds, tabla);
        DataTable miTabla = ds.Tables[tabla];
        fila = miTabla.Rows[0];
        return fila;
    }
    catch (Exception error)
    {
        MessageBox.Show("Error al extraer el registro:" + error.Message);
        return null;
    }
}
public DataSet consultarDataSet(string consulta)
{
    DataSet ds = new DataSet();
    MySqlCommand comando = new MySqlCommand();
    MySqlDataAdapter adapter = new MySqlDataAdapter();
    comando.CommandText = consulta;
    comando.Connection = conn;
    adapter.SelectCommand = comando;
    try
    {
        adapter.Fill(ds);
        return ds;
    }
    catch (Exception error)
    {
        MessageBox.Show("No se pudo crear el dataset:" + error.Message);
        return null;
    }
}
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;
using System.Windows.Forms;

namespace Class_Esfera
{
    public class Cliente
    {
        public int Id_cl { get; set; }
        public String Nombre { get; set; }
        public String Apellido_p { get; set; }
        public String Apellido_m { get; set; }
        public int Telefono { get; set; }

        public Cliente()
        {
            Id_cl = 0;
            Nombre = "";
            Apellido_p = "";
            Apellido_m = "";
            Telefono = 0;
        }

        public Cliente(int ip, String nom, String app, String apm, int tel)
        {
            Id_cl = ip;
            Nombre = nom;
            Apellido_p = app;
            Apellido_m = apm;
            Telefono = tel;
        }

        public int Alta_Cliente()
        {
            MySqlCommand Coman;
            datos inserta = new datos();
            int regresa = 0;

            if (inserta.Conectar())
            {
                String comando = "Insert into
cliente(Id_cliente,Nombre,Apellidos_p,Apellidos_m,Telefono)"
+ "values (" + Id_cl + "," + Nombre + "," + Apellido_p +
",'" + Apellido_m + "','" + Telefono + "');"

                Coman = inserta.construye_command(comando);

                if ((inserta.ejecutanonquery()) != 0)
                    regresa = 1;

                else
                    regresa = 0;
            }
        }
    }
}

```

```

        Coman.Connection.Close();
        inserta.desconectar();
    }

    else
        regresa = -1;

    return regresa;
}

}

public class Pintura
{
    public int Id_pintura { get; set; }
    public String Color { get; set; }
    public double Precio { get; set; }

    public Pintura()
    {
        Id_pintura = 0;
        Color = "";
        Precio = 0;
    }

    public Pintura(int idp, String col, double pre)
    {
        Id_pintura = idp;
        Color = col;
        Precio = pre;
    }

    public int Alta_Pintura()
    {
        MySqlCommand Coman;
        datos inserta = new datos();
        int regresa = 0;

        if (inserta.Conectar())
        {
            String ingresa = "Insert into Pintura(Id_pintura,Color,Precio)"
                + "values(" + Id_pintura + "," + Color + "," + Precio + ")";

            Coman = inserta.construye_command(ingresa);

            if ((inserta.ejecutanonquery()) != 0)
                regresa = 1;

            else
                regresa = 0;

            Coman.Connection.Close();
            inserta.desconectar();
        }

        else
            regresa = -1;

        return regresa;
    }
}

```



```

    }

    public int Carga_Id(ComboBox cmb)
    {
        datos selecciona = new datos();
        MySqlDataReader lector;

        if (selecciona.Conectar())
        {
            selecciona.construye_reader("select Id_pintura from Pintura");
            lector = selecciona.ejecuta_reader();

            while (lector.Read())
                cmb.Items.Add(lector.GetInt32(0).ToString());

            selecciona.desconectar();
            selecciona.dr.Close();

            return 1;
        }

        else
            return -1;
    }
}

public class Carton
{
    public int Id_carton { get; set; }
    public double Precio { get; set; }
    public int Tamaño { get; set; }

    public Carton()
    {
        Id_carton = 0;
        Precio = 0;
        Tamaño = 0;
    }

    public Carton(int ic, double pre, int tam)
    {
        Id_carton = ic;
        Precio = pre;
        Tamaño = tam;
    }

    public int Alta_Carton()
    {
        MySqlCommand Coman;
        datos inserta = new datos();
        int regresa = 0;

        if (inserta.Conectar())
        {
            String ingresa = "Insert into Carton(Id_carton,Precio,Tamaño)"
                + "values (" + Id_carton + "," + Precio + "," + Tamaño +
"
```

```

        Coman = inserta.construye_command(ingresa);

        if ((inserta.ejecutanonquery()) != 0)
            regresa = 1;

        else
            regresa = 0;

        Coman.Connection.Close();
        inserta.desconectar();
    }

    else
        regresa = -1;

    return regresa;
}

public int Carga_Id(ComboBox cmb)
{
    datos selecciona = new datos();
    MySqlDataReader lector;

    if (selecciona.Conectar())
    {
        selecciona.construye_reader("select Id_carton from Carton");
        lector = selecciona.ejecuta_reader();

        while (lector.Read())
            cmb.Items.Add(lector.GetInt32(0).ToString());

        selecciona.desconectar();
        selecciona.dr.Close();

        return 1;
    }

    else
        return -1;
}

}

public class Gancho
{
    public int Id_gancho { get; set; }
    public double Precio { get; set; }
    public int Existencia { get; set; }

    public Gancho()
    {
        Id_gancho = 0;
        Precio = 0;
        Existencia = 0;
    }

    public Gancho(int ig, double pre, int exi)
    {
        Id_gancho = ig;
    }
}

```

```

        Precio = pre;
        Existencia = exi;
    }

    public int Alta_Gancho()
    {
        MySqlCommand Coman;
        datos inserta = new datos();
        int regresa = 0;

        if (inserta.Conectar())
        {
            String ingresa = "Insert into
Gancho(Id_gancho,Precio_unidad,Exixtencia)"
            + "values (" + Id_gancho + "," + "0." + Precio + "," + Existencia + ")";

            Coman = inserta.construye_command(ingresa);

            if ((inserta.ejecutanonquery()) != 0)
                regresa = 1;

            else
                regresa = 0;

            Coman.Connection.Close();
            inserta.desconectar();
        }

        else
            regresa = -1;

        return regresa;
    }

    public int Carga_Id(ComboBox cmb)
    {
        datos selecciona = new datos();
        MySqlDataReader lector;

        if (selecciona.Conectar())
        {
            selecciona.construye_reader("select Id_gancho from Gancho");
            lector = selecciona.ejecuta_reader();

            while (lector.Read())
                cmb.Items.Add(lector.GetInt32(0).ToString());

            selecciona.desconectar();
            selecciona.dr.Close();

            return 1;
        }

        else
            return -1;
    }
}

```

```

public class Empleado
{
    public int Id_empleado { get; set; }
    public String Nombre { get; set; }
    public String App { get; set; }
    public String Apm { get; set; }
    public int Telefono { get; set; }

    public Empleado()
    {
        Id_empleado = 0;
        Nombre = "";
        App = "";
        Apm = "";
        Telefono = 0;
    }

    public Empleado(int id, String nom, String ap, String am, int tel)
    {
        Id_empleado = id;
        Nombre = nom;
        App = ap;
        Apm = am;
        Telefono = tel;
    }

    public int Alta_Empleado()
    {
        MySqlCommand Coman;
        datos inserta = new datos();
        int regresa = 0;

        if (inserta.Conectar())
        {
            String comando = "Insert into
Empleado(Id_Empleado,Nombre,Apellidos_P,Apellidos_M,Telefono)"
+ "values (" + Id_empleado + "," + Nombre + "," + App + "," +
+ Apm + "," + Telefono + ")";

