Universidad Mariano Gálvez

Diseño de Sistemas

Ing. Eduardo del Águila

Manual tecnico

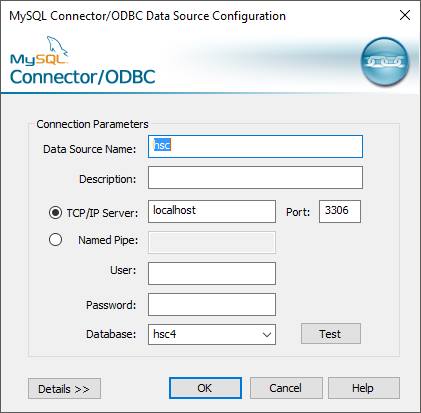
Grupo #2

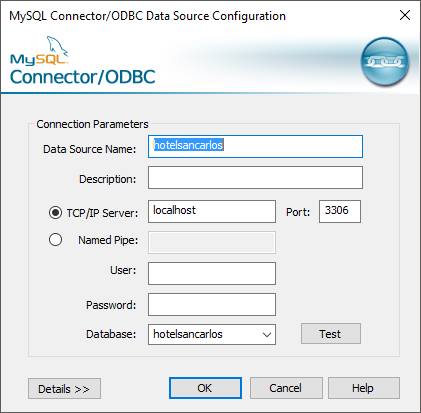
Guatemala, 3 de noviembre de 2016

manual tecnico

# Pasos para instalación

1. Instalar xampp
2. Instalar los conectores de mysql para la conexión con ODBC
3. Crear los dos odbc dependiendo de las siguientes imágenes





1. Importar las dos bases de datos hsc4 y hotelsancarlos (esto es debido a la implementacion de seguridad)
2. Instalar la aplicación para su uso

# Programadora: Ana Lilian Estevez Alvarado

1. Insertar nuevo registro

* editar = false;
* CapaNegocio fn = new CapaNegocio();
* fn.ActivarControles(this);
* fn.LimpiarComponentes(this);

1. Guardar nuevo registro

* CapaNegocio fn = new CapaNegocio();
* TextBox[] textbox = { txt\_nombre, txt\_direccion, txt\_region, txt\_estrellas, txt\_nit, txt\_correo, txt\_telefono };
* DataTable datos = fn.construirDataTable(textbox);
* if (datos.Rows.Count == 0)
* {
* MessageBox.Show("Hay campos vacios");
* }
* else
* {
* string tabla = "empresa";
* if (editar)
* {
* fn.modificar(datos, tabla, atributo, codigo);
* }
* else
* {
* fn.insertar(datos, tabla);
* }
* fn.LimpiarComponentes(this);
* fn.ActualizarGrid(datagridantes, "select \* from empresa", tabla);
* }

1. Eliminar registro

* try
* {
* editar = true;
* atributo = "id\_empresa\_pk";
* //codigo = this.dataGridView1.CurrentRow.Cells[0].Value.ToString();
* codigo = datagridantes.CurrentRow.Cells[0].Value.ToString();
* TextBox[] textbox = { txt\_nombre, txt\_direccion, txt\_region, txt\_estrellas, txt\_nit, txt\_correo, txt\_telefono };
* CapaNegocio fn = new CapaNegocio();
* fn.llenartextbox(textbox, datagridantes);
* }
* catch
* {
* MessageBox.Show("No ha seleccionado ningun registro a editar");
* }

