UNIVERSIDAD TECNOLÓGICA DE PUEBLA



MATERIA: DESARROLLO DE APLICACIONES II.

PROFESOR: ROJAS LÓPEZ ARTURO.

TRABAJO: DOCUMENTACIÓN (APLICACIÓN ESFERAS)

ALUMNOS: SAAVEDRA AGUILAR JHOVANNI.

CABALLERO MORALES OSVALDO.

GRUPO: 4-A MATUTINO

19/4/2016

ÍNDICE

PROBLEMÁTICA ………………………………………………………………….. (3)

BASE DE DATOS …………………………………………………………………… (4)

CÓDIGO BASE DE DATOS ……………………………………………………… (4)

CÓDIGO CLASE DATOS …………………………………………………………..(12)

CÓDIGO FORMULARIO 1 ……………………………………………………….(36)

CÓDIGO FORMULARIO VENTAS …………………………………………… (37)

CÓDIGO FORMULARIO CONSULTAS …………………………………….. (41)

CÓDIGO FORMULARIO COMPRA INSUMOS …………………………. (44)

PROBLEMÁTICA

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.

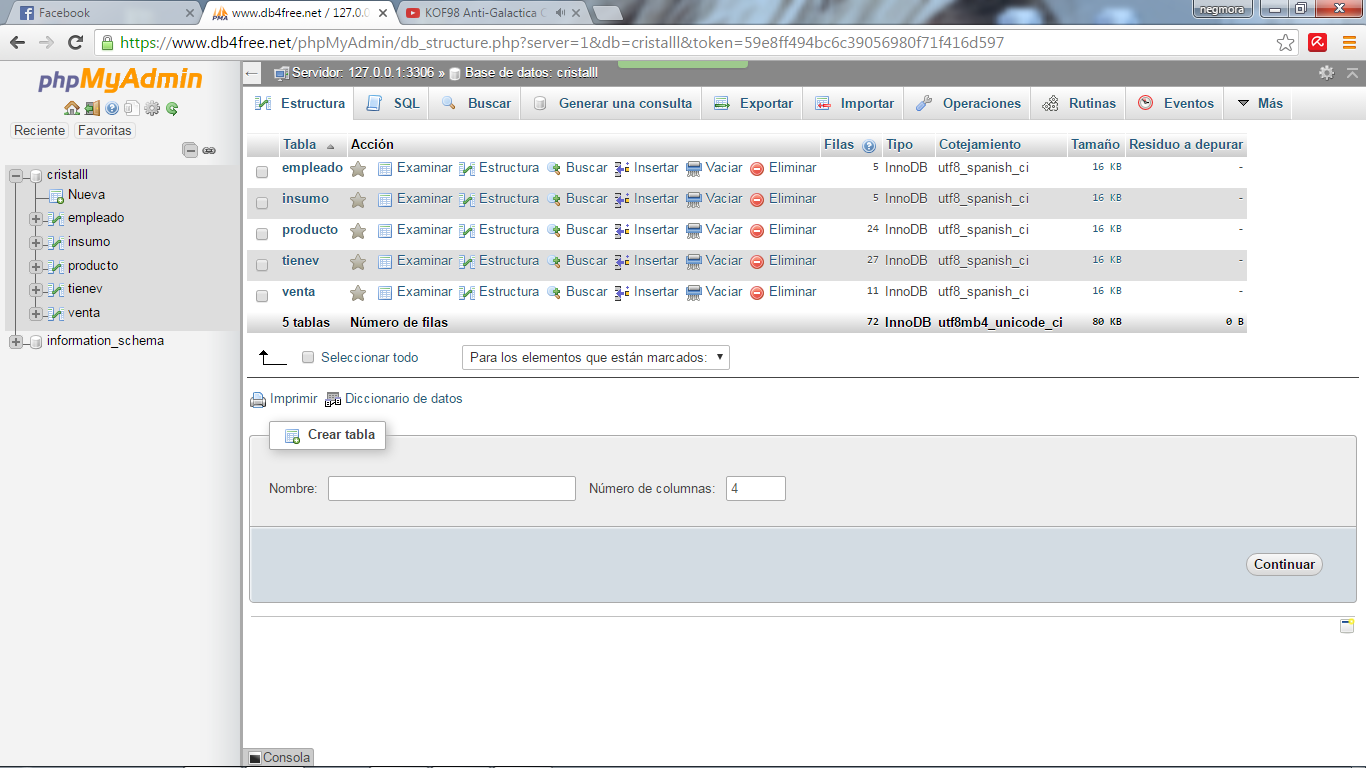
Crear un sistema que permita al Sr. Noel realizar la venta de su producto al público de una forma rápida considerando la compra de los insumos que permiten calcular los precios de la esfera. El software elaborará una factura al cliente que lo requiera, agregando el iva respectivo al precio de venta o 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.)