            Coman = inserta.construye_command(comando);

            if ((inserta.ejecutanonquery()) != 0)
                regresa = 1;

            else
                regresa = 0;

            Coman.Connection.Close();
            inserta.desconectar();
        }

        else
            regresa = -1;

        return regresa;
    }
}

```

```

public class Lote
{
    public int idLote { get; set; }
    public int idPintura { get; set; }
    public int idGancho { get; set; }
    public int idCarton { get; set; }
    public int CantEsferas { get; set; }
    public int Tamaño { get; set; }
    public double Invercion { get; set; }

    public Lote()
    {
        idLote = 0;
        idPintura = 0;
        idGancho = 0;
        idCarton = 0;
        CantEsferas = 0;
        Tamaño = 0;
        Invercion = 0;
    }

    public Lote(int idl, int idp, int idg, int idc, int cant, int tam, int inv)
    {
        idLote = idl;
        idPintura = idp;
        idGancho = idg;
        idCarton = idc;
        CantEsferas = cant;
        Tamaño = tam;
        Invercion = inv;
    }

    public int Alta_Lote()
    {
        MySqlCommand Coman;
        datos inserta = new datos();
        int regresa = 0;

        if (inserta.Conectar())
        {
            String ingresa = "Insert into
Lote(Id_lote,Id_pintura,id_carton,Id_gancho,Cantidad_esferas,Tamaño_cm,Invercion)"
+ "values (" + idLote + "," + idPintura + "," + idCarton +
",'" + idGancho + "'," + CantEsferas + "," +
+ Tamaño + "," + Invercion + ")";

            Coman = inserta.construye_command(ingresa);

            if ((inserta.ejecutanonquery()) != 0)
                regresa = 1;

            else
                regresa = 0;

            Coman.Connection.Close();
            inserta.desconectar();
        }
    }
}

```

```

        else
            regresa = -1;

        return regresa;
    }
}

public class Esferas_mar
{
    public string descripcion { get; set; }
    public int id_lote { get; set; }
    public double Precio_unitario { get; set; }
    public double Precio_sin_box { get; set; }

    public int Alta_esferas()
    {
        MySqlCommand Coman;
        datos inserta = new datos();
        int regresa = 0;

        if (inserta.Conectar())
        {
            String ingresa = "INSERT INTO
Esferas(descripcion,Id_lote,precio_unitario,Precio_sin_box)"
+ "values ('" + descripcion + "','" + id_lote + "','" +
Precio_unitario + "','" + Precio_sin_box + "')";

            Coman = inserta.construye_command(ingresa);

            if ((inserta.ejecutanonquery()) != 0)
                regresa = 1;

            else
                regresa = 0;

            Coman.Connection.Close();
            inserta.desconectar();
        }

        else
            regresa = -1;

        return regresa;
    }
}

public class compra
{
    public static MySqlConnection ObtenerConexion()
    {
        MySqlConnection conectar = new MySqlConnection("server=69.162.95.98;
database=eliarome33807DB; Uid=eliarome33807; pwd=x6aCKX20nTDb;");

        conectar.Open();
        return conectar;
    }
}

```

```

public int id_esfera { get; set; }
public int cantidad { get; set; }
public double subtotal { get; set; }

public int Alta_Compra(int id_fact)
{
    MySqlCommand Coman;
    datos inserta = new datos();
    int regresa = 0;

    if (inserta.Conectar())
    {
        String ingresa = "INSERT INTO `eliarome33807DB`.`compra`
(`Id_Fatura`, `Id_esferas`, `cantidad`, `subtotal`) VALUES ('" + id_fact + "', '" +
id_esfera + "', '" + cantidad + "', '" + subtotal + "')";

        Coman = inserta.construye_command(ingresa);

        if ((inserta.ejecutanonquery()) != 0)
            regresa = 1;

        else
            regresa = 0;

        Coman.Connection.Close();
        inserta.desconectar();
    }

    else
        regresa = -1;

    return regresa;
}
public void suma(int fa, Label l1, Label l2, Label l3)
{
    double sum = 0;
    MySqlCommand _comando = new MySqlCommand(String.Format(
"SELECT SUM(`subtotal`) FROM `compra` WHERE `Id_Fatura` = {0}", fa),
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 canti(int fa, Label cs)
{
    MySqlCommand _comando = new MySqlCommand(String.Format(

```

```

        "SELECT SUM(`cantidad`) FROM `compra` WHERE `Id_Fatura` = {0}", fa),
ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {
            csns.Text = _reader.GetString(0);
        }

    }
    public void ESPESIFIC(int fa,ListBox L)
    {
        MySqlCommand _comando = new MySqlCommand(String.Format(
            "SELECT `descripcion` FROM `Esferas` WHERE `Id_esferas` IN ( SELECT
`Id_esferas` FROM `compra` WHERE `Id_Fatura` = {0} )", fa), ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {
            L.Items.Add(_reader.GetString(0));
        }
    }
    public void UNIO(int fa,ListBox l2,ListBox l3)
    {
        MySqlCommand _comando = new MySqlCommand(String.Format(
            "SELECT `cantidad`,`subtotal` FROM `compra` WHERE `Id_Fatura` = {0}",
fa), ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {
            l2.Items.Add(_reader.GetInt32(0));
            l3.Items.Add(_reader.GetFloat(1));
        }
    }
    public static List<compra> carrito(int fa)
    {
        List<compra> _lista = new List<compra>();
        MySqlCommand _comando = new MySqlCommand(String.Format(
            "SELECT `Id_esferas`,`cantidad`,`subtotal` FROM `compra` WHERE
`Id_Fatura` = {0}",fa), ObtenerConexion());
        MySqlDataReader _reader = _comando.ExecuteReader();
        while (_reader.Read())
        {
            compra pin = new compra();

            pin.id_esfera = _reader.GetInt32(0);
            pin.cantidad= _reader.GetInt32(1);
            pin.subtotal = _reader.GetFloat(2);

            _lista.Add(pin);
        }
        return _lista;
    }
    public int cancelar(int fa)

```



```

{
    MySqlCommand Coman;
    datos inserta = new datos();
    int regresa = 0;

    if (inserta.Conectar())
    {
        String comando = "DELETE FROM `compra` WHERE `Id_Fatura` = "+fa;

        Coman = inserta.construye_command(comando);

        if ((inserta.ejecutanonquery()) != 0)
            regresa = 1;

        else
            regresa = 0;

        Coman.Connection.Close();
        inserta.desconectar();
    }

    else
        regresa = -1;

    return regresa;
}

}

public class factura
{
    public int Id_Fatura { get; set; }
    public int Id_Empleado { get; set; }
    public int Id_cliente { get; set; }
    public string Fecha { get; set; }
    public int Cantidad { get; set; }
    public double Subtotal { get; set; }
    public double Total { get; set; }

    public int alta_fact()
    {
        MySqlCommand Coman;
        datos inserta = new datos();
        int regresa = 0;
        if (inserta.Conectar())
        {
            String ingresa = "INSERT INTO `eliarome33807DB`.`Factura` (`Id_Fatura`,
            `Id_Empleado`, `Id_cliente`, `Fecha`, `Cantidad`, `Subtotal`, `Total`) VALUES ('" +
            Id_Fatura + "','" + Id_Empleado + "','" + Id_cliente + "','" + Fecha + "','" +
            Cantidad + "','" + Subtotal + "','" + Total + "')";