1. Enviar por parámetro el nombre de la tabla a utilizar

* if (tablaa == "empresa")
* {
* if (empresa == null)
* {
* empresa = new Frm\_MantenimientoEmpresa(dataGridView1);
* empresa.MdiParent = padre;
* empresa.FormClosed += new FormClosedEventHandler(empresa\_FormClosed);
* empresa.Show();
* }
* else
* {
* empresa.Activate();
* }
* }
* else if (tablaa == "tipo")
* {
* if (tipohab == null)
* {
* tipohab = new Frm\_MantemientoTipoHabitacion(dataGridView1);
* tipohab.MdiParent = padre;
* tipohab.FormClosed += new FormClosedEventHandler(tipohab\_FormClosed);
* tipohab.Show();
* }
* else
* {
* tipohab.Activate();
* }
* }
* else if (tablaa == "habitacion")
* {
* if (habit == null)
* {
* habit = new Frm\_MantenimientoHabitaciones(dataGridView1);
* habit.MdiParent = padre;
* habit.FormClosed += new FormClosedEventHandler(habit\_FormClosed);
* habit.Show();
* }
* else
* {
* habit.Activate();
* }
* }
* else if (tablaa == "salon")
* {
* if (salo == null)
* {
* salo = new Frm\_MantenimientoSalones(dataGridView1);
* salo.MdiParent = padre;
* salo.FormClosed += new FormClosedEventHandler(salo\_FormClosed);
* salo.Show();
* }
* else
* {
* salo.Activate();
* }
* }
* else if (tablaa == "promocion")
* {
* if (promo == null)
* {
* promo = new Frm\_Promocion(dataGridView1);
* promo.MdiParent = padre;
* promo.FormClosed += new FormClosedEventHandler(promo\_FormClosed);
* promo.Show();
* }
* else
* {
* promo.Activate();
* }
* }
* else if (tablaa == "invitado")
* {
* if (invitado == null)
* {
* invitado = new Frm\_Invitado(dataGridView1);
* invitado.MdiParent = padre;
* invitado.FormClosed += new FormClosedEventHandler(invitado\_FormClosed);
* invitado.Show();
* }
* else
* {
* invitado.Activate();
* }
* }
* else if (tablaa == "buzon")
* {
* if (suge == null)
* {
* suge = new Frm\_Sugerencia(dataGridView1);
* suge.MdiParent = padre;
* suge.FormClosed += new FormClosedEventHandler(suge\_FormClosed);
* suge.Show();
* }
* else
* {
* suge.Activate();
* }
* }
* else if (tablaa == "problema")
* {
* if (proble == null)
* {
* proble = new Frm\_Problema(dataGridView1);
* proble.MdiParent = padre;
* proble.FormClosed += new FormClosedEventHandler(proble\_FormClosed);
* proble.Show();
* }
* else
* {
* proble.Activate();
* }
* }
* else if (tablaa == "obj\_perdido")
* {
* if (objolvi == null)
* {
* objolvi = new Frm\_ObjetosOlvidados(dataGridView1);
* objolvi.MdiParent = padre;
* objolvi.FormClosed += new FormClosedEventHandler(objolvi\_FormClosed);
* objolvi.Show();
* }
* else
* {
* objolvi.Activate();
* }
* }

1. Proceso de check-In

* clienteseleccionado = cbo\_cliente.SelectedValue.ToString();
* //MessageBox.Show(clienteseleccionado);
* DataSet ds1 = new DataSet();
* String Query1 = "select rh.fecha\_entreda, rh.id\_cliente\_pk, rh.id\_habitacion\_pk, h.nombre from reservacion\_habitacion rh, habitacion h where rh.id\_cliente\_pk="+clienteseleccionado+" and rh.id\_habitacion\_pk=h.id\_habitacion\_pk and rh.fecha\_entreda=curdate();";
* MySqlDataAdapter dad1 = new MySqlDataAdapter(Query1, con.rutaconectada());
* dad1.Fill(ds1, "reservacion\_habitacion");
* checkedListBox1.DataSource = ds1.Tables[0].DefaultView;
* checkedListBox1.ValueMember = ("id\_habitacion\_pk");
* checkedListBox1.DisplayMember = ("nombre");

1. Creacion de MDI

* if (grid == null)
* {
* grid = new Frm\_Grid("habitacion");
* grid.MdiParent = this;
* grid.FormClosed += new FormClosedEventHandler(grid\_FormClosed);
* grid.Show();
* }
* else
* {
* grid.Activate();
* }

1. Llenado de Combobox

* DataSet ds1 = new DataSet();
* String Query1 = "select id\_empresa\_pk, nombre from empresa;";
* MySqlDataAdapter dad1 = new MySqlDataAdapter(Query1, con.rutaconectada());
* dad1.Fill(ds1, "empresa");
* cbo\_Empresa.DataSource = ds1.Tables[0].DefaultView;
* cbo\_Empresa.ValueMember = ("id\_empresa\_pk");
* cbo\_Empresa.DisplayMember = ("nombre");

# Programador: Rodrigo Sifontes

1. FUNCIONES DE LA CLASE CONEXION Y MANIPULACION

Permite llenar los combobox con los valores de una tabla.

public void getColumnas(ComboBox cb,String tabla,String parametro)

{

OdbcCommand cm = new OdbcCommand("SELECT "+parametro+" FROM " + tabla +";" , rutaconectada());

OdbcDataReader adaptador = cm.ExecuteReader();

while(adaptador.Read())