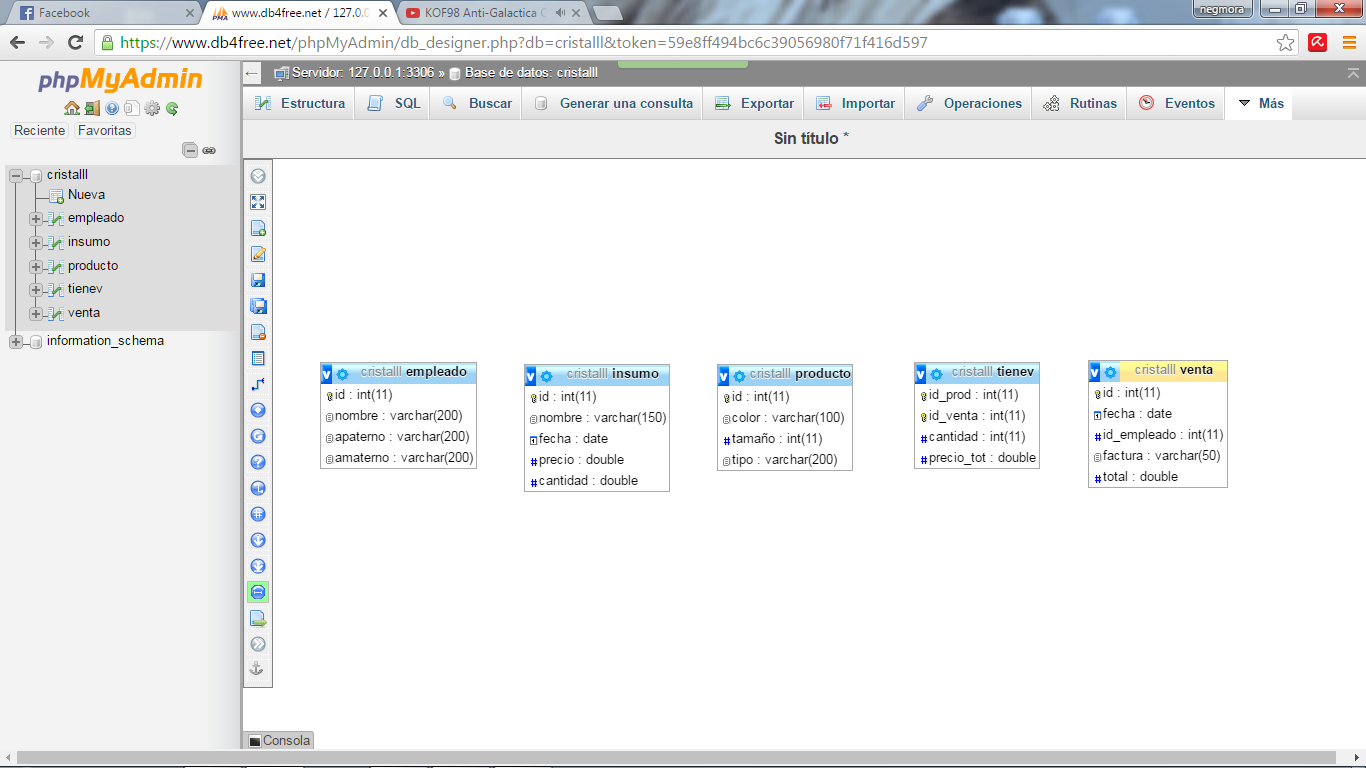
Utilizar una base de datos para gestionar la información de la empresa, conectar la aplicación a la base de datos y ofrecer las consultas de ventas indicadas además de la funcionalidad de la venta misma.

BASE DE DATOS (cristal)

Se ha decidido crear la base de datos “cristal” en el servidor db4free para la conexión remota de la aplicación. La base contiene las siguientes tablas:



Cada una de las tablas contiene los siguientes campos y atributos:



Código de la base de datos “cristal”.

-- phpMyAdmin SQL Dump

-- version 4.6.0

-- http://www.phpmyadmin.net

--

-- Servidor: 127.0.0.1:3306

-- Tiempo de generación: 19-04-2016 a las 13:57:35

-- Versión del servidor: 5.7.12

-- Versión de PHP: 5.5.9-1ubuntu4.14

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

--

-- Base de datos: `cristalll`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `empleado`

--

CREATE TABLE `empleado` (

`id` int(11) NOT NULL,

`nombre` varchar(200) COLLATE utf8\_spanish\_ci NOT NULL,

`apaterno` varchar(200) COLLATE utf8\_spanish\_ci NOT NULL,

`amaterno` varchar(200) COLLATE utf8\_spanish\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

--

-- Volcado de datos para la tabla `empleado`

--

INSERT INTO `empleado` (`id`, `nombre`, `apaterno`, `amaterno`) VALUES

(1, 'Noel', 'Hernández', 'Pérez'),

(2, 'Jhovanni ', 'Saavedra', 'Quintero'),

(3, 'Irving', 'Loranca', 'DeCarmona'),

(4, 'Christopher', 'Aguilar', 'Contreras'),

(5, 'Osvaldo', 'Caballero', 'Morales');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `insumo`

--

CREATE TABLE `insumo` (

`id` int(11) NOT NULL,

`nombre` varchar(150) COLLATE utf8\_spanish\_ci NOT NULL,

`fecha` date NOT NULL,

`precio` double NOT NULL,

`cantidad` double NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

--

-- Volcado de datos para la tabla `insumo`

--

INSERT INTO `insumo` (`id`, `nombre`, `fecha`, `precio`, `cantidad`) VALUES

(32, 'Dorada', '2016-04-14', 100, 1),

(33, 'Plateada', '2016-04-14', 150, 1),

(34, 'Roja', '2016-04-14', 200, 1),

(35, 'Gancho', '2016-04-14', 300, 1),

(36, 'Cartón', '2016-04-14', 80, 1);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `producto`

--

CREATE TABLE `producto` (

`id` int(11) NOT NULL,

`color` varchar(100) COLLATE utf8\_spanish\_ci NOT NULL,

`tamaño` int(11) NOT NULL,

`tipo` varchar(200) COLLATE utf8\_spanish\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