            Coman = inserta.construye_command(ingresa);

            if ((inserta.ejecutanonquery()) != 0)
                regresa = 1;

            else
                regresa = 0;

            Coman.Connection.Close();

```

```

        inserta.desconectar();
    }

    else
        regresa = -1;

    return regresa;
}
}
}

```

CODIGO FORMULARIO

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Esferas
{
    public partial class frmMenu : Form
    {
        public frmMenu()
        {
            InitializeComponent();
        }

        private void ventasToolStripMenuItem_Click(object sender, EventArgs e)
        {
            venta obj= new venta();
            obj.Show();
        }

        private void consultasToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Consultas obj2 = new Consultas();
            obj2.Show();
        }

        private void registrosToolStripMenuItem_Click(object sender, EventArgs e)
        {
            frmPrincipal obj3 = new frmPrincipal();
            obj3.Show();
        }

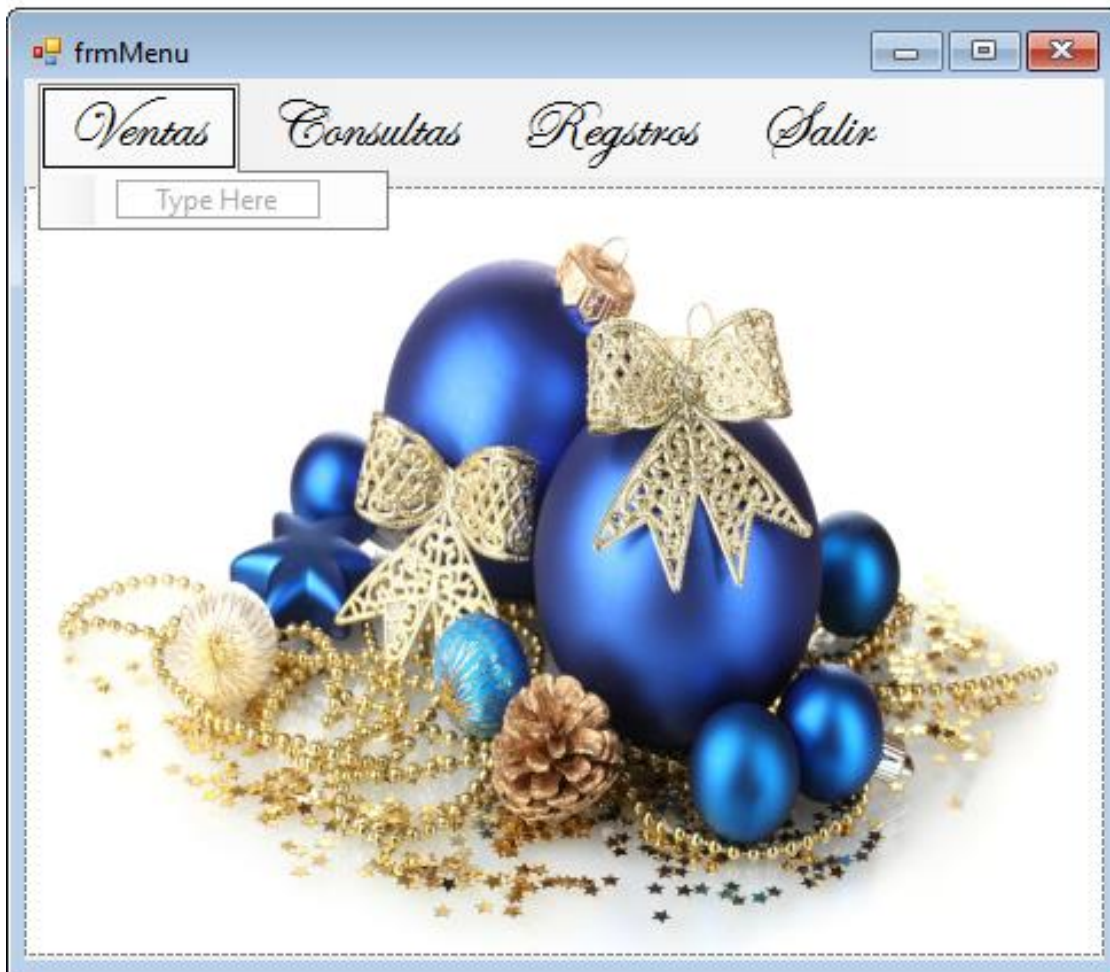
        private void salirToolStripMenuItem_Click(object sender, EventArgs e)

```

```

        {
            Application.Exit();
        }
    }
}

```



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

```

```

using class_esferas;
using Class_Esfera;

namespace Esferas
{
    public partial class Consultas : Form
    {
        public Consultas()
        {
            InitializeComponent();
        }

        private void rdbpintura_CheckedChanged(object sender, EventArgs e)
        {
            DTpintura.Visible = true;
            DTBgancho.Visible = false;
            DTBcarton.Visible = false;
        }

        private void rdbcarton_CheckedChanged(object sender, EventArgs e)
        {
            DTpintura.Visible = false;
            DTBgancho.Visible = false;
            DTBcarton.Visible = true;
        }

        private void rdbgancho_CheckedChanged(object sender, EventArgs e)
        {
            DTBgancho.DataSource = conec.Ganchos();
            DTpintura.Visible = false;
            DTBgancho.Visible = true;
            DTBcarton.Visible = false;
        }

        private void Consultas_Load(object sender, EventArgs e)
        {
            // TODO: esta línea de código carga datos en la tabla
            'eliarome33807DBDataSet.Pintura' Puede moverla o quitarla según sea necesario.
            this.pinturaTableAdapter.Fill(this.eliarome33807DBDataSet.Pintura);
            // TODO: esta línea de código carga datos en la tabla
            'eliarome33807DBDataSet1.Esferas' Puede moverla o quitarla según sea necesario.
            this.esferasTableAdapter1.Fill(this.eliarome33807DBDataSet1.Esferas);

            // TODO: esta línea de código carga datos en la tabla
            'eliarome33807DBDataSet.Factura' Puede moverla o quitarla según sea necesario.
            this.facturaTableAdapter.Fill(this.eliarome33807DBDataSet.Factura);
            // TODO: esta línea de código carga datos en la tabla
            'eliarome33807DBDataSet.Carton' Puede moverla o quitarla según sea necesario.
            this.cartonTableAdapter.Fill(this.eliarome33807DBDataSet.Carton);

            dtbcliente.DataSource = conec.Cliente();
            DTBempleado.DataSource = Person.Empleado();
            DTpintura.Visible = false;
            DTBgancho.Visible = false;
            DTBcarton.Visible = false;
        }
    }
}

```

