# **if (DropDownList2.SelectedItem.Value.ToString()=="0")**

# **cat="tc";**

# **QUERIDO VISITANTE: SIÉNTETE LIBRE DE EDITAR O DE AGREGAR LO QUE CREAS NECESARIO. PUEDES AGREGAR A MÁS GENTE PERO SÓLO COMO COMENTADORES. VOY A ACEPTAR LA MAYORÍA DE LOS CAMBIOS, ERES COMENTADOR PORQUE NO QUIERO QUE POR ERROR SE BORRE TODO. ESPERO TE AYUDE SHARING IS CARING:D**

# 

# 

# WPF

## Conexión:

Proyecto - Agregar nuevo origen de datos - Microsoft SQL Server

class Conexion

{

SqlConnection cnn;

SqlCommand cmd;

SqlDataReader dr;

public static SqlConnection agregarConexion()

{

SqlConnection cnn;

cnn = new SqlConnection("Data Source=112SALAS25;Initial Catalog=Ej1;User ID=sa;Password=sqladmin");

cnn.Open();

MessageBox.Show("Conectado");

return cnn;

}

## Llenar ComboBox:

public void llenarCombo(ComboBox cb)

{

try

{

SqlConnection c = Conexion.agregarConexion();

cmd = new SqlCommand("SELECT nomberIng FROM ingenierias", c);

dr = cmd.ExecuteReader();

while (dr.Read())

{

cb.Items.Add(dr["nomberIng"].ToString());

}

cb.SelectedIndex = 3;

dr.Close();

}

catch (Exception ex)

{

MessageBox.Show("Conexion no posible" + ex);

}

}

}

### Llenar comboBox2

public void llenarCombo(ComboBox cb)

{

try

{

SqlConnection c = Conexion.agregarConexion();

cmd = new SqlCommand("SELECT DISTINCT nombre FROM carrera", c);

dr = cmd.ExecuteReader();

while (dr.Read())

{

cb.Items.Add(dr["nombre"].ToString());

}

dr.Close();

cb.SelectedIndex = 0;

}

catch (Exception ex)

{

MessageBox.Show("No se puede conectar");

}

}

## Llenar Grid:

private void Button\_Click(object sender, RoutedEventArgs e)

{

SqlConnection con;

con = Conexion.agregarConexion();

SqlCommand cmm = new SqlCommand(String.Format("SELECT \* FROM aspirante WHERE ingenieria='"+cbIngenieria.Text.ToString()+"'"),con);

SqlDataReader lec = cmm.ExecuteReader();

gridInfo.ItemsSource = lec;

con.Close();

}

## Clase asistente

class Asistente

{

public int claveU { get; set; }

public String nombre { get; set; }

public String apellido { get; set; }

public String sexo {get; set; }

public String correo {get; set; }

public Asistente(int clave, String nombre, String apellido, String sexo, String correo)

{

this.claveU = clave;

this.nombre = nombre;

this.apellido = apellido;

this.sexo = sexo;

this.correo = correo;

}

public Asistente(String nombre)

{

this.nombre = nombre;

}

public Asistente(String nombre, String correo)

{

this.nombre = nombre;

this.correo = correo;

}

}

}

## AsistenteG

class AsistenteG

{

public static int agregarAsistente(Asistente a)

{

SqlConnection con;

con = Conexion.agregarConexion();

SqlCommand cmm = new SqlCommand(String.Format("INSERT INTO asistente (claveU, nombre, apellido, sexo, correo) values('{0}','{1}','{2}','{3}','{4}')", a.claveU, a.nombre, a.apellido, a.sexo, a.correo), con);

int res = cmm.ExecuteNonQuery();

con.Close();

return res;

}

}

## MainWindow

DAR CLICK EN DISEÑO

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

Conexion c = new Conexion();

c.llenarCombo(cbInge);

c.llenarCombo2(cbUni);

}

### Agregar

private void btAgregar\_Click(object sender, RoutedEventArgs e)

{

int clave = int.Parse(tbClave.Text);

Asistente a = new Asistente(clave, tbNombre.Text, tbApellido.Text, tbSexo.Text, tbCorreo.Text);

int res = AsistenteG.agregarAsistente(a);

if (res > 0){

MessageBox.Show("Agregado");