-- Volcado de datos para la tabla `producto`

--

INSERT INTO `producto` (`id`, `color`, `tamaño`, `tipo`) VALUES

(1, 'Dorada', 3, 'Esfera 3cm Dorada'),

(2, 'Dorada', 5, 'Esfera 5cm Dorada'),

(3, 'Dorada', 3, 'Caja 6 piezas 3cm Dorada'),

(4, 'Dorada', 5, 'Caja 6 piezas 5cm Dorada'),

(5, 'Dorada', 3, 'Caja 12 piezas 3cm Dorada'),

(6, 'Dorada', 5, 'Caja 12 piezas 5cm Dorada'),

(7, 'Dorada', 3, 'Caja 24 piezas 3cm Dorada'),

(8, 'Dorada', 5, 'Caja 24 piezas 5cm Dorada'),

(9, 'Plateada', 3, 'Esfera 3cm Plateada'),

(10, 'Plateada', 5, 'Esfera 5cm Plateada'),

(11, 'Plateada', 3, 'Caja 6 piezas 3cm Plateada'),

(12, 'Plateada', 5, 'Caja 6 piezas 5cm Plateada'),

(13, 'Plateada', 3, 'Caja 12 piezas 3cm Plateada'),

(14, 'Plateada', 5, 'Caja 12 piezas 5cm Plateada'),

(15, 'Plateada', 3, 'Caja 24 piezas 3cm Plateada'),

(16, 'Plateada', 5, 'Caja 24 piezas 5cm Plateada'),

(17, 'Roja', 3, 'Esfera 3cm Roja'),

(18, 'Roja', 5, 'Esfera 5cm Roja'),

(19, 'Roja', 3, 'Caja 6 piezas 3cm Roja'),

(20, 'Roja', 5, 'Caja 6 piezas 5cm Roja'),

(21, 'Roja', 3, 'Caja 12 piezas 3cm Roja'),

(22, 'Roja', 5, 'Caja 12 piezas 5cm Roja'),

(23, 'Roja', 3, 'Caja 24 piezas 3cm Roja'),

(24, 'Roja', 5, 'Caja 24 piezas 5cm Roja');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `tienev`

--

CREATE TABLE `tienev` (

`id\_prod` int(11) NOT NULL,

`id\_venta` int(11) NOT NULL,

`cantidad` int(11) NOT NULL,

`precio\_tot` double NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

--

-- Volcado de datos para la tabla `tienev`

--

INSERT INTO `tienev` (`id\_prod`, `id\_venta`, `cantidad`, `precio\_tot`) VALUES

(1, 12, 1, 0),

(1, 15, 1, 0),

(1, 20, 1, 0.99047619047619),

(3, 21, 1, 4.45714285714286),

(4, 14, 20, 0),

(7, 8, 1, 0),

(7, 19, 3, 0),

(8, 13, 1, 0),

(9, 12, 1, 0),

(9, 15, 1, 0),

(9, 20, 1, 1.20714285714286),

(10, 17, 2, 0),

(11, 18, 3, 0),

(11, 21, 1, 4.67380952380952),

(12, 18, 2, 0),

(13, 14, 10, 0),

(14, 16, 1, 0),

(17, 12, 1, 0),

(17, 15, 1, 0),

(17, 20, 1, 1.42380952380952),

(19, 21, 1, 4.89047619047619),

(20, 13, 1, 0),

(21, 16, 1, 0),

(23, 18, 3, 0),

(24, 13, 1, 0),

(24, 14, 24, 0),

(24, 19, 2, 0);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `venta`

--

CREATE TABLE `venta` (

`id` int(11) NOT NULL,

`fecha` date NOT NULL,

`id\_empleado` int(11) NOT NULL,

`factura` varchar(50) COLLATE utf8\_spanish\_ci NOT NULL,

`total` double NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

--

-- Volcado de datos para la tabla `venta`

--

INSERT INTO `venta` (`id`, `fecha`, `id\_empleado`, `factura`, `total`) VALUES

(11, '2016-04-14', 3, 'si', 100),

(12, '2016-04-15', 4, 'si', 300),

(13, '2016-04-15', 3, 'si', 200),

(14, '2016-04-17', 3, 'si', 300),

(15, '2016-04-17', 5, 'si', 250),

(16, '2016-04-18', 3, 'no', 40),

(17, '2016-04-18', 2, 'si', 360),

(18, '2016-04-18', 4, 'no', 470),

(19, '2016-04-18', 5, 'no', 457),

(20, '2016-04-18', 2, 'si', 3.62142857142857),

(21, '2016-04-18', 5, 'si', 14.0214285714286);

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `empleado`

--

ALTER TABLE `empleado`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `insumo`

ALTER TABLE `insumo`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `producto`

--

ALTER TABLE `producto`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `tienev`

--

ALTER TABLE `tienev`

ADD PRIMARY KEY (`id\_prod`,`id\_venta`);

--

-- Indices de la tabla `venta`

--

ALTER TABLE `venta`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT de las tablas volcadas

--

--

-- AUTO\_INCREMENT de la tabla `insumo`

--

ALTER TABLE `insumo`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=37;

--

-- AUTO\_INCREMENT de la tabla `venta`

--

ALTER TABLE `venta`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=22;

Código de la clase “Datos”.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data;

using MySql.Data.MySqlClient;

namespace Cristalhelado

{

static class Datos

{

private static MySqlConnection conectar;

private static MySqlDataReader dr;

public static double totalventa;

static Datos()

{

//conectar = new MySqlConnection("Server=localhost;Database=c+ristal23;Uid=root"); //Conexión Local

conectar = new MySqlConnection("Server=db4free.net; Database=cristalll; userid=osvaldoutp; pwd=proyectoarturo; port=3306;");

try

{

conectar.Open();

MessageBox.Show("Conectado a la base de datos remota");

}

catch (MySqlException ex)

{

MessageBox.Show("No se pudo conectar a la base de datos " + ex);

}

}

public static void Insertar(string cadena)

{

MySqlCommand comando = new MySqlCommand(cadena, conectar);

comando.ExecuteNonQuery();

}

public static void Update(string consulta)

{

MySqlCommand comando = new MySqlCommand(consulta, conectar);

comando.ExecuteNonQuery();

}

public static void BuscarDato(string busqueda, DataGridView dg)

{

DataTable tabla = new DataTable();

MySqlDataAdapter da = new MySqlDataAdapter(busqueda, conectar);

da.Fill(tabla);

dg.DataSource = tabla;

}

public static void ObtenerTabla(string consulta, DataGridView dg)

{

try

{

DataTable tabla = new DataTable();

MySqlDataAdapter da = new MySqlDataAdapter(consulta, conectar);

da.Fill(tabla);

dg.DataSource = tabla;

