using MySql.Data.MySqlClient;

using sigal.Clases;

using System;

using System.Windows.Forms;

namespace sigal

{

public partial class Det\_Salidas : Form

{

int r = 0;

Int64 ID\_SAL;

#pragma warning disable CS0169 // El campo 'Det\_Salidas.PRECIO' nunca se usa

Double PRECIO, TOTAL = 0;

#pragma warning restore CS0169 // El campo 'Det\_Salidas.PRECIO' nunca se usa

Boolean resp;

conexion conn = new conexion();

conexion conn1 = new conexion();

conexion conn2 = new conexion();

conexion conn3 = new conexion();

conexion conn4 = new conexion();

conexion conn5 = new conexion();

MySqlDataReader R;

public Det\_Salidas()

{

InitializeComponent();

}

private void bt\_agregar\_Click(object sender, EventArgs e)

{

if (validar() == 0 && validar\_sal() == 0) { return; }

panel2.Enabled = false;

Grid\_Det\_Sal.Rows.Insert(r, txtid.Text, txtnombre.Text, txtcantidad.Text, txtprecio.Text, txtcosto.Text, txtmarca.Text, txtrequisicion.Text, txtcaducidad.Text);

r = r + 1;

TOTAL = double.Parse(txtimporte.Text) + double.Parse(txtcosto.Text);

txtimporte.Text = TOTAL.ToString();

limpiar();

cargar\_det();

txtid.Focus();

}

internal static void show()

{

throw new NotImplementedException();

}

private void txtid\_Leave(object sender, EventArgs e)

{

if (txtid.Text.ToString() != "")

{

R = conn.GetData("SELECT nombre\_corto, preciopromedio FROM catarticulos where id=" + txtid.Text + ";");

if (R.HasRows)

{

R.Read();

txtnombre.Text = R[0].ToString().ToUpper();

txtprecio.Text = R[1].ToString();

}

else

{

MessageBox.Show("No existe ese numero de articulo¡");

txtid.Focus();

}

R.Close();

}

}

private void Det\_Salidas\_Load(object sender, EventArgs e)

{

// mensajes.SetToolTip(txtBusqueda, "Registro a buscar...");

mensajes.SetToolTip(btnNuevo, "Clic para agregar un nuevo registro...");

// mensajes.SetToolTip(btnEditar, "Mostrar Detalle del Registro...");

mensajes.SetToolTip(btnSalir, "Regresar a Pantalla Anterior");

mensajes.SetToolTip(btnGrabar, "Guardar Registro...");

cargar();

conn.AbrirBD();

conn1.AbrirBD();

cbtipo.Focus();

}

void cargar\_det()

{

return;

#pragma warning disable CS0162 // Se detectó código inaccesible

Grid\_Det\_Sal.DataSource = conn.GetTable("select d.id,d.idarticulo as Id\_Articulo,a.nombre\_corto as Descripcion,cantidad as Cantidad," +

"d.precio as Precio,totalprecio as Total,estado as Estado,d.marbete as Marbete,e.nombre as Auditor " +

"from det\_inventarios d, empleados e, catarticulos a where d.idauditor=e.id and d.idarticulo=a.id and d.idinventario=" +

ID\_SAL + " order by d.id");

#pragma warning restore CS0162 // Se detectó código inaccesible

if ((conn.PropertyDataSet.Tables[0].Rows.Count != 0))

{

Grid\_Det\_Sal.Columns["id"].Visible = false;

Grid\_Det\_Sal.AutoResizeColumns();

}

}

void cargar()

{

txtfecha.Value = DateTime.Now;

conn2.AbrirBD();

cbentrego.DisplayMember = "nombre";

cbentrego.ValueMember = "id";

cbentrego.DataSource = conn2.GetTable("select id,nombre from empleados");

conn2.cerrarBd();

conn4.AbrirBD();

cbalmacen.DisplayMember = "nombre";

cbalmacen.ValueMember = "id";

if (configuracion.ID\_ALMACEN != 0)

{

cbalmacen.DataSource = conn4.GetTable("select id,nombre from catalmacen where id=" + configuracion.ID\_ALMACEN);

cbalmacen.Enabled = false;

}

else { cbalmacen.DataSource = conn4.GetTable("select id,nombre from catalmacen where id>0"); }

conn4.cerrarBd();

conn5.AbrirBD();

cbarea.DisplayMember = "nombre";

cbarea.ValueMember = "id";

cbarea.DataSource = conn5.GetTable("select id,nombre from catareas");

conn5.cerrarBd();