String carrera = cbInge.Text.ToString();

if (carrera.Equals("otro") || carrera.Equals("externo"))

{

MessageBox.Show("se les solicita un pago de $150.00 pesos y solo se les da el código de entrada (567890). ");

}

else{ //Es ingeniero

MessageBox.Show("código para recoger su playera (12345) y un código para entrar a la fiesta (56789)");

}

}

else{

MessageBox.Show("No agregado");

}

}

## Cambio entre ventanas

private void btReporte\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Reporte a = new Reporte();

a.Show();

}

## Layout

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="380\*"/>

<ColumnDefinition Width="137\*"/>

</Grid.ColumnDefinitions>

<Label Content="Ingenieria:" HorizontalAlignment="Left" Margin="37,158,0,0" VerticalAlignment="Top"/>

<Label Content="Correo:" HorizontalAlignment="Left" Margin="37,106,0,0" VerticalAlignment="Top"/>

<Label Content="Sexo:" HorizontalAlignment="Left" Margin="37,80,0,0" VerticalAlignment="Top"/>

<Label Content="Nombre:" HorizontalAlignment="Left" Margin="37,28,0,0" VerticalAlignment="Top"/>

<ComboBox x:Name="cbInge" HorizontalAlignment="Left" Margin="189,158,0,0" VerticalAlignment="Top" Width="120" Height="26"/>

<TextBox x:Name="tbCorreo" HorizontalAlignment="Left" Height="23" Margin="189,106,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="tbSexo" HorizontalAlignment="Left" Height="23" Margin="189,83,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="tbNombre" HorizontalAlignment="Left" Height="23" Margin="189,28,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="btAgregar" Content="Agregar" HorizontalAlignment="Left" Margin="37,242,0,0" VerticalAlignment="Top" Width="272" Click="btAgregar\_Click"/>

<Button x:Name="btReporte" Content="Reporte" HorizontalAlignment="Left" Margin="37,279,0,0" VerticalAlignment="Top" Width="272" Click="btReporte\_Click"/>

<Label Content="Universidad:&#xD;&#xA;" HorizontalAlignment="Left" Margin="37,189,0,0" VerticalAlignment="Top"/>

<ComboBox x:Name="cbUni" HorizontalAlignment="Left" Margin="189,189,0,0" VerticalAlignment="Top" Width="120" Height="26"/>

<Label Content="Apellido:" HorizontalAlignment="Left" Margin="37,54,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="tbApellido" HorizontalAlignment="Left" Height="23" Margin="189,55,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="ClaveU:" HorizontalAlignment="Left" Margin="37,7,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="tbClave" HorizontalAlignment="Left" Height="23" Margin="189,7,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