}

catch (Exception e)

{

MessageBox.Show(e.ToString());

}

}

public static MySqlDataAdapter construye\_adapter(string cadena)

{

MySqlDataAdapter adapt = new MySqlDataAdapter(cadena, conectar);

return (adapt);

}

public static 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 oEx)

{

MessageBox.Show(oEx.Message);

return (null);

}

}

public static void cargaCombo(ComboBox cmb, string cadena)

{

cmb.Items.Clear();

MySqlDataReader lector;

lector = construye\_reader(cadena);

while (lector.Read())

{

cmb.Items.Add(lector.GetString(0).ToString());

}

dr.Close();

}

public static double TotalPagar(int cantidad, string color, int id)

{

string com;

string com1;

string com2;

double total = 0;

double iva = 0;

double totalidad = 0;

double pregan;

double presfe;

double precar;

com = " Select \* From insumo where nombre = '" + color + "'";

com1 = " Select \* From insumo where nombre = 'gancho' ";

com2 = " Select \* From insumo where nombre = 'carton' ";

if (id == 1)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 300) + (pregan / 700)) \*cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 2)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 255) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 3)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 4)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 5)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 6)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 7)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 8)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 9)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 300) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 10)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 255) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 11)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 12)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 13)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 14)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 15)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 16)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 17)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 300) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 18)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 255) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 19)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 20)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 21)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 22)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 23)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

else

if (id == 24)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return totalidad;

}

return totalidad;

}

public static double IVA(int cantidad, string color, int id)

{

string com;

string com1;

string com2;

double total = 0;

double iva = 0;

double totalidad = 0;

double pregan;

double presfe;

double precar;

com = " Select \* From insumo where nombre = '" + color + "'";

com1 = " Select \* From insumo where nombre = 'gancho' ";

com2 = " Select \* From insumo where nombre = 'carton' ";

if (id == 1)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 300) + (pregan / 700)) \*cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 2)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 255) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 3)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 4)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 5)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 6)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 7)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 8)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 9)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 300) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 10)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 255) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 11)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 12)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 13)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 14)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 15)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 16)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 17)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 300) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 18)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 255) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 19)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 20)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 21)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 22)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 23)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

else

if (id == 24)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return iva;

}

return iva;

}

public static double PRECIONORMAL(int cantidad, string color, int id)

{

string com;

string com1;

string com2;

double total = 0;

double iva = 0;

double totalidad = 0;

double pregan;

double presfe;

double precar;

com = " Select \* From insumo where nombre = '" + color + "'";

com1 = " Select \* From insumo where nombre = 'gancho' ";

com2 = " Select \* From insumo where nombre = 'carton' ";

if (id == 1)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 300) + (pregan / 700)) \*cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 2)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 255) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 3)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 4)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 5)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 6)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 7)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 8)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 9)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 300) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 10)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 255) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 11)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 12)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 13)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 14)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 15)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 16)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 17)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 300) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 18)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

total = ((presfe / 255) + (pregan / 700)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 19)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 20)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 30)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 21)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 22)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 20)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 23)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 300) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

else

if (id == 24)

{

pregan = Convert.ToDouble(extrae(com1));

presfe = Convert.ToDouble(extrae(com));

precar = Convert.ToDouble(extrae(com2));

total = ((presfe / 255) + (pregan / 700) + (precar / 15)) \* cantidad;

iva = total \* .30;

totalidad = total + iva;

return total;

}

return total;

}

public static string extrae(string cadena)

{

MySqlDataAdapter da = construye\_adapter(cadena);

DataRow dt = extrae\_registro(da, "insumo");

string idv1 = "";

if (dt != null)

{

idv1 = dt["precio"].ToString();

}

return idv1;

}

public static MySqlDataReader construye\_reader(string cadena)

{

MySqlCommand cadena\_sql;

cadena\_sql = new MySqlCommand();

cadena\_sql.Connection = conectar;

cadena\_sql.CommandText = cadena;

cadena\_sql.CommandType = CommandType.Text;

try

{

dr = cadena\_sql.ExecuteReader();

return dr;

}

catch (Exception oEx)

{

MessageBox.Show(oEx.Message);

return null;