}

private void Det\_Salidas\_FormClosing(object sender, FormClosingEventArgs e)

{

conn.cerrarBd();

conn1.cerrarBd();

}

private void txtid\_TextChanged(object sender, EventArgs e)

{

if (txtid.Text == "")

{ txtnombre.Enabled = true; }

else { txtnombre.Enabled = false; }

}

private void txtcantidad\_KeyPress(object sender, KeyPressEventArgs e)

{

if (Char.IsDigit(e.KeyChar))

{ e.Handled = false; }

else if (Char.IsControl(e.KeyChar))

{ e.Handled = false; }

else if (Char.IsSeparator(e.KeyChar))

{ e.Handled = false; }

else if (e.KeyChar == 46)

{ e.Handled = false; }

else { e.Handled = true; }

}

void limpiar()

{

txtid.Text = "";

txtnombre.Text = "";

txtcantidad.Text = "";

txtprecio.Text = "";

txtcosto.Text = "";

txtmarca.Text = "";

txtrequisicion.Text = "";

}

private void txtnombre\_TextChanged(object sender, EventArgs e)

{

if (txtnombre.TextLength > 3 && txtid.Text == "")

{

Grid\_Articulos.Visible = true;

//grid1.DataSource = conn.GetTable("select id,nombre\_corto,nombre\_largo from catarticulos");

llena\_grid("select id as Id\_Producto,nombre\_corto as Nombre," +

" nombre\_largo as Descripcion from catarticulos where nombre\_corto like '%" + txtnombre.Text + "%' order by nombre\_corto limit 20");

}

}

private void llena\_grid(string sql)

{

Grid\_Articulos.DataSource = conn1.GetTable(sql);

if (Grid\_Articulos.RowCount < 2)

{

Grid\_Articulos.Visible = false;

return;

}

Grid\_Articulos.AutoResizeColumns();

}

private void Grid\_Articulos\_DoubleClick(object sender, EventArgs e)

{

txtid.Text = Grid\_Articulos.CurrentRow.Cells["Id\_Producto"].Value.ToString();

txtnombre.Text = Grid\_Articulos.CurrentRow.Cells["Nombre"].Value.ToString();

Grid\_Articulos.Visible = false;

txtcantidad.Focus();

txtnombre.Enabled = false;

}

private void txtid\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{ txtcantidad.Focus(); }

else if (e.KeyCode == Keys.Down) { txtnombre.Focus(); }

}

int validar()

{

if (txtid.Text != "" && txtcantidad.Text != "")

{ return 1; }

return 0;

}

int validar\_sal()

{

string P = suma\_costos();

if (txtfecha.Text != "" && cbtipo.Text != "" && cbalmacen.Text != "" && cbentrego.Text != "" && txtimporte.Text == P.ToString())

{ return 1; }

return 0;

}

private void bt\_grabar\_Click(object sender, EventArgs e)

{

try

{

if (validar\_sal() == 0) { return; }

conn.AbrirBD();

conn.inicio();

resp = conn.Executa("insert into salidas (tipo,fechasalida,importe,idrequisicion," +

"identrego,idarea,idalmacen,estatus) values('" + cbtipo.Text.ToUpper() +

"','" + txtfecha.Value.ToString("yyyy-MM-dd") + "'," + txtimporte.Text.ToUpper() +

",'" + txtrequisicion.Text.ToUpper() + "'," + cbentrego.SelectedValue + "," +

cbarea.SelectedValue + "," + cbalmacen.SelectedValue + ",'A')");

if (!resp) { conn.fallo(); return; }

//OBTENER ID INVENTARIO

R = conn.GetData("SELECT id FROM salidas where tipo='" + cbtipo.Text.ToUpper()

+ "' and fechasalida='" + txtfecha.Value.ToString("yyyy-MM-dd")

+ "' and importe=" + txtimporte.Text.ToUpper()

+ " and idarea=" + cbarea.SelectedValue + ";");

if (R.HasRows)

{

R.Read();

ID\_SAL = Int32.Parse(R[0].ToString());

label16.Visible = true;

label16.Text = "S A L I D A #" + ID\_SAL.ToString();

}

R.Close();

resp = grabar\_grid();

if (!resp) { conn.fallo(); return; }

resp = conn.Executa("insert into bitacora (fechasis,usuario,motivo,tabla,idtabla) values('"