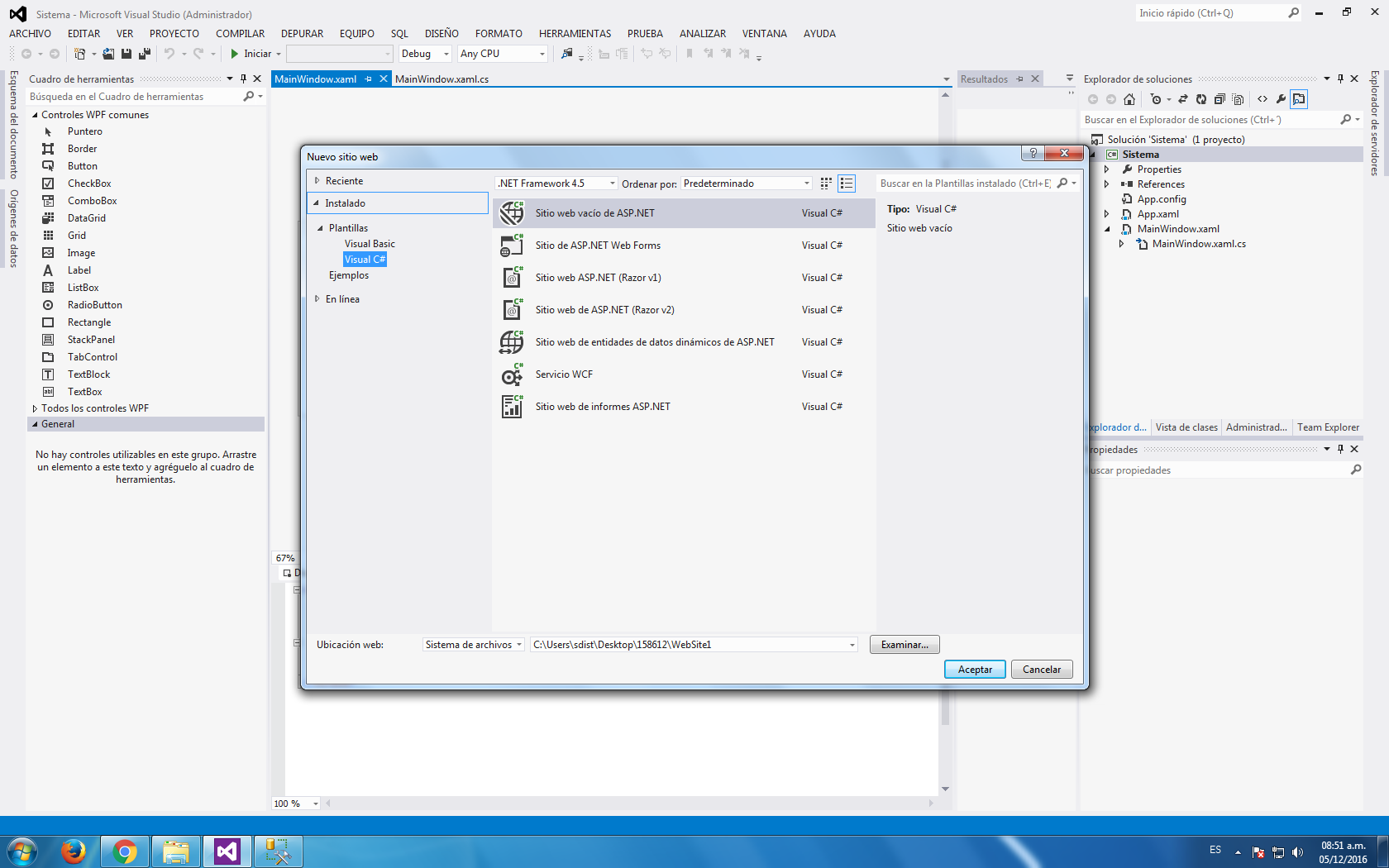
</Grid>

# 

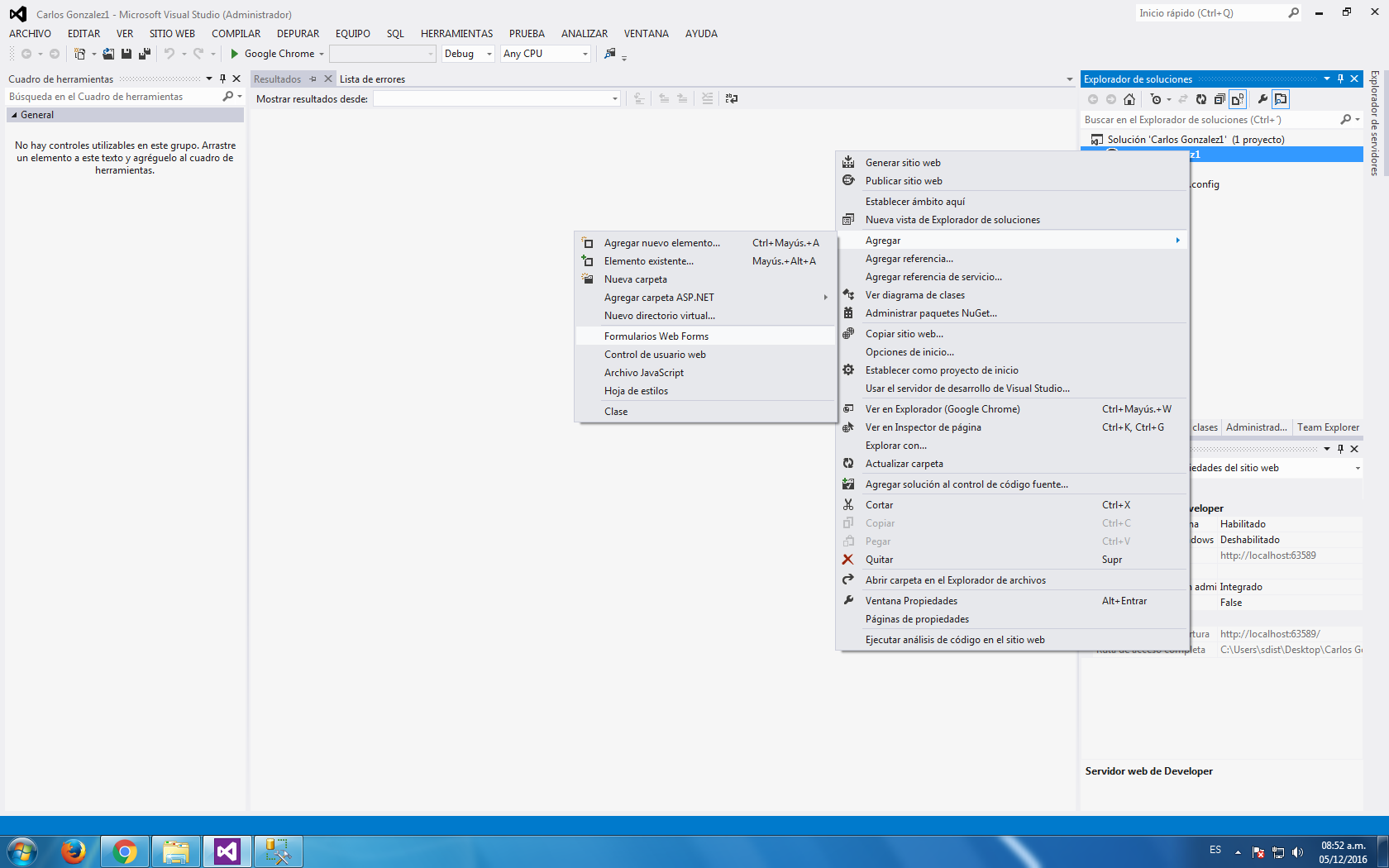
# 

# WEB

Nuevo Proyecto



Nueva Página



## 

## Conexión

protected OdbcConnection conectarBD()

{

String stringDeConexion = "Driver={SQL Server Native Client 11.0};Server=CC201-05;Uid=sa;Pwd=sqladmin;Database=Mensajes;";

try

{

OdbcConnection conexion = new OdbcConnection(stringDeConexion);

conexion.Open();

return conexion;

}

catch (Exception ex)

{

Label1.Text = "No se pudo hacer la conexion" + ex.StackTrace.ToString();

return null;

}

}

## Session

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

OdbcCommand sql = new OdbcCommand("SELECT nombre FROM usuario WHERE claveU='" + tbClave.Text + "' and passwd='" + tbContra.Text + "'", miConexion);

OdbcDataReader lec = sql.ExecuteReader();

if (lec.Read())

{

if (!lec["nombre"].ToString().Equals(""))

{

Session["Usuario"] = tbClave.Text;

Response.Redirect("Informacion.aspx");

}

}

else

{

Label1.Text = "ClaveU o contraseña incorrecta";

}

lec.Close();

miConexion.Close();

}

}

### 

### Fácil

protected void btSig\_Click(object sender, EventArgs e)

{

Session["Usuario"] = tbCarrera.Text;

if (!tbCarrera.Text.Equals(""))

{

Response.Redirect("Informacion.aspx");

}

else

{

Label1.Text = "Ingrese carrera";

}

}

}

## Modificar

protected void btModificar\_Click(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBaseDeDatos();

if (miConexion != null)

{

String correo = tbCorreo.Text;

**int edad = int.Parse(tbEdad.Text);**

String password = tbPass.Text;

OdbcCommand sql = new OdbcCommand(String.Format("UPDATE usuario SET email = '{0}', password = '{1}', edad = '{2}' WHERE claveU = '{3}'", correo, password, edad, DropDownList1.SelectedValue), miConexion);

int result = sql.ExecuteNonQuery();

if (result > 0)