```

private void btmactualizar_Click(object sender, EventArgs e)
{
    DTBempleado.DataSource = Person.Empleado();
}

private void button1_Click(object sender, EventArgs e)
{
    dtbcliente.DataSource = conec.Cliente();
}

private void button2_Click(object sender, EventArgs e)
{
    Dtblote.DataSource = conec.lote();
}

private void Buscar_Click(object sender, EventArgs e)
{
}

private void button3_Click(object sender, EventArgs e)
{
    frmPrincipal op = new frmPrincipal();
    op.Show();
    this.Hide();
}

private void cartonBindingSource_CurrentChanged(object sender, EventArgs e)
{
}

private void Lotes_Click(object sender, EventArgs e)
{
}

private void button3_Click_1(object sender, EventArgs e)
{
    compra com = new compra();
    listBox1.Items.Clear();
    listBox2.Items.Clear();
    listBox3.Items.Clear();
    com.ESPECIFIC(Convert.ToInt32(id_FaturaLabel1.Text),listBox1);
    com.UNIO(Convert.ToInt32(id_FaturaLabel1.Text), listBox2, listBox3);
    dataGridView2.DataSource =
conec.buscar(Convert.ToInt32(id_clienteLabel1.Text));
    dataGridView3.DataSource =
Person.Buscar_emp(Convert.ToInt32(id_EmpleadoLabel1.Text));
}
}
}

```



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Class_Esfera;
using class_esferas;

namespace Esferas
{
    public partial class frmPrincipal : Form
    {
    }
```

```

Esferas_mar objesfer = new Esferas_mar();
Lote objlote = new Lote();
Pintura objPintura = new Pintura();
Carton objCarton = new Carton();
Gancho objGancho = new Gancho();
Cliente pCliente = new Cliente();
Empleado objEmpleado = new Empleado();

public frmPrincipal()
{
    InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{
    //pCliente.Id_cl = Convert.ToInt32(null);
    pCliente.Nombre = txtNombreCL.Text;
    pCliente.Apellido_p = txtApellidoP.Text;
    pCliente.Apellido_m = txtApellidoM.Text;
    pCliente.Telefono = Convert.ToInt32(txtTelefono.Text);

    int r = pCliente.Alta_Cliente();

    if (r == 1)
        MessageBox.Show("Los datos se agregaron correctamente");

    else
        if (r == 0)
            MessageBox.Show("No se agregaron los datos del empleado");

        else
            MessageBox.Show("Error en la base de datos");
}

private void btnAgregarE_Click(object sender, EventArgs e)
{
    //objEmpleado.Id_empleado = Convert.ToInt32(null);
    objEmpleado.Nombre = txtEmpladoN.Text;
    objEmpleado.App = txtAppE.Text;
    objEmpleado.Apm = txtEmpleadoAPM.Text;
    objEmpleado.Telefono = Convert.ToInt32(txtTelefonoE.Text);

    int r = objEmpleado.Alta_Empleado();

    if (r == 1)
        MessageBox.Show("Los datos se agregaron correctamente");

    else
        if (r == 0)
            MessageBox.Show("No se agregaron los datos del empleado");

        else
            MessageBox.Show("Error en la base de datos");
}

private void btnGancho_Click_1(object sender, EventArgs e)
{
    objGancho.Existencia = Convert.ToInt32(nudCantidad.Value);
}

```

```

objGancho.Precio = Convert.ToDouble(txtGancho.Text);

int r = objGancho.Alta_Gancho();

if (r == 1)
    MessageBox.Show("Datos guardados");

else
    if (r == 0)
        MessageBox.Show("Datos no guardados");

    else
        MessageBox.Show("Error en la base de datos");
}

private void Form1_Load_1(object sender, EventArgs e)
{
    cmbColor.Items.Add("Rojo");
    cmbColor.Items.Add("Amarillo");
    cmbColor.Items.Add("Verde");
    cmbColor.Items.Add("Azul");

    int r = objPintura.Carga_Id(cmbIdPintura);
    int j = objGancho.Carga_Id(cmbIdGancho);
    int k = objCarton.Carga_Id(cmbIDCarton);

    if (r == -1 && j == -1 && k == -1)
        MessageBox.Show("No se encontraron datos");
}

private void btnPintura_Click_1(object sender, EventArgs e)
{
    objPintura.Id_pintura = Convert.ToInt32(null);
    objPintura.Color = cmbColor.Text;
    objPintura.Precio = Convert.ToDouble(txtPintura.Text);

    int resp = objPintura.Alta_Pintura();

    if (resp == 1)
        MessageBox.Show("Los datos se agregaron correctamente");

    else
        if (resp == 0)
            MessageBox.Show("No se agregaron los datos");

        else
            MessageBox.Show("Error en la base de datos");
}

private void btnCarton_Click(object sender, EventArgs e)
{
    objCarton.Id_carton = Convert.ToInt32(null);

    if (rbTseis.Checked)
    {
        objCarton.Tamaño = 6;
    }
}

```



```

else
    if (rbTdoce.Checked)
    {
        objCarton.Tamaño = 12;
    }

    else
    {
        objCarton.Tamaño = 24;
    }

objCarton.Precio = Convert.ToDouble(txtCarton.Text);

int resp = objCarton.Alta_Carton();

if (resp == 1)
    MessageBox.Show("Los datos se agregaron correctamente");

else
    if (resp == 0)
        MessageBox.Show("No se agregaron los datos");

    else
        MessageBox.Show("Error en la base de datos");
}

private void cmbIdPintura_SelectedIndexChanged(object sender, EventArgs e)
{
    if (Convert.ToInt32(cmbIdPintura.Text) >= 0)
    {
        infor.pre_Pintura(Convert.ToInt32(cmbIdPintura.Text), lblpin,
lblcolor);
    }
}

infor_unit infor = new infor_unit();
private void cmbIdGancho_SelectedIndexChanged(object sender, EventArgs e)
{
    if (Convert.ToInt32(cmbIdGancho.Text) >= 0)
    {
        infor.Gancho(Convert.ToInt32(cmbIdGancho.Text), lblganch);
    }
}

private void cmbIDCarton_SelectedIndexChanged(object sender, EventArgs e)
{
    if (Convert.ToInt32(cmbIDCarton.Text) >= 0)
    {
        infor.carto(Convert.ToInt32(cmbIDCarton.Text), blcart);
    }
}

private void linkLabel1_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    Consultas frm = new Consultas();
    frm.ShowDialog();
}

```

```

private void tabPage2_Click(object sender, EventArgs e)
{
}

private void tabPage2_MouseMove(object sender, MouseEventArgs e)
{
    if (cmbIDCarton.Text != "")
    {
        if (cmbIdGancho.Text != "")
        {
            if (cmbIdPintura.Text != "")
            {
                double a, b, c;
                a = Convert.ToDouble(lblganch.Text);
                b = Convert.ToDouble(blcart.Text);
                c = Convert.ToDouble(lblpin.Text);
                lblinver.Text = Convert.ToString(a + b + c);
            }
        }
    }
}

private void button1_Click_1(object sender, EventArgs e)
{
    objlote.idLote = Convert.ToInt32(textBox4.Text);
    objlote.idPintura = Convert.ToInt32(cmbIdPintura.Text);
    objlote.idCarton = Convert.ToInt32(cmbIDCarton.Text);
    objlote.idGancho = Convert.ToInt32(cmbIdGancho.Text);
    objlote.CantEsferas = Convert.ToInt32(txtCantidaL.Text);
    objlote.Tamaño = Convert.ToInt32(lbltam.Text);
    objlote.Invercion = Convert.ToDouble(lblinver.Text);

    int r = objlote.Alta_Lote();

    if (r == 1)
    {
        MessageBox.Show("Lote registrado");
        string descr = "Esfera color: " + lblcolor.Text + " Tamaño " +
lbltam.Text;
        lblsce.Text = descr;

        objesfer.descripcion = lblsce.Text;
        objesfer.id_lote = Convert.ToInt32(textBox4.Text);
        objesfer.Precio_unitario = (Convert.ToDouble(lblinver.Text) /
Convert.ToDouble(txtCantidaL.Text));
        double ca_pre = (Convert.ToDouble(lblpin.Text) +
Convert.ToDouble(blcart.Text)) / Convert.ToDouble(txtCantidaL.Text);
        objesfer.Precio_sin_box = ca_pre;
        objesfer.Alta_esferas();
    }
    else
        if (r == 0)
            MessageBox.Show("Lote no registrado");
        else