+ DateTime.Now.ToString("yyyy-MM-dd hh:mm:ss") + "','" + configuracion.USER +

"','ALTA','SALIDAS'," + ID\_SAL + ")");

if (!resp) { conn.fallo(); return; }

conn.exito();

MessageBox.Show("Alta de Entrada al Almacen " + ID\_SAL + " Realizada con Exito¡");

this.Close();

}

catch (Exception ex)

{

MessageBox.Show("Fallo al guardar en la BD:" + ex.Message, Application.ProductName, MessageBoxButtons.YesNo, MessageBoxIcon.Question);

}

}

private void Grid\_Det\_Sal\_DoubleClick(object sender, EventArgs e)

{

if (Grid\_Det\_Sal.CurrentRow.Index != -1)

{

TOTAL = double.Parse(txtimporte.Text) - double.Parse(Grid\_Det\_Sal.CurrentRow.Cells[4].Value.ToString());

txtimporte.Text = TOTAL.ToString();

Grid\_Det\_Sal.Rows.RemoveAt(Grid\_Det\_Sal.CurrentRow.Index);

}

}

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

{

this.Close();

}

private void txtprecio\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)

{

txtmarca.Focus();

}

else if (e.KeyCode == Keys.Up)

{

txtcantidad.Focus();

}

}

private void txtcantidad\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)

{

btnNuevo.Focus();

}

else if (e.KeyCode == Keys.Up)

{

txtid.Focus();

}

}

private void bt\_agregar\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{ txtid.Focus(); }

}

private void txtid\_KeyPress(object sender, KeyPressEventArgs e)

{

if (Char.IsDigit(e.KeyChar))

{ e.Handled = false; }

else if (Char.IsControl(e.KeyChar))

{ e.Handled = false; }

else if (Char.IsSeparator(e.KeyChar))

{ e.Handled = false; }

else { e.Handled = true; }

}

private void txtprecio\_Leave(object sender, EventArgs e)

{

if (modulo1.IsNumeric(txtcantidad.Text.ToString()) && modulo1.IsNumeric(txtprecio.Text.ToString()))

{

costo();

}

}

private void txtprecio\_KeyPress(object sender, KeyPressEventArgs e)

{

if (Char.IsDigit(e.KeyChar))

{ e.Handled = false; }

else if (Char.IsControl(e.KeyChar))

{ e.Handled = false; }

else if (Char.IsSeparator(e.KeyChar))

{ e.Handled = false; }

else if (e.KeyChar == 46)

{ e.Handled = false; }

else { e.Handled = true; }

}

private void txtcantidad\_Leave(object sender, EventArgs e)

{

if (modulo1.IsNumeric(txtcantidad.Text.ToString()) && modulo1.IsNumeric(txtprecio.Text.ToString()))

{

costo();

R = conn.GetData("SELECT sum(de.existencia) FROM det\_entradas de, entradas e " +

" where de.identrada=e.id and e.idalmacen=" + cbalmacen.SelectedValue + " and idarticulo=" + txtid.Text + ";");

if (R.HasRows)

{

R.Read();

if (Int64.Parse(txtcantidad.Text.ToString()) > Int64.Parse(R[0].ToString()))

{

MessageBox.Show("No se cuenta con esa existencia en el almacen de ese articulo¡");

txtcantidad.Focus();

}

}

else

{

MessageBox.Show("No existe ese numero de articulo¡");

txtid.Focus();

}

R.Close();

}

}

void costo()

{

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

{

float costo = float.Parse(txtcantidad.Text) \* float.Parse(txtprecio.Text);

float iva1 = float.Parse(costo.ToString()) / configuracion.IVA;

txtcosto.Text = costo.ToString();

txtiva.Text = iva1.ToString();

}

}

private void txtmarca\_KeyDown\_1(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)

{

txtrequisicion.Focus();

}

else if (e.KeyCode == Keys.Up)

{

txtprecio.Focus();

}

}

private void txtrequisicion\_KeyDown\_1(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)

{

cbentrego.Focus();

}

else if (e.KeyCode == Keys.Up)

{

txtfecha.Focus();

}

}

public string suma\_costos()

{

string resp = "";

TOTAL = 0;

for (int fila = 0; fila < Grid\_Det\_Sal.Rows.Count - 1; fila++)

{

TOTAL = TOTAL + double.Parse(Grid\_Det\_Sal.Rows[fila].Cells[4].Value.ToString());

}

resp = TOTAL.ToString();

return resp;

}

private void cbtipo\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)

{

txtfecha.Focus();

}

}

private void txtfecha\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)