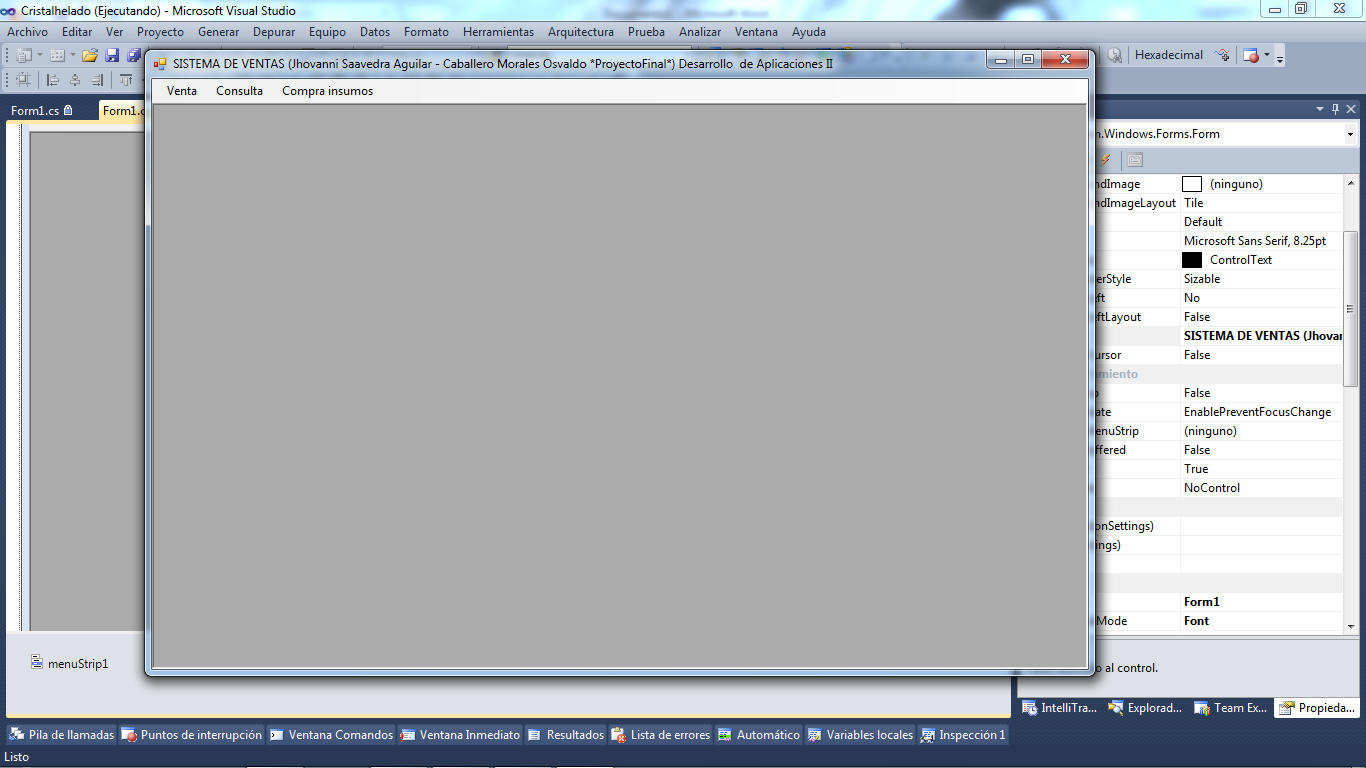
}

}

}//fin de clase

}//fin de namespace

FORMULARIO 1



Código Formulario 1

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 Cristalhelado

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

private void MostrarFormulario(Form Formulario)

{

Formulario.MdiParent = this;

Formulario.Show();

}

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

{

this.menuStrip1.Enabled = false;

frmventa formuhijo = new frmventa ();

formuhijo.FormClosing += new FormClosingEventHandler(form\_FormClosing);

formuhijo.Show();

}

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

{

this.menuStrip1.Enabled = false;

frmconsulta formuhijo = new frmconsulta();

formuhijo.FormClosing += new FormClosingEventHandler(form\_FormClosing);

formuhijo.Show();

}

private void form\_FormClosing(object sender, FormClosingEventArgs e)

{

this.menuStrip1.Enabled = true;

}

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

{

this.menuStrip1.Enabled = false;

frminsumos formuhijo = new frminsumos();

formuhijo.FormClosing += new FormClosingEventHandler(form\_FormClosing);

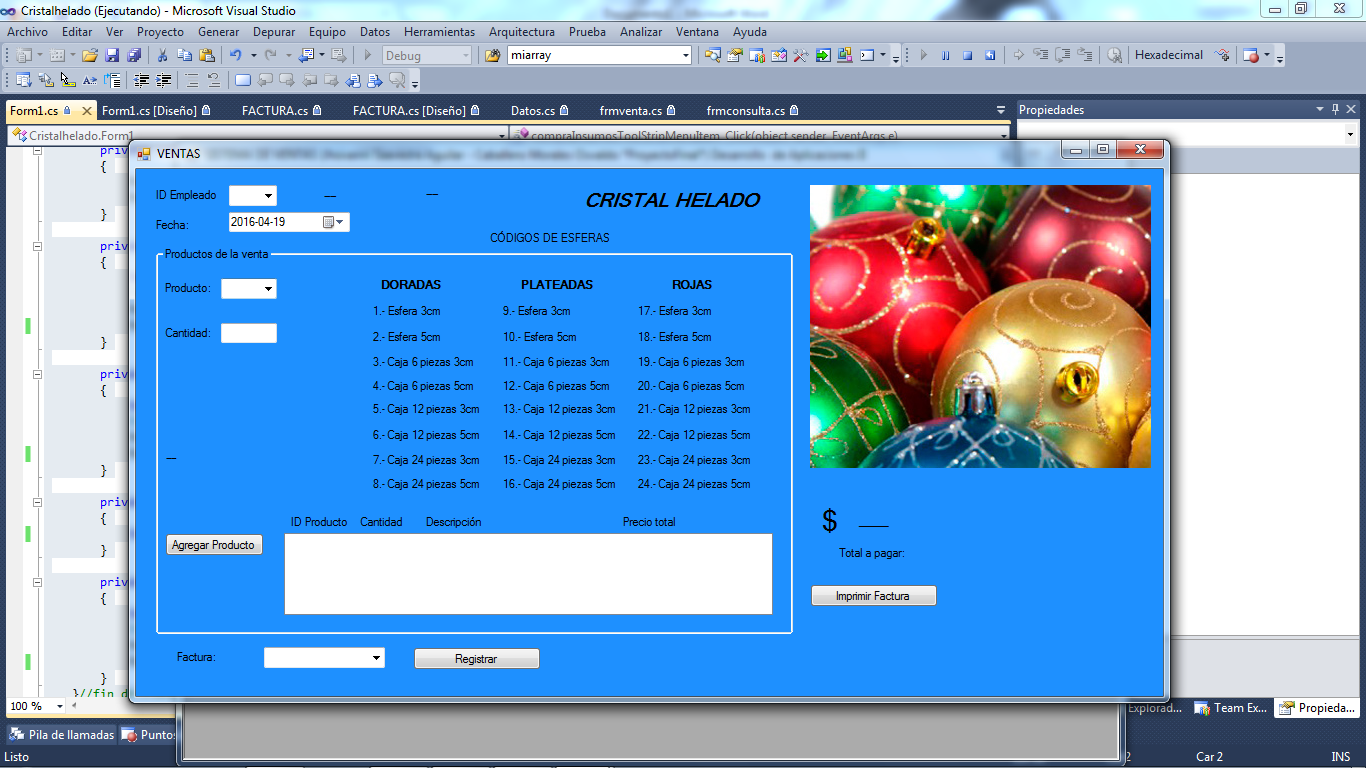
formuhijo.Show();

}

}//fin de clase

}

FORMULARIO VENTA



Código formulario venta.