{

cb.Items.Add(adaptador[parametro].ToString());

}

adaptador.Close();

}

//llena todos los textox

public void llenartext(TextBox tx,String Query,String parametro)

{

OdbcCommand cm = new OdbcCommand(Query, rutaconectada());

OdbcDataReader adaptador = cm.ExecuteReader();

if(adaptador.Read()==true)

{

tx.Text=(adaptador[parametro].ToString());

}

adaptador.Close();

}

// permite guardar el resultado de una consulta en una variable

public string verificacionhabitaicon(String Query)

{

DataTable dt = new DataTable();

OdbcCommand comando = new OdbcCommand(Query, rutaconectada());

OdbcDataAdapter adaptador = new OdbcDataAdapter(comando);

adaptador.Fill(dt);

DataRow row = dt.Rows[0];

String id\_cat = row[0].ToString().Trim();

return id\_cat;

}

1. CODIGO DE RESERVACION

// llena el checklist con las habitaciones disponibles para ciertas fechas y a la vez perteneces al tipo de habitación que se seleccionó.

public void llenarcheck(ComboBox cb, string entrada, String salida)

{

String query1 = "select count(\*) from habitacion;";

int validacion1 = Convert.ToInt32(conexion.verificacionhabitaicon(query1));

for (int a=1; a<=validacion1; a++)

{

String query2 = "Select count(\*) from habitacion a, tipo\_habitacion b where a.id\_habitacion\_pk=" + a + " and b.nivel\_tipo='" + cb.Text + "'and a.id\_tipo\_Pk = b.id\_tipo\_pk and a.estado='activo';";

int validacion2 = Convert.ToInt32(conexion.verificacionhabitaicon(query2));

if(validacion2>0)

{

// filtrps para verificar si una habitación esta disponible o no

String query3 = "SELECT count(\*) from reservacion\_habitacion where id\_habitacion\_pk =" + a + " and fecha\_entreda >= '" + entrada + "'and fecha\_entreda < '" + salida + "' and estado='reservado' or id\_habitacion\_pk =" + a + " and fecha\_entreda >= '" + entrada + "'and fecha\_entreda < '" + salida + "' and estado='ocupado' or id\_habitacion\_pk =" + a + " and fecha\_entreda <'" + entrada + "' and fecha\_salida>'" + entrada + "' and estado='reservado' or id\_habitacion\_pk =" + a + " and fecha\_entreda <'" + entrada + "' and fecha\_salida>'" + entrada + "' and estado='ocupado' or id\_habitacion\_pk =" + a + " and fecha\_entreda<'" + entrada + "' and fecha\_salida > '" + salida + "' and estado ='reservado'or id\_habitacion\_pk =" + a + " and fecha\_entreda<'" + entrada + "' and fecha\_salida > '" + salida + "' and estado ='ocupado' or id\_habitacion\_pk =" + a + " and fecha\_entreda = '" + entrada + "'and fecha\_salida= '" + salida + "' and estado='reservado' or id\_habitacion\_pk =" + a + " and fecha\_entreda = '" + entrada + "'and fecha\_salida ='" + salida + "' and estado='ocupado';";

int validacion3 = Convert.ToInt32(conexion.verificacionhabitaicon(query3));

if (validacion3==0)

{

String query4 = "Select nombre from habitacion where id\_habitacion\_pk=" + a + ";";

ckl\_reservacion.Items.Add(conexion.verificacionhabitaicon(query4));

}

}

}

}

// guarda las habitaciones seleccionada del checklist en una lista para luego recorrerla y así insertar en la tabla las reservaciones con las distintas habitaciones

private void btn\_guardar\_Click(object sender, EventArgs e)

{

CapaNegocio fn = new CapaNegocio();

if (txt\_cant1.Text != "" && txt\_acompañantes.Text != "" && txt\_niñosacompañantes.Text != "" && cbo\_codigo.Text != "")

{

int total\_adultos = Convert.ToInt32(txt\_adultos1.Text) \* Convert.ToInt32(txt\_cant1.Text);

int total\_niños = Convert.ToInt32(txt\_niño1.Text) \* Convert.ToInt32(txt\_cant1.Text);

int niños\_habitacion = Convert.ToInt32(txt\_niñosacompañantes.Text);

int adultos\_habitacion = Convert.ToInt32(txt\_acompañantes.Text);

if (total\_adultos >= adultos\_habitacion && total\_niños >= niños\_habitacion)