{

txtrequisicion.Focus();

}

else if (e.KeyCode == Keys.Up)

{

cbtipo.Focus();

}

}

private void cbentrego\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)

{

cbarea.Focus();

}

else if (e.KeyCode == Keys.Up)

{

txtrequisicion.Focus();

}

}

private void cbarea\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)

{

cbalmacen.Focus();

}

else if (e.KeyCode == Keys.Up)

{

cbentrego.Focus();

}

}

private void cbalmacen\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)

{

txtid.Focus();

}

else if (e.KeyCode == Keys.Up)

{

cbarea.Focus();

}

}

Boolean grabar\_grid()

{

DateTime fec = DateTime.Now;

double nuevo\_costo = 0, nueva\_exis = 0, sal = 0, precio = 0;

Int64 ID\_ART = 0;

DateTime caducidad;

string f = " order by de.identrada asc", marca = "";

if (configuracion.METODO == "PE")

{ f = " order by de.identrada desc"; }

for (int fila = 0; fila < Grid\_Det\_Sal.Rows.Count - 1; fila++)

{

fec = DateTime.Parse(txtfecha.Value.ToString());

ID\_ART = Int64.Parse(Grid\_Det\_Sal.Rows[fila].Cells[0].Value.ToString());

nueva\_exis = Int32.Parse(Grid\_Det\_Sal.Rows[fila].Cells[2].Value.ToString());

precio = double.Parse(Grid\_Det\_Sal.Rows[fila].Cells[3].Value.ToString());

nuevo\_costo = double.Parse(Grid\_Det\_Sal.Rows[fila].Cells[4].Value.ToString());

llena\_grid("select de.existencia as existencia, de.precio as precio, de.caducidad as caducidad, de.marca, de.id as id from det\_entradas de, entradas e " +

" where de.identrada=e.id and de.existencia>0 and de.idarticulo=" + ID\_ART + " and e.idalmacen=" + cbalmacen.SelectedValue + f.ToString());

for (int fila2 = 0; (fila2 < Grid\_Articulos.Rows.Count - 1 && nueva\_exis > 0); fila2++)

{

if (Int64.Parse(Grid\_Articulos.Rows[fila2].Cells["existencia"].Value.ToString()) >= nueva\_exis)

{ sal = nueva\_exis; }

else

{ sal = Int64.Parse(Grid\_Articulos.Rows[fila2].Cells["existencia"].Value.ToString()); }

caducidad = DateTime.Parse(Grid\_Articulos.Rows[fila2].Cells["caducidad"].Value.ToString());

marca = Grid\_Articulos.Rows[fila2].Cells["marca"].Value.ToString();

nueva\_exis = nueva\_exis - sal;

if (configuracion.METODO != "PP")

{

//cambiar precio y costo

precio = double.Parse(Grid\_Articulos.Rows[fila2].Cells["precio"].Value.ToString());

nuevo\_costo = precio \* sal;

//MessageBox.Show("cambiar precio y costo");

}

string str = "insert into det\_salidas (idsalida,idarticulo," +

"cantidad,precio,totalprecio,marca,caducidad) " +

"values(" + ID\_SAL + "," + ID\_ART + "," + sal + "," + precio + "," + nuevo\_costo + ",'" +

marca.ToString() + "','" + caducidad.ToString("yyyy-MM-dd") + "')";

resp = conn.Executa(str);

if (!resp) { conn.fallo(); return false; }

//GUARDAR EN LA TABLA KARDEX EL MOV DE ENTRADA

str = "insert into kardex (idtabla,idarticulo,entrada,salida,precio,costo,fecha)" +

"values(" + ID\_SAL + "," + ID\_ART + ",0," + sal + "," + precio + "," +

nuevo\_costo + ",'" + caducidad.ToString("yyyy-MM-dd") + "')";

//MessageBox.Show(str);

resp = conn.Executa(str);

if (!resp) { conn.fallo(); return false; }

//CALCULAR PRECIO PROMEDIO

str = "update catarticulos set existencia=existencia-" + sal + " where id=" + ID\_ART;

resp = conn.Executa(str);

if (!resp) { conn.fallo(); return false; }

str = "update det\_entradas set existencia=existencia-"

+ sal + " where id=" + Grid\_Articulos.Rows[fila2].Cells["id"].Value.ToString();

resp = conn.Executa(str);

if (!resp) { conn.fallo(); return false; }

}//segundo for

}//primer for

//Inventario.ActiveForm=true;

return true;

}

}

}