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 System.Collections;

using MySql.Data.MySqlClient;

namespace Cristalhelado

{

public partial class frmventa : Form

{

Queue producto = new Queue();

Queue cantidad = new Queue();

Queue preciocola = new Queue();

private double acumuladorTotal = 0.0;

private double acumuladorNormal = 0.0;

private double acumuladorIva = 0.0;

int idv;

string idv1;

public frmventa()

{

InitializeComponent();

}

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

{

string cad = "Select \* From empleado";

Datos.cargaCombo(cmbempleado, cad);

string cad1 = "Select \* From producto";

Datos.cargaCombo(cmbproducto, cad1);

MySqlDataAdapter da = Datos.construye\_adapter("SELECT MAX(id) AS id FROM venta");

DataRow dt = Datos.extrae\_registro(da, "venta");

if (dt != null)

{

idv1= dt["id"].ToString();

}

idv = Convert.ToInt32(idv1) + 1;

MessageBox.Show("Número de venta "+ idv.ToString());

}

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

{

double total = Datos.TotalPagar(Convert.ToInt32(txtcantidad.Text), cmbcolor.Text, Convert.ToInt32(cmbproducto.Text));

acumuladorTotal += total;

double iba = Datos.IVA(Convert.ToInt32(txtcantidad.Text), cmbcolor.Text, Convert.ToInt32(cmbproducto.Text));

acumuladorIva += iba;

lblIVA.Text = iba.ToString();

double normal = Datos.PRECIONORMAL(Convert.ToInt32(txtcantidad.Text), cmbcolor.Text, Convert.ToInt32(cmbproducto.Text));

acumuladorNormal += normal;

lblPRECIONORMAL.Text = normal.ToString();

lblIVA.Text = acumuladorIva.ToString();

lblPRECIONORMAL.Text = acumuladorNormal.ToString();

lblMontoTotal.Text = acumuladorTotal.ToString();

try

{

preciocola.Enqueue(total);

producto.Enqueue(cmbproducto.Text);

cantidad.Enqueue(txtcantidad.Text);

lstproductos.Items.Add("\t" + cmbproducto.Text.ToString() + "\t" + txtcantidad.Text.ToString() + "\t" + lbventa.Text + "\t" + /\*cmbcolor.Text.ToString() +\*/ "\t" + total.ToString());

}

catch (Exception)

{

MessageBox.Show("Error intenta de nuevo");

}

}

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

{

Datos.totalventa = acumuladorTotal;

if (cmbempleado.SelectedIndex != -1 && cmbfactura.SelectedIndex != -1)

{

string cadena = "";

string cadena2 = "";

if (MessageBox.Show("¿Deseas agregar un nuevo registro?", "Cristal Helado", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==

DialogResult.Yes)

{

try

{

cadena = "INSERT INTO venta (fecha, id\_empleado, factura, total) values ('" + dtpfechav.Text +

"', '" + cmbempleado.Text + "', '" + cmbfactura.Text + "', '"+acumuladorTotal.ToString()+"')";

Datos.Insertar(cadena);

MessageBox.Show("Registro agregado");

}

catch (Exception)

{

MessageBox.Show("Error al insertar");

}

try

{

int tot = producto.Count;

for (int x = 0; x < tot; x++)

{

cadena2 = " Insert into tienev values ('" + producto.Dequeue().ToString() + "', '"+idv+"', '" + cantidad.Dequeue().ToString() + "', '"+preciocola.Dequeue().ToString()+"')";

Datos.Insertar(cadena2);

}

}

catch (Exception)

{

MessageBox.Show("Error al insertar productos");

}

}

}

else

{

MessageBox.Show("Debes llenar todos los campos");

}

}

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

{

if (cmbproducto.SelectedIndex >= 0 && cmbproducto.SelectedIndex <= 7)

{

cmbcolor.SelectedIndex = 0;

}

else

if (cmbproducto.SelectedIndex >= 8 && cmbproducto.SelectedIndex <= 15)

{

cmbcolor.SelectedIndex = 1;

}

else

if (cmbproducto.SelectedIndex >= 16 && cmbproducto.SelectedIndex <= 23)

{

cmbcolor.SelectedIndex = 2;

}

MySqlDataAdapter da = Datos.construye\_adapter("select \* from producto where id ='" + cmbproducto.Text + "'");

DataRow dt = Datos.extrae\_registro(da, "producto");

if (dt != null)

{

lbventa.Text = dt["tipo"].ToString();

}

}

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

{

FACTURA fact = new FACTURA();

fact.lblEmpleado.Text = cmbempleado.Text;

fact.lblFecha.Text = dtpfechav.Text;

fact.lblTOTALPAGAR.Text = lblMontoTotal.Text;

fact.lbNombreEmpleado.Text = lbNombre.Text;

fact.lblFacturaApellido.Text = lblApellidoEmpleado.Text;

fact.lblFACTURAIVA.Text = lblIVA.Text;

fact.lblTOTALFACTURA.Text = lblPRECIONORMAL.Text;

for (int i = 0; i <= lstproductos.Items.Count -1; i++)

{

fact.listBox1.Items.Insert(i, lstproductos.Items[i]);

}

fact.ShowDialog();

}

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

{

MySqlDataAdapter de = Datos.construye\_adapter("select \* from empleado where id ='" + cmbempleado.Text + "'");

DataRow dte = Datos.extrae\_registro(de, "empleado");

if (dte != null)

{

lbNombre.Text = dte["nombre"].ToString();

lblApellidoEmpleado.Text = dte["apaterno"].ToString();