{

int s = Convert.ToInt32(txt\_cant1.Text);

if (ckl\_reservacion.CheckedItems.Count != 0 && ckl\_reservacion.CheckedItems.Count == s)

{

// If so, loop through all checked items and print results.

for (int x = 0; x <= ckl\_reservacion.CheckedItems.Count - 1; x++)

{

cuartosseleccionados.Add(ckl\_reservacion.CheckedItems[x].ToString());

contador = contador + 1;

}

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

{

String query = "select id\_habitacion\_pk from habitacion where nombre='" + cuartosseleccionados[x] + "';";

txt\_codigocliente.Text = cbo\_codigo.SelectedItem.ToString();

fechaentrada.Format = DateTimePickerFormat.Custom;

fechaentrada.CustomFormat = "yyyy-MM-dd";

string rs = fechaentrada.Text;

fechaentrada.Format = DateTimePickerFormat.Custom;

fechaentrada.CustomFormat = "dd-MM-yyyy";

txt\_fechaentrada.Text = rs;

fechasalida.Format = DateTimePickerFormat.Custom;

fechasalida.CustomFormat = "yyyy-MM-dd";

string rf = fechasalida.Text;

fechasalida.Format = DateTimePickerFormat.Custom;

fechasalida.CustomFormat = "dd-MM-yyyy";

txt\_fechasalida.Text = rf;

txt\_idhabitacion.Text = conexion.verificacionhabitaicon(query);

txt\_estadoreservacion.Text = "Reservado";

txt\_fechasistema.Text = fechaHoy.ToString("yyyy-MM-dd");

TextBox[] textbox = { txt\_codigocliente, txt\_idhabitacion, txt\_fechaentrada, txt\_fechasalida, txt\_fechasistema, txt\_estadoreservacion };

DataTable datos = fn.construirDataTable(textbox);

if (datos.Rows.Count == 0)

{

MessageBox.Show("Hay campos vacios");

}

else

{

string tabla = "reservacion\_habitacion";

if (editar)

{

fn.modificar(datos, tabla, atributo, codigo);

}

else

{

fn.insertar(datos, tabla);

}

}

}

fn.ActualizarGrid(datagridantes, "select \* from reservacion\_habitacion", "reservacion\_habitacion");

contador = 0;

cuartosseleccionados.Clear();

limpiar();

}

else

{

MessageBox.Show("Debe seleccionar la cantidad de habitaciones seleccionadas ");

}

}

else

{

MessageBox.Show("Debe seleccionar mas habitaciones , el numero de acompañantes es mayor a la capacidad de la habitacion");

}

}

else

{

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

}

}

// llena los textos automáticamente en el área de reservación al momento de seleccionar un tipo.

public void cargar1()

{

String r1 = "SELECT ninio\_tipo,adulto\_tipo,especificaciones\_tipo from tipo\_habitacion WHERE nivel\_tipo = '" + cbo\_tipo1.Text + "';";

if (cbo\_tipo1.Text != "")

{

conexion.llenartext(txt\_niño1, r1, "ninio\_tipo");

conexion.llenartext(txt\_adultos1, r1, "adulto\_tipo");

conexion.llenartext(txt\_descp1, r1, "especificaciones\_tipo");

txt\_niño1.ReadOnly = true;

txt\_adultos1.ReadOnly = true;

txt\_descp1.ReadOnly = true;

}

}

# Programador: Walter Recinos

**Manual técnico - FOLIO**

//==================================================================================

//----------------------LLENADO DE COMBOBOX-----------------------

//==================================================================================

//programador:walter Recinos Rosales

//fecha inicio: 12-10-2016

//fecha finalizacion: 19-10-2016

private void cb1(string id\_folio)

{

try

{

con.Conectar();

//se inicia un DataSet

DataSet ds = new DataSet();

//se indica la consulta en sql

String Query = "select id\_cuenta\_pagar\_pk from folio WHERE estado = 'Pendiente' AND id\_cliente\_pk = '"+ id\_folio + "';";

OdbcDataAdapter dad = new OdbcDataAdapter(Query, con.rutaconectada());

//se indica con quu tabla se llena

dad.Fill(ds, "folio");

cbo\_folio\_hab.DataSource = ds.Tables[0].DefaultView;

//indicamos el valor de los miembros

cbo\_folio\_hab.ValueMember = ("id\_cuenta\_pagar\_pk");