{

lbMensaje.Text = "Modificación exitosa." + result.ToString();

}

else

{

lbMensaje.Text = "Error al modificar." + result.ToString();

}

}

}

## Agregar

### obtener información

#### Clave maxima

sql = new OdbcCommand("SELECT MAX(claveC) FROM carrera", miConexion);

Object o = sql.ExecuteScalar();

int claveC = (int)o + 1;

#### Clave WHERE

sql = new OdbcCommand("SELECT claveUni FROM universidad WHERE nombre='" + DropDownList2.SelectedItem.ToString() + "'", miConexion);

o = sql.ExecuteScalar();

int claveUni = (int)o;

protected void btEnvia\_Click(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

OdbcCommand sql = new OdbcCommand("SELECT MAX(claveM) FROM mensaje", miConexion);

Object o = sql.ExecuteScalar();

if (DBNull.Value.Equals(o))

{

//Es el primer dato y la llave primaria debe ser 1

String query = "INSERT INTO mensaje VALUES(1,'" + tbTitulo.Text + "','" + tbCont.Text + "','" + DateTime.Now.ToShortDateString() + "','" + DateTime.Now.ToShortDateString() + "'," + int.Parse(lbEmisor.Text) + "," + int.Parse(DropDownList2.Text) + ")";

sql = new OdbcCommand(query, miConexion);

int res = sql.ExecuteNonQuery();

lbNoticias.Text = res.ToString();

}

else

{

int llaveprimaria = (int)o;

llaveprimaria++;

String query = "INSERT INTO mensaje VALUES(" + llaveprimaria + ",'" + tbTitulo.Text + "','" + tbCont.Text + "'," + DateTime.Now.ToShortDateString() + ",'" + DateTime.Now.ToShortDateString() + "'," + int.Parse(lbEmisor.Text)+ "," + int.Parse(DropDownList2.Text) + ")";

sql = new OdbcCommand(query, miConexion);

int res = sql.ExecuteNonQuery();

lbNoticias.Text = res.ToString();

if (res > 0)

{

lbNoticias.Text = "Mensaje Enviado";

}

else

lbNoticias.Text = "No se pudo enviar el mensaje";

}

miConexion.Close();

}

}

### Agregar completo (3)

protected void btRegistrar\_Click(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

//Insertar 1

String query = "INSERT INTO asistente VALUES("+ int.Parse(tbClave.Text)+",'" + tbNombre.Text + "','" + tbApellido.Text + "','" + tbSexo.Text + "','" + tbCorreo.Text + "')";

OdbcCommand sql = new OdbcCommand(query, miConexion);

int res = sql.ExecuteNonQuery();

Label1.Text = res.ToString();

//Obtener información

sql = new OdbcCommand("SELECT MAX(claveC) FROM carrera", miConexion);

Object o = sql.ExecuteScalar();

int claveC = (int)o + 1;

//Insertar2

query = "INSERT INTO carrera VALUES("+claveC+", '" + DropDownList1.SelectedItem.ToString() + "', "+ int.Parse(tbClave.Text)+")";

sql = new OdbcCommand(query, miConexion);

int res2 = sql.ExecuteNonQuery();

res= res\*res2;

//Insertar3

sql = new OdbcCommand("SELECT claveUni FROM universidad WHERE nombre='" + DropDownList2.SelectedItem.Text + "'", miConexion);

o = sql.ExecuteScalar();

int claveUni = (int)o;

query = "INSERT INTO universidad\_asistente VALUES(" + int.Parse(tbClave.Text) + ", '" + claveUni + "')";

sql = new OdbcCommand(query, miConexion);

int res3 = sql.ExecuteNonQuery();

res = res \* res3;

if (res > 0)

{

Label1.Text = "Dado de alta exitosamente";

if (DropDownList1.SelectedItem.Text.Equals("externo") || DropDownList1.SelectedItem.Text.Equals("otro") || claveUni>1)

{

Label1.Text = "Paga" + claveUni;

}

else{

Label1.Text = "El codigo es 1";

}

}

else

Label1.Text = "No se pudo dar de alta";

}

miConexion.Close();