}

}

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

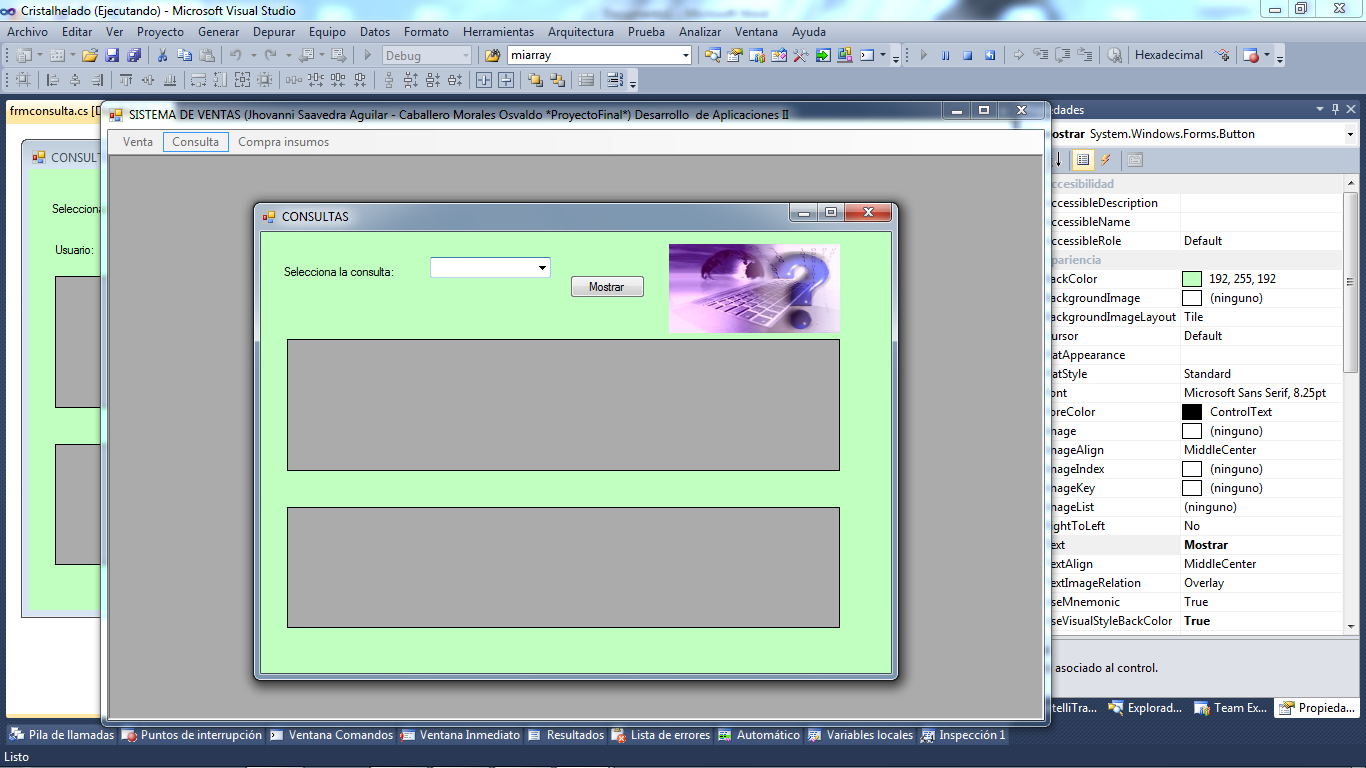
{

}

}//fin de clase

}

FORMULARIO CONSULTA



Código Formulario consulta.

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 Cristalhelado

{

public partial class frmconsulta : Form

{

int id1;

public frmconsulta()

{

InitializeComponent();

}

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

{

}

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

{

string cadena;

if (cmbconsulta.SelectedIndex == 0)

{

cadena = "Select v.id, v.fecha, v.id\_empleado, v.factura, v.total From venta v ";

Datos.ObtenerTabla(cadena, dataGridView1);

cmbusuario.Visible = false;

}

else

if (cmbconsulta.SelectedIndex == 1)

{

cadena = "Select \* from venta v inner join tienev t on v.id = t.id\_venta where v.factura = 'si' ";

Datos.ObtenerTabla(cadena, dataGridView1);

cmbusuario.Visible = false;

}

else

if (cmbconsulta.SelectedIndex == 2)

{

lbusuario.Visible = true;

cmbusuario.Visible = true;

cadena = "Select \* from venta v inner join tienev t on v.id = t.id\_venta where v.id\_empleado = '" + cmbusuario.Text + "' ";

Datos.ObtenerTabla(cadena, dataGridView1);

}

else

if (cmbconsulta.SelectedIndex == 3)

{

cadena = "Select \* from insumo";

Datos.ObtenerTabla(cadena, dataGridView1);

cmbusuario.Visible = false;

}

}//fin del boton mostrar

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

{

string cad = "Select \* From empleado";

Datos.cargaCombo(cmbusuario, cad);

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

string cadena3 = "";

try

{

id1 = Convert.ToInt32(dataGridView1[0, e.RowIndex].Value);

//label8.Text = id1.ToString();

}

catch (Exception)

{

MessageBox.Show("Selecciona un registro valido");

}

try

{

cadena3 = "Select tv.id\_venta, tv.id\_prod, p.tipo, tv.cantidad, tv.precio\_tot From tienev tv inner join producto p on tv.id\_prod = p.id where id\_venta = '" + id1 + "'";

Datos.ObtenerTabla(cadena3, dataGridView2);

}

catch (Exception)

{

MessageBox.Show("no se puede mostrar");