//se indica el valor a desplegar en el combobox

cbo\_folio\_hab.DisplayMember = ("id\_cuenta\_pagar\_pk");

con.Desconectar();

}

catch (Exception ex)

{

MessageBox.Show("Los datos no cargaron de forma correcta" + ex, "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void cb2(string id\_clientefiltro)

{

//string clientefiltro = cbo\_clientefiltro.SelectedValue.ToString();

try

{

con.Conectar();

//se inicia un DataSet

DataSet ds = new DataSet();

//se indica la consulta en sql

String Query = "select id\_reserhabit\_pk from reservacion\_habitacion where id\_cliente\_pk = '"+ id\_clientefiltro + "';";

//String Query = "select id\_reserhabit\_pk from reservacion\_habitacion ;";

OdbcDataAdapter dad = new OdbcDataAdapter(Query, con.rutaconectada());

//se indica con quu tabla se llena

dad.Fill(ds, "reservacion\_habitacion");

cbo\_habitacion.DataSource = ds.Tables[0].DefaultView;

//indicamos el valor de los miembros

cbo\_habitacion.ValueMember = ("id\_reserhabit\_pk");

//se indica el valor a desplegar en el combobox

cbo\_habitacion.DisplayMember = ("id\_reserhabit\_pk");

con.Desconectar();

}

catch (Exception )

{

MessageBox.Show("Los datos no cargaron de forma correcta", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

//===============================================================================

//--------------------------------------INSERTAR TABLAS------------------------------------------------------------

//===============================================================================

//programador:walter Recinos Rosales

//fecha inicio: 12-10-2016

//fecha finalizacion: 19-10-2016

public void insertartodo()

{

if (tabControl1.SelectedTab == tabPage1)

{

try

{

string cliente = cbo\_cliente\_folio.SelectedValue.ToString();

con.Conectar();

String Squery = "insert into folio (estado,fecha\_ingreso,costo,id\_cliente\_pk) values('Pendiente',Sysdate(),'0.00','" + cliente + "');";

con.EjecutarQuery(Squery);

con.actualizargrid(Squeery4, dgv\_lista\_folio);

//nombre\_columna();

con.Desconectar();

//MessageBox.Show("Folio");

string clientefiltro = cbo\_clientefiltro.SelectedValue.ToString();

cb1(clientefiltro);//carga de combobox de folio1

cb3(clientefiltro);//carga de combobox de folio2

cb4(clientefiltro);//carga de combobox de folio3

cb5(clientefiltro);//carga de combobox de folio4

}

catch

{

MessageBox.Show("Llene todos los campos por favor", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

if (tabControl1.SelectedTab == tabPage2)

{

if (cbo\_folio\_hab.Text == "" || cbo\_habitacion.Text == "")

{

MessageBox.Show("Llene todos los campos por favor", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

string folio = cbo\_folio\_hab.SelectedValue.ToString();

string habitacion = cbo\_habitacion.SelectedValue.ToString();

if (Comparar\_folio\_con\_credito(cbo\_folio\_hab.Text) == true)

{

}

else

{

MessageBox.Show("HA EXEDIDO LIMITE DE CREDITO", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

if (verificar\_reservacion(folio, habitacion) == false)

{

string nombre = cbo\_habitacion.Text;

DateTime tiempo = DateTime.Now;

con.Conectar();

DataTable dt2 = new DataTable();

String sQuery2 = "SELECT id\_habitacion\_pk FROM reservacion\_habitacion WHERE id\_reserhabit\_pk = '" + habitacion + "'";

OdbcCommand comando2 = new OdbcCommand(sQuery2, con.rutaconectada());

OdbcDataAdapter adaptador2 = new OdbcDataAdapter(comando2);

adaptador2.Fill(dt2);

DataRow fila2 = dt2.Rows[0];

string sid2 = Convert.ToString(fila2[0]);

MessageBox.Show(sid2);

con.Desconectar();

con.Conectar();

DataTable dt = new DataTable();

String sQuery = "SELECT id\_tipo\_pk FROM habitacion WHERE id\_habitacion\_pk = '" + sid2 + "'";

OdbcCommand comando = new OdbcCommand(sQuery, con.rutaconectada());

OdbcDataAdapter adaptador = new OdbcDataAdapter(comando);

adaptador.Fill(dt);

DataRow fila = dt.Rows[0];

string sid = Convert.ToString(fila[0]);

MessageBox.Show(sid);

con.Desconectar();

con.Conectar();