```

```

        MessageBox.Show("Error en el servidor");
    }
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Class_Esfera;

namespace Esferas
{

```

```

public partial class venta : Form
{
    public venta()
    {
        InitializeComponent();
    }

    private void clienteBindingNavigatorSaveItem_Click(object sender, EventArgs
e)
    {
        this.Validate();
        this.clienteBindingSource.EndEdit();
        this.tableAdapterManager.UpdateAll(this.eliarome33807DBDataSet);

    }

    private void venta_Load(object sender, EventArgs e)
    {
        // TODO: esta línea de código carga datos en la tabla
        'eliarome33807DBDataSet.Esferas' Puede moverla o quitarla según sea necesario.
        this.esferasTableAdapter1.Fill(this.eliarome33807DBDataSet.Esferas);
        // TODO: esta línea de código carga datos en la tabla
        'eliarome33807DBDataSet1.Esferas' Puede moverla o quitarla según sea necesario.

        // TODO: esta línea de código carga datos en la tabla
        'eliarome33807DBDataSet.compra' Puede moverla o quitarla según sea necesario.
        this.compraTableAdapter.Fill(this.eliarome33807DBDataSet.compra);
        // TODO: esta línea de código carga datos en la tabla
        'eliarome33807DBDataSet.compra' Puede moverla o quitarla según sea necesario.
        this.compraTableAdapter.Fill(this.eliarome33807DBDataSet.compra);

        this.empleadoTableAdapter.Fill(this.eliarome33807DBDataSet.Empleado);
        // TODO: esta línea de código carga datos en la tabla
        'eliarome33807DBDataSet.cliente' Puede moverla o quitarla según sea necesario.
        this.clienteTableAdapter.Fill(this.eliarome33807DBDataSet.cliente);
        // TODO: esta línea de código carga datos en la tabla
        'eliarome33807DBDataSet.cliente' Puede moverla o quitarla según sea necesario.
        this.clienteTableAdapter.Fill(this.eliarome33807DBDataSet.cliente);

    }

    private void clienteBindingNavigatorSaveItem_Click_1(object sender,
EventArgs e)
    {
        this.Validate();
        this.clienteBindingSource.EndEdit();
        this.tableAdapterManager.UpdateAll(this.eliarome33807DBDataSet);

    }

    private void clienteBindingNavigator_RefreshItems(object sender, EventArgs
e)
    {

    }

    private void id_EmpleadoNumericUpDown_ValueChanged(object sender, EventArgs
e)

```

```

{
}

private void groupBox1_Enter(object sender, EventArgs e)
{
}

private void groupBox2_Enter(object sender, EventArgs e)
{
}

private void compraDataGridView_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
}

compra fac = new compra();
private void button1_Click(object sender, EventArgs e)
{

    double pre = Convert.ToDouble(precio_unitarioLabel1.Text);
    double cant = Convert.ToDouble(cantidadTextBox.Text);

    double resp = 0;

    resp = pre * cant;

    lblImporte.Text = resp.ToString();

    int id_fac = Convert.ToInt32(id_FaturaTextBox.Text);
    fac.id_esfera = Convert.ToInt32(id_esferasLabel1.Text);
    fac.cantidad = Convert.ToInt32(cantidadTextBox.Text);

    int c = Convert.ToInt32(cantidadTextBox.Text);
    double n = c * Convert.ToDouble(precio_unitarioLabel1.Text);
    fac.subtotal = n;

    fac.Alta_Compra(Convert.ToInt32(id_FaturaTextBox.Text));
    compraDataGridView.DataSource =
compra.carrito(Convert.ToInt32(id_FaturaTextBox.Text));
    fac.suma(Convert.ToInt32(id_FaturaTextBox.Text), lblsun, lbliva,
lbltol);
    fac.canti(Convert.ToInt32(id_FaturaTextBox.Text), canti);
    //cantidadTextBox.Text = "0";
    numericUpDown1.Value = 0;

    lblfactura.Text = id_FaturaTextBox.Text;
    lblFecha.Text = fech.Value.ToShortDateString();
    lblCliente.Text = nombreLabel1.Text + apellidos_PLabel1.Text +
apellidos_MLabel1.Text;

    ListViewItem item;

    item = lvlFactura.Items.Add(cantidadTextBox.Text);
    item.SubItems.Add(descripcionLabel5.Text);

```

```

        item.SubItems.Add(precio_unitarioLabel1.Text);
        item.SubItems.Add(lblImporte.Text);

        lblSubtotal1.Text = lblsun.Text;
        label6.Text = lbliva.Text;
        label10.Text = lbltol.Text;

    }

    private void radioButton4_CheckedChanged(object sender, EventArgs e)
    {
        if (radioButton4.Checked == true) { cantidadTextBox.Enabled = true; }
    else { cantidadTextBox.Enabled = false; }
    }

    private void button3_Click(object sender, EventArgs e)
    {
        if (id_FaturaTextBox.Text != "")
        {
            fac.cancelar(Convert.ToInt32(id_FaturaTextBox.Text));
        }
        this.Close();
    }

    private void bindingNavigator1_RefreshItems(object sender, EventArgs e)
    {

    }

    private void id_clienteLabel_Click(object sender, EventArgs e)
    {

    }

    private void numericUpDown1_ValueChanged(object sender, EventArgs e)
    {
        if (radioButton1.Checked == true)
        {
            cantidadTextBox.Text = Convert.ToString(6 * numericUpDown1.Value);
        }

    }

    private void radioButton1_CheckedChanged(object sender, EventArgs e)
    {
        if (radioButton1.Checked == true)
        {
            numericUpDown1.Visible = true;
        }
        else
            numericUpDown1.Visible = false;
    }

    private void radioButton2_CheckedChanged(object sender, EventArgs e)
    {
        if (radioButton2.Checked == true)
        {