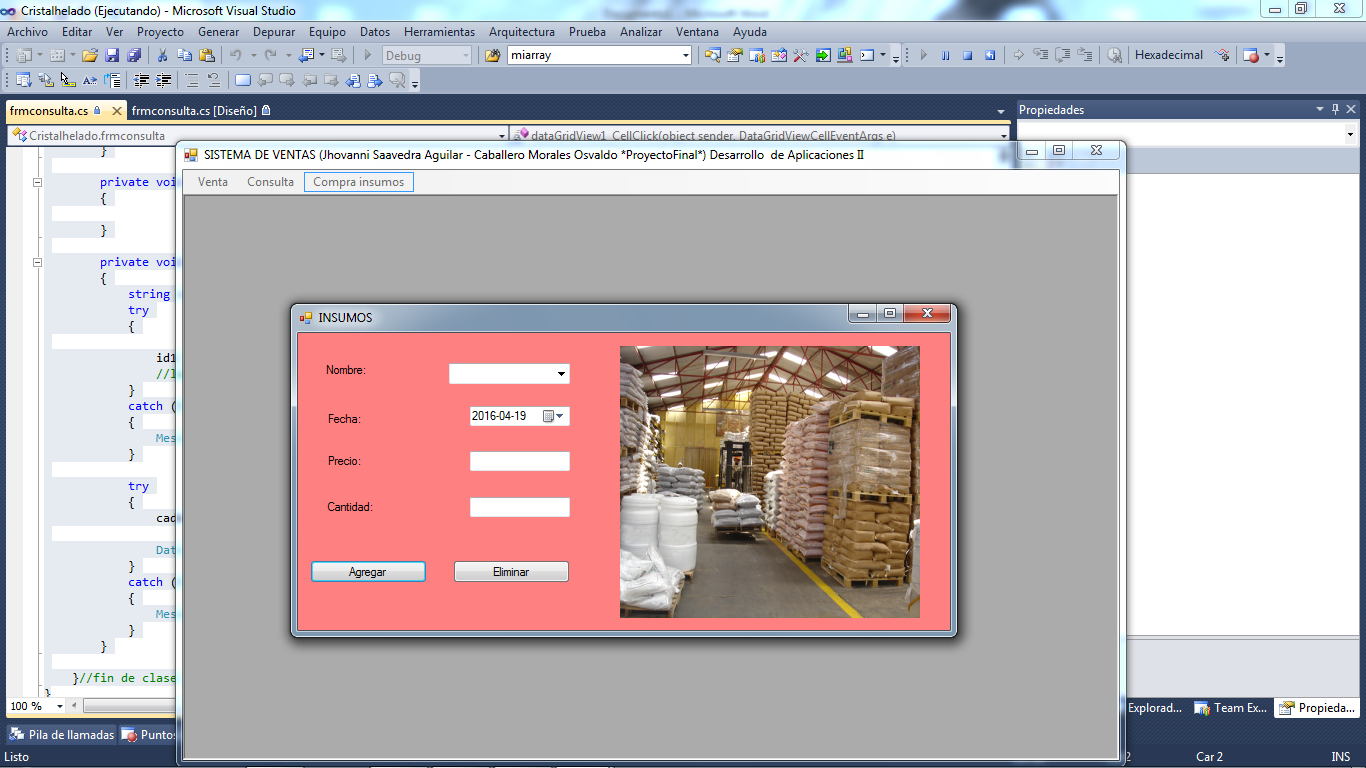
}

}

}//fin de clase

}

FORMULARIO COMPRA INSUMOS



Código formulario compra insumos.

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 Cristalhelado

{

public partial class frminsumos : Form

{

public frminsumos()

{

InitializeComponent();

}

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

{

if (cmbInsumos.Text != "" && txtprecio.Text != "" && txtcantidad.Text != "")

{

string cadena = "";

if (MessageBox.Show("Deseas agregar un nuevo registro", "Cristal Helado", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==

DialogResult.Yes)

{

try

{

cadena = "INSERT INTO insumo (nombre, fecha, precio, cantidad) values ('" + cmbInsumos.Text +

"', '" + dtpfechainsu.Text + "', '" + txtprecio.Text + "', '" + txtcantidad.Text + "')";

Datos.Insertar(cadena);

MessageBox.Show("Registro agregado");

cmbInsumos.ResetText();

txtprecio.Clear();

txtcantidad.Clear();

}

catch (Exception)

{

MessageBox.Show("Error al insertar");

}

}

}

else

{

MessageBox.Show("Debes llenar todos los campos");

}

}

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

{

}

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

{

if (cmbInsumos.Text != "")

{

string cadena = "";

if (MessageBox.Show("¿Estás seguro de querer eliminar este registro?", "Cristal Helado", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==

DialogResult.Yes)

{

try

{

if (cmbInsumos.SelectedIndex == 0)

{

cadena = "DELETE FROM insumo where nombre = 'Dorada'";

Datos.Insertar(cadena);

MessageBox.Show("Eliminación exitosa");

cmbInsumos.ResetText();

}

else

if (cmbInsumos.SelectedIndex == 1)

{

cadena = "DELETE FROM insumo where nombre = 'Plateada'";

Datos.Insertar(cadena);

MessageBox.Show("Eliminación exitosa");

cmbInsumos.ResetText();

}

else

if (cmbInsumos.SelectedIndex == 2)

{

cadena = "DELETE FROM insumo where nombre = 'Roja'";

Datos.Insertar(cadena);

MessageBox.Show("Eliminación exitosa");

cmbInsumos.ResetText();

}

else

if (cmbInsumos.SelectedIndex == 3)

{

cadena = "DELETE FROM insumo where nombre = 'Gancho'";

Datos.Insertar(cadena);

MessageBox.Show("Eliminación exitosa");

cmbInsumos.ResetText();

}

else

if (cmbInsumos.SelectedIndex == 4)

{

cadena = "DELETE FROM insumo where nombre = 'Cartón'";

Datos.Insertar(cadena);

MessageBox.Show("Eliminación exitosa");

cmbInsumos.ResetText();

}

}

catch (Exception)

{

MessageBox.Show("Error al eliminar el registro");

}

}

}

else

{

MessageBox.Show("Debes llenar todos los campos");

}

}

}//fin de clase

}