DataTable dt1 = new DataTable();

String sQuery1 = "SELECT costo\_tipo FROM tipo\_habitacion WHERE id\_tipo\_pk = '" + sid + "'";

OdbcCommand comando1 = new OdbcCommand(sQuery1, con.rutaconectada());

OdbcDataAdapter adaptador1 = new OdbcDataAdapter(comando1);

adaptador1.Fill(dt1);

DataRow fila1 = dt1.Rows[0];

string sid1 = Convert.ToString(fila1[0]);

MessageBox.Show(sid1);

con.Desconectar();

con.Conectar();

String Squery = "insert into folio\_habitacion (id\_reserhabit\_pk,id\_cuenta\_pagar\_pk,costo,fecha) values('" + habitacion + "','" + folio + "','" + sid1 + "','" + tiempo + "');";

con.EjecutarQuery(Squery);

con.actualizargrid(Squeery, dgv\_folio\_habitacion);

//nombre\_columna();

con.Desconectar();

con.guardar\_costo\_folio(cbo\_folio\_hab.Text);

con.actualizargrid(Squeery4, dgv\_lista\_folio);

detalle\_folio\_insertar\_habitacion(sid1, nombre, folio, tiempo);

}

else

{

MessageBox.Show("ESTA ACCION YA EXISTE", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

if (tabControl1.SelectedTab == tabPage3)

{

if (cbo\_folio\_salon.Text == "" || cbo\_salon.Text == "")

{

MessageBox.Show("Llene todos los campos por favor", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

string folio = cbo\_folio\_salon.SelectedValue.ToString();

string salon = cbo\_salon.SelectedValue.ToString();

if (Comparar\_folio\_con\_credito(cbo\_folio\_salon.Text) == true)

{

}

else

{

MessageBox.Show("HA EXEDIDO LIMITE DE CREDITO", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

if (verificar\_evento(folio, salon) == false)

{

string nombre = cbo\_salon.Text;

DateTime tiempo = DateTime.Now;

con.Conectar();

DataTable dt = new DataTable();

String sQuery = "SELECT costo FROM evento WHERE id\_evento\_pk = '" + salon + "'";

OdbcCommand comando = new OdbcCommand(sQuery, con.rutaconectada());

OdbcDataAdapter adaptador = new OdbcDataAdapter(comando);

adaptador.Fill(dt);

DataRow fila = dt.Rows[0];

string sid = Convert.ToString(fila[0]);

//MessageBox.Show(sid);

con.Desconectar();

con.Conectar();

String Squery = "insert into folio\_salon (id\_evento\_pk,id\_cuenta\_pagar\_pk,costo,fecha) values('" + salon + "','" + folio + "','" + sid + "','" + tiempo + "');";

con.EjecutarQuery(Squery);

con.actualizargrid(Squeery1, dgv\_folio\_salon);

//nombre\_columna();

con.Desconectar();

con.guardar\_costo\_folio(cbo\_folio\_salon.Text);

con.actualizargrid(Squeery4, dgv\_lista\_folio);

detalle\_folio\_insertar\_salon(sid, nombre, folio, tiempo);

}

else

{

MessageBox.Show("ESTA ACCION YA EXISTE", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

if (tabControl1.SelectedTab == tabPage4)

{

if (cbo\_folio\_articulo.Text == "" || cbo\_articulo.Text == "")

{

MessageBox.Show("Llene todos los campos por favor", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

if (Comparar\_folio\_con\_credito(cbo\_folio\_articulo.Text) == true)

{

string folio = cbo\_folio\_articulo.SelectedValue.ToString();

string bien = cbo\_articulo.SelectedValue.ToString();

string nombre = cbo\_articulo.Text;

DateTime tiempo4 = DateTime.Now;

con.Conectar();

DataTable dt = new DataTable();

String sQuery = "SELECT costo FROM bien WHERE id\_bien\_pk = '" + bien + "'";

OdbcCommand comando = new OdbcCommand(sQuery, con.rutaconectada());

OdbcDataAdapter adaptador = new OdbcDataAdapter(comando);

adaptador.Fill(dt);

DataRow fila = dt.Rows[0];

string sid = Convert.ToString(fila[0]);

//MessageBox.Show(sid);

con.Desconectar();

con.Conectar();

String Squery = "insert into folio\_bien (id\_bien\_pk,id\_cuenta\_pagar\_pk,costo,fecha) values('" + bien + "','" + folio + "','" + sid + "','" + tiempo4 + "');";

con.EjecutarQuery(Squery);

con.actualizargrid(Squeery2, dgv\_bien\_folio);