```

```

        numericUpDown1.Visible = true;
    }
    else
        numericUpDown1.Visible = false;
}

private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton3.Checked == true)
    {
        numericUpDown1.Visible = true;
    }
    else
        numericUpDown1.Visible = false;
}

private void button2_Click(object sender, EventArgs e)
{
    factura factu = new factura();

    factu.Id_Fatura = Convert.ToInt32(id_FaturaTextBox.Text);
    factu.Id_Empleado = Convert.ToInt32(id_EmpleadoLabel1.Text);
    factu.Id_cliente = Convert.ToInt32(id_clienteLabel1.Text);
    factu.Fecha = Convert.ToString(fech.Value.Year + "-" + fech.Value.Month
+ "-" + fech.Value.Day);
    factu.Cantidad = Convert.ToInt32(canti.Text);
    factu.Subtotal = Convert.ToDouble(lblsun.Text);
    factu.Total = Convert.ToDouble(lbltol.Text);
    int n = factu.alta_fact();

    tabControl11.SelectTab(tabPage2);

    //this.Close();
}

private void button4_Click(object sender, EventArgs e)
{
    printForm1.Print();
}

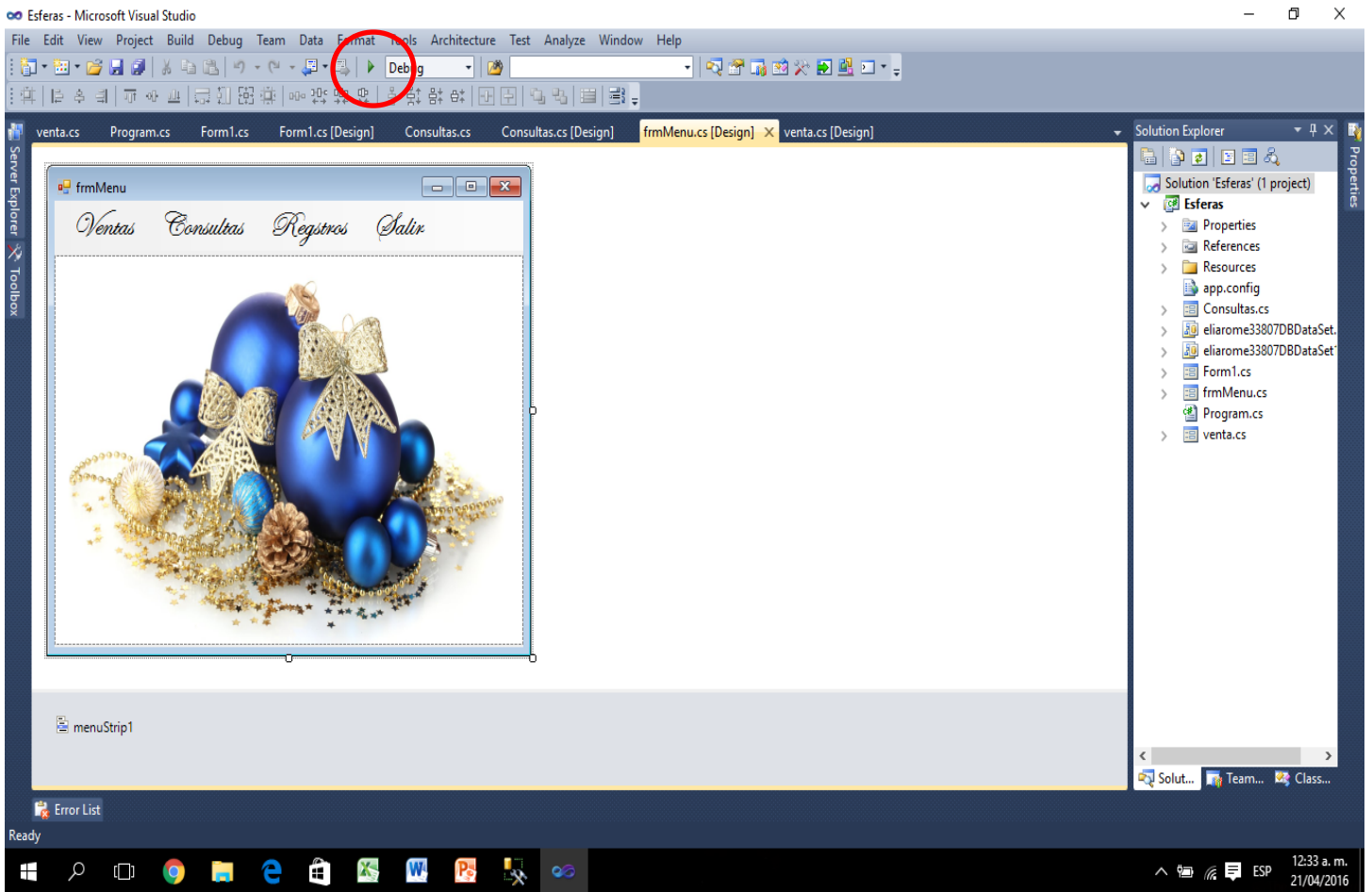
private void tabControl11_SelectedIndexChanged(object sender, EventArgs e)
{
}

}
}

```


VENTANAS

1° Ejecutar la aplicación



2° Ya ejecutada la aplicación nos aparecerá una ventana de menú para el cliente en la cual podremos encontrar Ventas en las cuales el cliente podrá comprar las esferas que el desee



Venta Factura

Id Fatura: jueves . 21 de abril de 2016

Cliente

Nombre: label1
Apellidos P: label1
Apellidos M: label1
label4

Empleado

Nombre: label1
Apellidos P: label1
Apellidos M: label1
label4

	Id_esferas	descripcion	precio_unitario	Precio_sin_box
*				

Id esferas: label4
precio unitario: label7
Precio en caja: label7
descripcion: label4