}

}

## Grid

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

OdbcCommand sql = new OdbcCommand("SELECT \* FROM vendedor ", miConexion);

OdbcDataReader lec = sql.ExecuteReader();

GridView1.DataSource = lec;

GridView1.DataBind();

}

else

Label1.Text = "no se pudo hacer el llenado del grid.";

### Grid 2

protected void Page\_Load(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

OdbcCommand sql = new OdbcCommand("SELECT \* FROM usuario", miConexion);

OdbcDataReader lec = sql.ExecuteReader();

GridView1.DataSource = lec;

GridView1.DataBind();

if (Session["Usuario"] != null)

{

OdbcCommand sql2 = new OdbcCommand("SELECT \* FROM usuario WHERE claveU=" + Session["Usuario"], miConexion);

lec = sql2.ExecuteReader();

if (lec.HasRows)

{

lec.Read();

lbInfoUsuario.Text = "ClaveU: " + lec.GetInt32(0) + "</br> Nombre: " + lec.GetString(1) + "</br> Correo: " + lec.GetString(2) + "</br> Contraseña: " + lec.GetString(3) + "</br> Edad: " + lec.GetInt32(4);

}

else

{

lbInfoUsuario.Text = "No hay usuarios";

}

}

else

{

lbInfoUsuario.Text = "No hay usuario ingresado";

}

}

else

{

lbNoticias.Text = "no se pudo hacer el llenado del grid.";

lbInfoUsuario.Text = "No hay usuarios";

}

}

Bien HECHO

EN el Page LOAD

//Try-catch para llenar un DropDown list

try{

OdbcConnection con;

//Segundo Paso

con = abrirConexion();

//OdbcCommand sql = new OdbcCommand(String.Format("SELECT \* FROM asistente JOIN carrera ON asistente.claveU = carrera.claveU WHERE carrera.nombre = '{0}';", Session["carr"]), con);

OdbcCommand sql = new OdbcCommand("Select nombre from universidad", con);

OdbcDataReader lector = sql.ExecuteReader();

while (lector.Read())

{

drop1.Items.Add(lector["nombre"].ToString());

}

//Cerramos lector y conexion.

lector.Close();

con.Close();

}catch(Exception ex){

}

## DropDown List

protected void Page\_Load(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

OdbcCommand sql = new OdbcCommand("SELECT claveU FROM usuario", miConexion);

OdbcDataReader lec = sql.ExecuteReader();

DropDownList2.Items.Clear();

while (lec.Read())

{

DropDownList2.Items.Add(lec.GetString(0));

}

lec.Close();

}

}

### Opción 2 DDList

protected void Page\_Load(object sender, EventArgs e)

{

OdbcConnection miConexion = conectarBD();

if (miConexion != null)

{

OdbcCommand sql = new OdbcCommand("SELECT nombre FROM dueno", miConexion);

try

{

OdbcDataReader lec = sql.ExecuteReader();

DropDownList2.Items.Clear();

while (lec.Read())

{

DropDownList2.Items.Add(lec.GetString(0));

}

lec.Close();

}

catch (Exception ex)

{

Label1.Text = "No se pudo hacer la conexion" + ex.StackTrace.ToString();

}

### Obtener texto de DDList

DropDownList1.SelectedItem.ToString()

int cRaza = (Int32) sql.ExecuteScalar();

Object o = sql.ExecuteScalar();

cRaza = (int)o;

## Moverse entre páginas

Response.Redirect("Sueldo.aspx");