//nombre\_columna();

con.Desconectar();

con.guardar\_costo\_folio(cbo\_folio\_articulo.Text);

con.actualizargrid(Squeery4, dgv\_lista\_folio);

detalle\_folio\_insertar\_bien(sid, nombre, folio, tiempo4);

}

else

{

MessageBox.Show("HA EXEDIDO LIMITE DE CREDITO", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

if (tabControl1.SelectedTab == tabPage5)

{

if (cbo\_folio\_paquete.Text == "" || cbo\_paquete.Text == "")

{

MessageBox.Show("Llene todos los campos por favor", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

if (Comparar\_folio\_con\_credito(cbo\_folio\_paquete.Text) == true)

{

string folio = cbo\_folio\_paquete.SelectedValue.ToString();

string paquete = cbo\_paquete.SelectedValue.ToString();

string nombre = cbo\_paquete.Text;

DateTime tiempo = DateTime.Now;

con.Conectar();

DataTable dt = new DataTable();

String sQuery = "SELECT costo FROM promocion WHERE id\_promocion\_pk = '" + paquete + "'";

OdbcCommand comando = new OdbcCommand(sQuery, con.rutaconectada());

OdbcDataAdapter adaptador = new OdbcDataAdapter(comando);

adaptador.Fill(dt);

DataRow fila = dt.Rows[0];

string sid = Convert.ToString(fila[0]);

//MessageBox.Show(sid);

con.Desconectar();

con.Conectar();

String Squery = "insert into folio\_promocion (id\_promocion\_pk,id\_cuenta\_pagar\_pk,costo,fecha) values('" + paquete + "','" + folio + "','" + sid + "','" + tiempo + "');";

con.EjecutarQuery(Squery);

con.actualizargrid(Squeery3, dgv\_folio\_paquete);

//nombre\_columna();

con.Desconectar();

con.guardar\_costo\_folio(cbo\_folio\_paquete.Text);

con.actualizargrid(Squeery4, dgv\_lista\_folio);

detalle\_folio\_insertar\_promo(sid, nombre, folio, tiempo);

}

else

{

MessageBox.Show("HA EXEDIDO LIMITE DE CREDITO", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

}

//---------------------------------------consulta de precios otras tablas---------------------------------------------

//Programador: walter recinos

//fecha inicio: 21-10-2016

//fecha finalizacion: 02-11-2016

public double consulta\_subtotal\_folio\_bien(string dato)

{

try

{

Conectar();

DataTable dt = new DataTable();

String sQuery = "SELECT SUM(costo) FROM folio\_bien WHERE id\_cuenta\_pagar\_pk = '" + dato + "'";

OdbcCommand comando = new OdbcCommand(sQuery, rutaconectada());

OdbcDataAdapter adaptador = new OdbcDataAdapter(comando);

adaptador.Fill(dt);

DataRow fila = dt.Rows[0];

string sid = Convert.ToString(fila[0]);

//MessageBox.Show("El Precio del articulo es: Q." + sid);

double cantidad = Convert.ToDouble(sid);

Desconectar();

return cantidad;

}

catch (Exception)

{

return 0;

}

}

//---------------------------------------------------------------------------------------------------------------------

//Programador: walter recinos

//fecha inicio: 21-10-2016

//fecha finalizacion: 02-11-2016

public double consulta\_subtotal\_folio\_paquete(string dato)

{

try

{

Conectar();

DataTable dt = new DataTable();

String sQuery = "SELECT SUM(costo) FROM folio\_promocion WHERE id\_cuenta\_pagar\_pk = '" + dato + "'";

OdbcCommand comando = new OdbcCommand(sQuery,rutaconectada());

OdbcDataAdapter adaptador = new OdbcDataAdapter(comando);

adaptador.Fill(dt);

DataRow fila = dt.Rows[0];

string sid = Convert.ToString(fila[0]);

//MessageBox.Show("El Precio de la promocion es: Q." + sid);

double cantidad = Convert.ToDouble(sid);

Desconectar();

return cantidad;

}

catch (Exception)

{

return 0;

}

}

//---------------------------------------------------------------------------------------------------------------------

//Programador: walter recinos

//fecha inicio: 21-10-2016

//fecha finalizacion: 02-11-2016

public double consulta\_subtotal\_folio\_habitacion(string dato)