Caja

☐ 6
☐ 12
☐ 24
☐ Unitario

cantidad: Agregar

Cantidad 0
Importe: 0
Subtotal 0.0
IVA 0.0
Total 0.0

Pagar Cancelar

En donde el cliente podrá solicitar una factura de lo comprado

Venta

Factura

Cristal Helado

No.Factura: _____

Fecha de expedicion: _____

Cliente: _____

CANTIDAD	DESCRIPCION	PRECIO UNITARIO	IMPORTE

Subtotal: _____

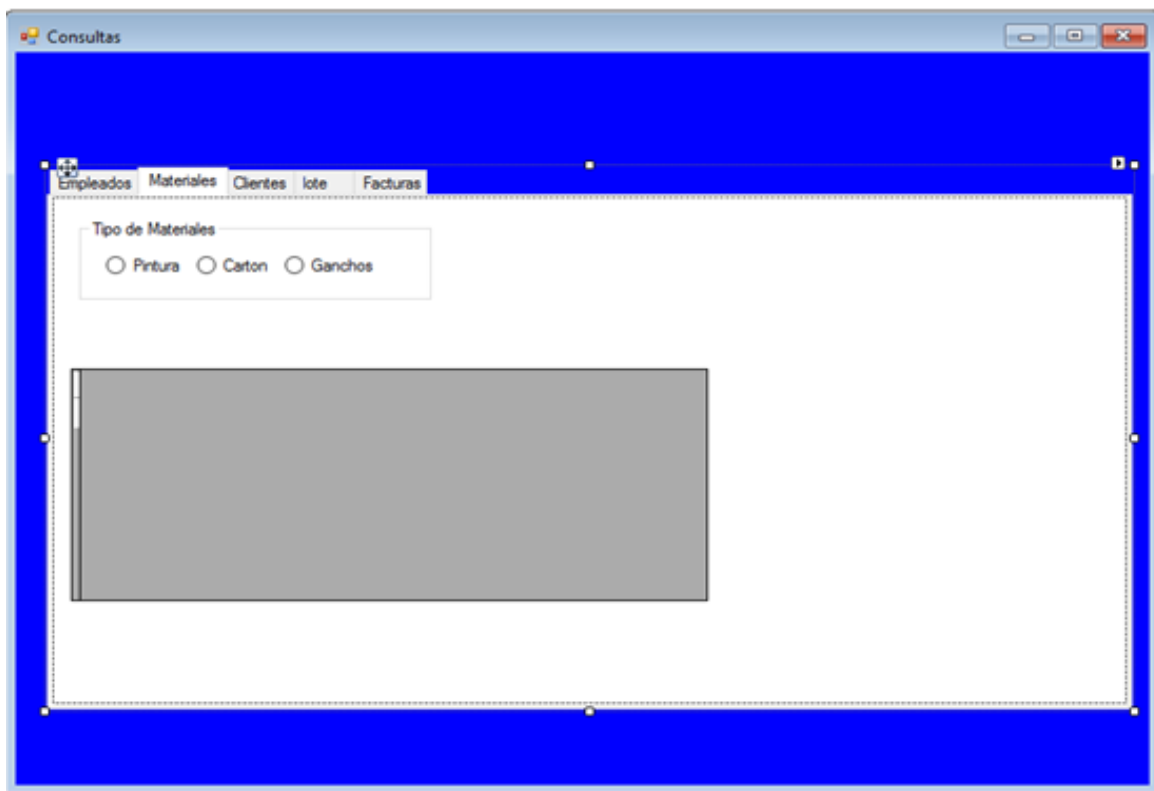
IVA(16%): _____

Total: _____

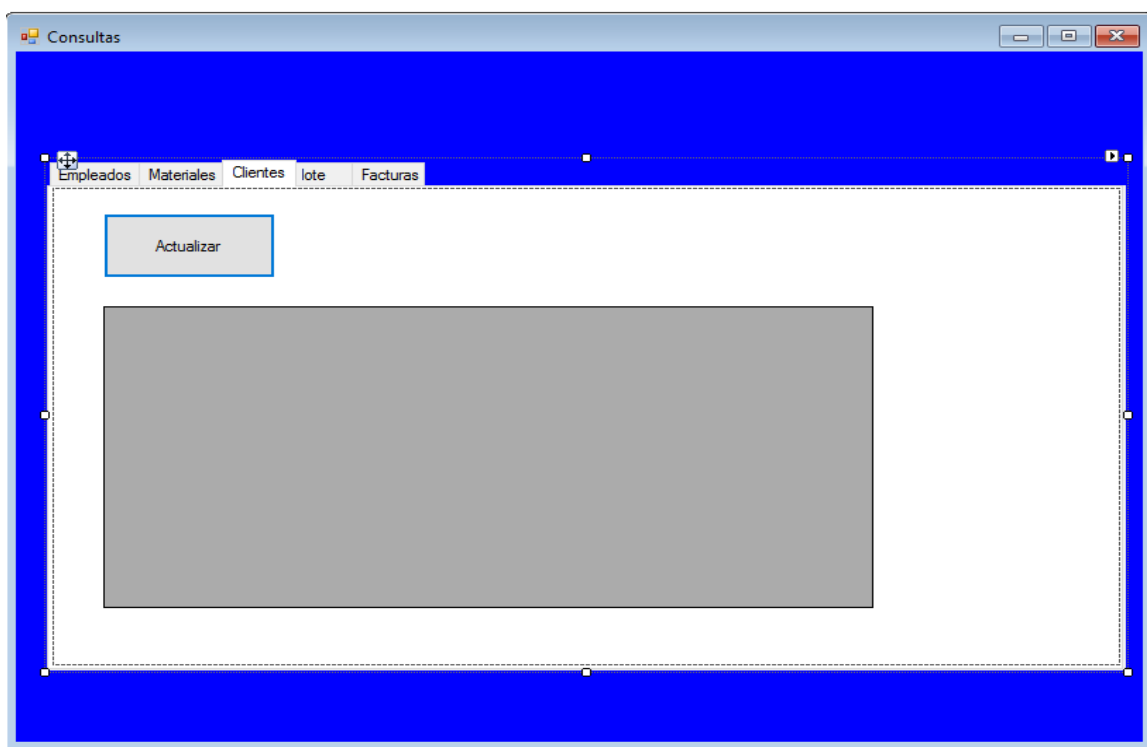
Imprimir

4°En las demás partes del sistema podremos accesar alos datos del empeado





En esta pantalla podremos encontrar a los clientes dados de alta por el sistema



Consultas

Empleados Materiales Clientes lote Facturas

Actualizar

	Id_esferas	descripcion	Id_lote
*			

En esta parte se puede consultar las facturas ya realizadas por el empleado al cliente

Consultas

Empleados Materiales Clientes lote Facturas

0 of {0}

	Id_Fatura	Id_Empleado	Id_cliente	Fecha	Cantidad	Subtotal	Total
*							

Id Fatura: label1 Buscar

Cliente

Empleado

Descripcion listBox1

Cantidad listBox2

Subtotal listBox3

5° Registro podremos dar de alta a los clientes que se les venderá los productos así como también a los empleados que realizan la venta

The screenshot shows a window titled "Venta de Esferas" with a tabbed interface. The "Cliente" tab is selected. Inside the tab, there is a dashed rectangular area containing a form with the following fields: "Nombre:" with a text input, "Apellido paterno:" with a text input, "Apellido materno:" with a text input, and "Telefono:" with a text input. Below these fields is a button labeled "Agregar".

The screenshot shows the same "Venta de Esferas" window, but with the "Nuevo Lote" tab selected. The dashed rectangular area contains a form with the following fields: "Id_lote" with a text input, "Id_Pintura" with a dropdown menu, "Id_carton" with a dropdown menu, and "Id_Gancho" with a dropdown menu. Below these are three dollar signs (\$). There is a blue hyperlink labeled "Consultar". Below that are "Cantidad" with a text input, "Tamaño (cm)" with a text input, and "Color" with a text input. At the bottom are "Inversión" and "Descripción" with text inputs, and an "Agregar" button.

Venta de Esferas

Cliente Nuevo Lote Materiales Empleado

Pintura

Color Precio por litro

Carton

Tipo de caja ☐ 6 ☐ 12 ☐ 24 Precio por kilo

Gancho

Cantidad Precio por unidad (c)

Venta de Esferas

Cliente Nuevo Lote Materiales Empleado

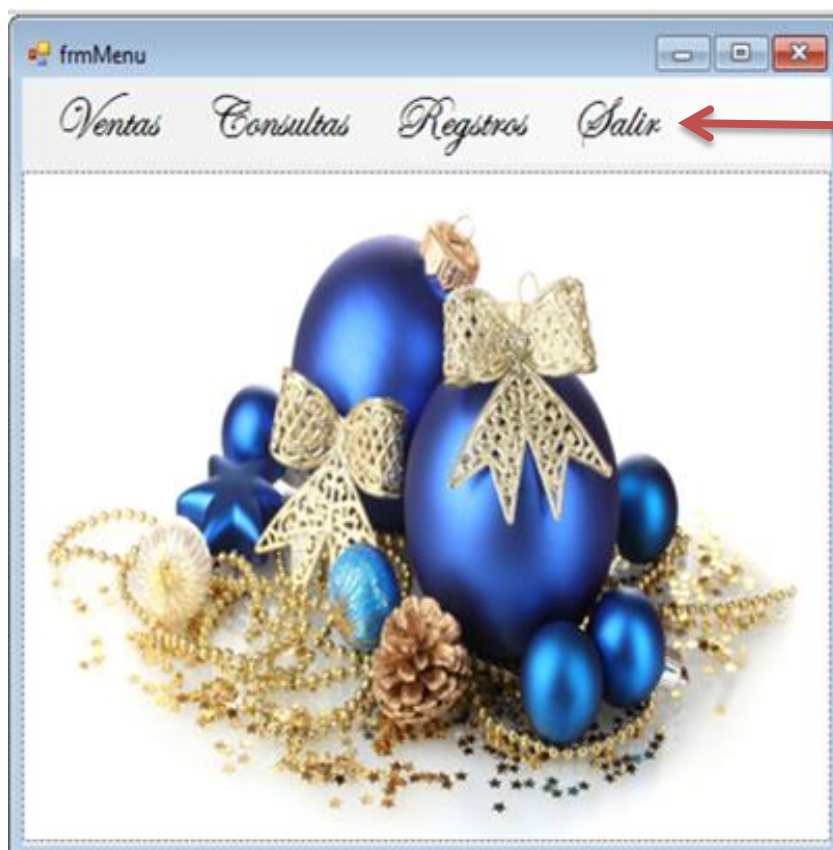
Nombre:

Apellido paterno:

Apellido materno:

Telefono:

6° Al final si el usuario ya no requiere hacer otra compra puede seleccionar la opción de salir

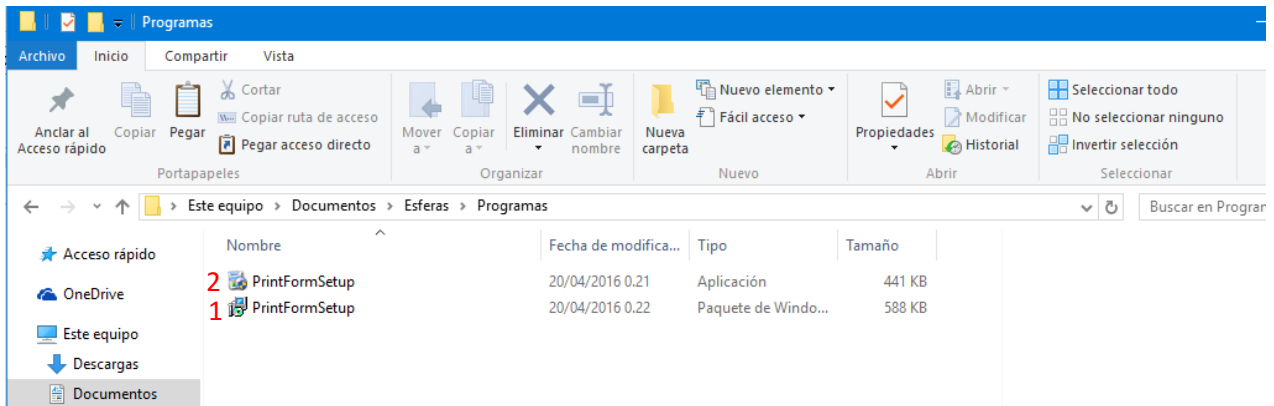


REQUISITOS DE LA APLICACIÓN

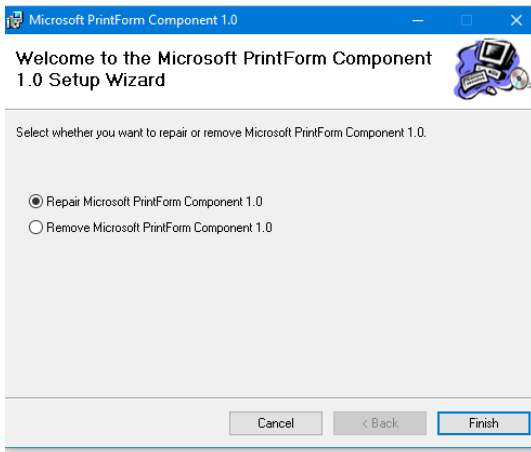
Para el uso de aplicación deberá contar con ciertos programas instalados

- Contar Mínimo con Mysql Conectot 6.0.0
- Tener instalado PrintForm

Instalación de PrintForm

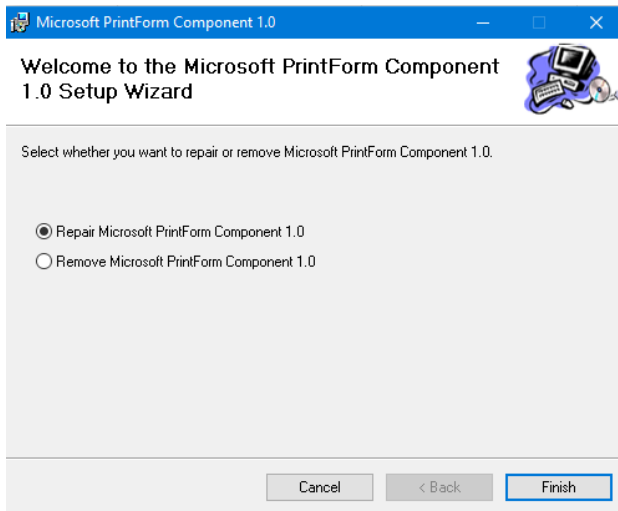


Instalamos Primero el paquete de Windows



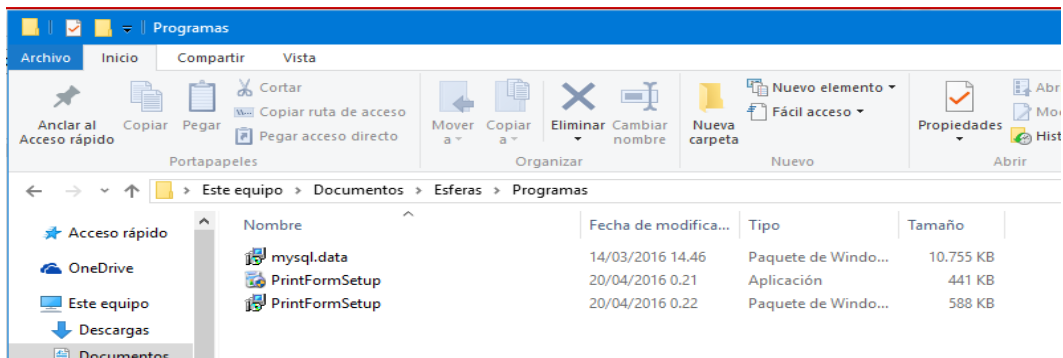
Le daremos click en Finish después de solicitar que aceptas licencia escoges la opción si y daremos next después de instalar le daremos cerrar.

Una vez instalado abriremos la segunda aplicación.

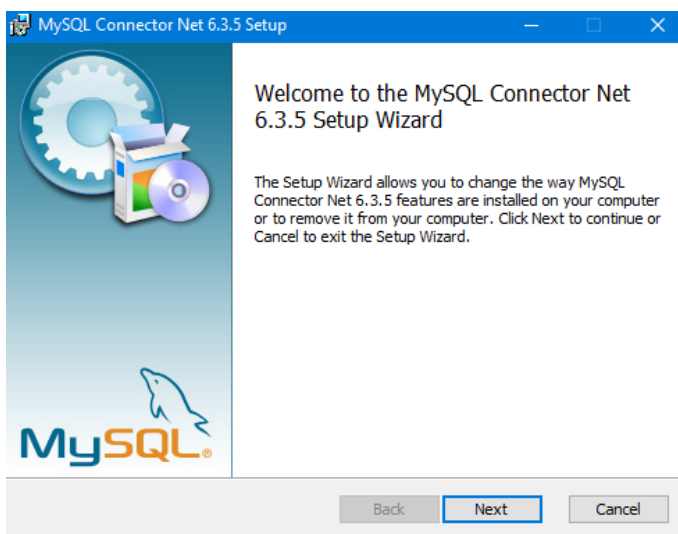


Dejaremos la opción default y daremos a finish una vez instalado podemos abrir la aplicación en el área de ventas podremos imprimir la factura gracias a esta librería.

Instalar Mysql Conector



Abrimos la aplicación de mysql.data



Seguimos los pasos de instalación para que la aplicación esferas pueda conectar con la base de datos la cual se encuentra montada en el servidor.

Ejecutar la aplicación Esferas

Para ejecutar la aplicación directamente basta con dar click en el acceso directo fuera de la carpeta esferas