{

try

{

Conectar();

DataTable dt = new DataTable();

String sQuery = "SELECT SUM(costo) FROM folio\_habitacion WHERE id\_cuenta\_pagar\_pk = '" + dato + "'";

OdbcCommand comando = new OdbcCommand(sQuery,rutaconectada());

OdbcDataAdapter adaptador = new OdbcDataAdapter(comando);

adaptador.Fill(dt);

DataRow fila = dt.Rows[0];

string sid = Convert.ToString(fila[0]);

//MessageBox.Show("Costo subtotal habitacion: Q." + sid);

Desconectar();

double cantidad = Convert.ToDouble(sid);

return cantidad;

}

catch (Exception)

{

return 0;

}

}

//--------------------------------------------------------------------------------------------------------------------

//Programador: walter recinos

//fecha inicio: 21-10-2016

//fecha finalizacion: 02-11-2016

public double consulta\_subtotal\_folio\_salon(string dato)

{

try

{

Conectar();

DataTable dt = new DataTable();

String sQuery = "SELECT SUM(costo) FROM folio\_salon WHERE id\_cuenta\_pagar\_pk = '" + dato + "'";

OdbcCommand comando = new OdbcCommand(sQuery,rutaconectada());

OdbcDataAdapter adaptador = new OdbcDataAdapter(comando);

adaptador.Fill(dt);

DataRow fila = dt.Rows[0];

string sid = Convert.ToString(fila[0]);

//MessageBox.Show("Costo subtotal de salon: Q." + sid);

Desconectar();

double cantidad = Convert.ToDouble(sid);

return cantidad;

}

catch (Exception)

{

return 0;

}

}

//---------------------------------------------------------------------------------------------------------------------

//este sirve para contar todos los totales por folio

//Programador: walter recinos

//fecha inicio: 21-10-2016

//fecha finalizacion: 02-11-2016

public double subtotal\_por\_folio(string campo)

{

double n1 = consulta\_subtotal\_folio\_habitacion(campo);

double n2 = consulta\_subtotal\_folio\_salon(campo);

double n3 = consulta\_subtotal\_folio\_bien(campo);

double n4 = consulta\_subtotal\_folio\_paquete(campo);

double subtotal = n1 + n2 + n3 + n4;

string Ssubtotal = Convert.ToString(subtotal);

//MessageBox.Show("El Total es: Q." + Ssubtotal);

return subtotal;

}

//--------------------------------------------------------------------------------------------------------------------

//este sirve para ejecultar subtotal\_por\_folio y almacena el total en el campo costo de la tabla folio

//Programador: walter recinos

//fecha inicio: 21-10-2016

//fecha finalizacion: 02-11-2016

public void guardar\_costo\_folio(string campo\_referencia)

{

double total = subtotal\_por\_folio(campo\_referencia);

Conectar();

String Squery = "update folio set costo ='" + total + "' where id\_cuenta\_pagar\_pk ='" + campo\_referencia + "'";

EjecutarQuery1(Squery);

//nombre\_columna();

Desconectar();

}

//--------------------------suma con detalle folio--------------------------------------------------------------------

//Programador: walter recinos

//fecha inicio: 21-10-2016

//fecha finalizacion: 02-11-2016

public void actualizar\_total(string dato)

{

try

{

Conectar();

DataTable dt = new DataTable();

String sQuery = "SELECT sum(costo) FROM detalle\_folio WHERE id\_cuenta\_pagar\_pk = '" + dato + "'";

OdbcCommand comando = new OdbcCommand(sQuery, rutaconectada());

OdbcDataAdapter adaptador = new OdbcDataAdapter(comando);

adaptador.Fill(dt);

DataRow fila = dt.Rows[0];

string sid = Convert.ToString(fila[0]);

//MessageBox.Show("Costoooooo subtotal de salon: Q." + sid);

Desconectar();

double cantidad = Convert.ToDouble(sid);

Conectar();

String Squery = "update folio set costo ='" + cantidad + "' where id\_cuenta\_pagar\_pk ='" + dato + "'";

EjecutarQuery1(Squery);

//nombre\_columna();

Desconectar();

}

catch (Exception)

{

double cantidad = 0;

Conectar();

String Squery = "update folio set costo ='" + cantidad + "' where id\_cuenta\_pagar\_pk ='" + dato + "'";

EjecutarQuery1(Squery);

//nombre\_columna();

Desconectar();

}

}

}

}