## Layout

#### Registro comida ingenierías

Clave U<br />

<asp:TextBox ID="tbClave" runat="server"></asp:TextBox>

<br />

<br />

Nombre<br />

<asp:TextBox ID="tbNombre" runat="server"></asp:TextBox>

<br />

<br />

Apellido<br />

<asp:TextBox ID="tbApellido" runat="server"></asp:TextBox>

<br />

<br />

Sexo<br />

<asp:TextBox ID="tbSexo" runat="server"></asp:TextBox>

<br />

<br />

Correo<br />

<asp:TextBox ID="tbCorreo" runat="server"></asp:TextBox>

<br />

<br />

Ingenieria<br />

<asp:DropDownList ID="DropDownList1" runat="server">

</asp:DropDownList>

<br />

<br />

Universidades<br />

<asp:DropDownList ID="DropDownList2" runat="server">

</asp:DropDownList>

<br />

<br />

<asp:Button ID="btRegistrar" runat="server" Text="Button" />

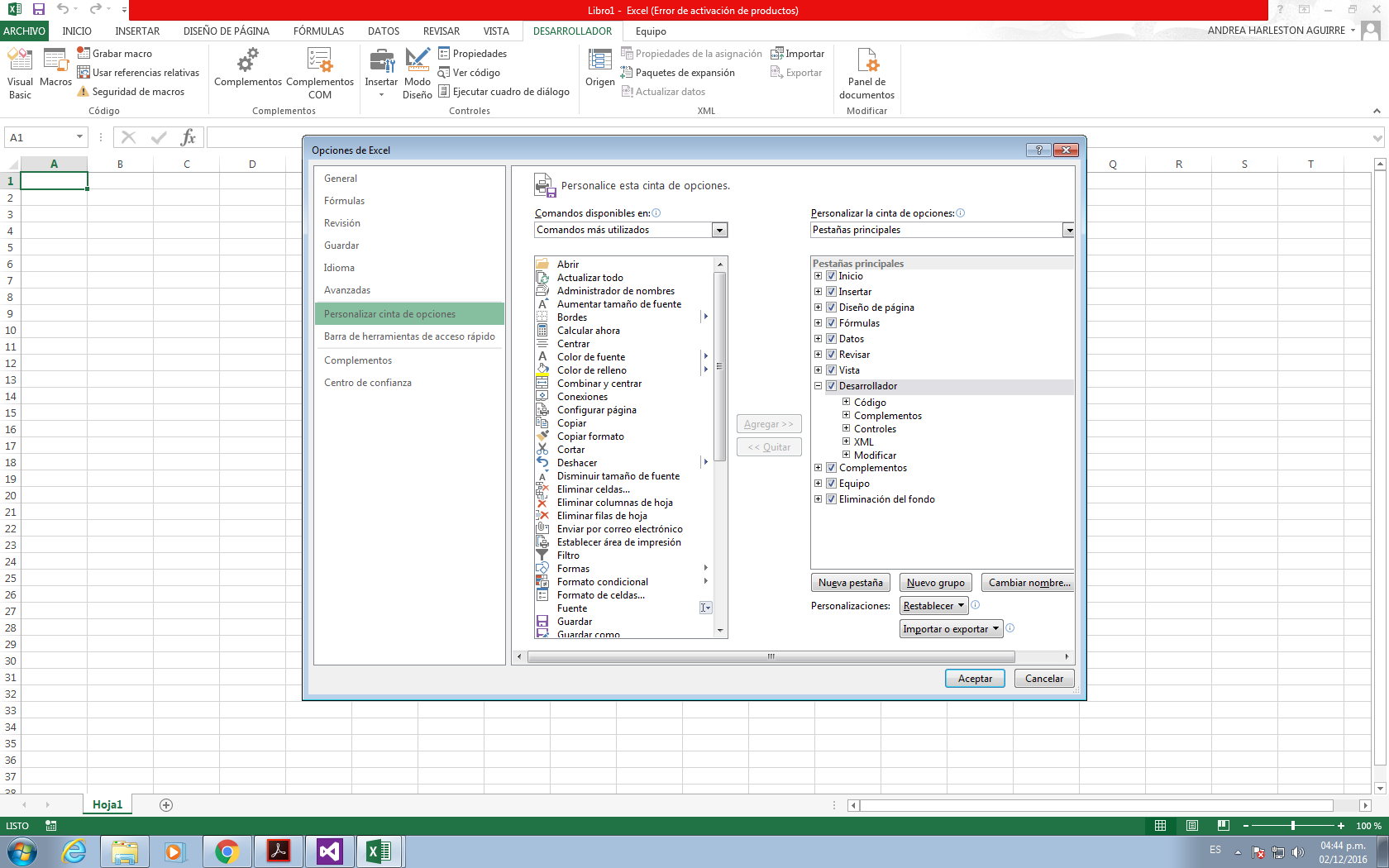
<br />

<br />

# 

# Excel

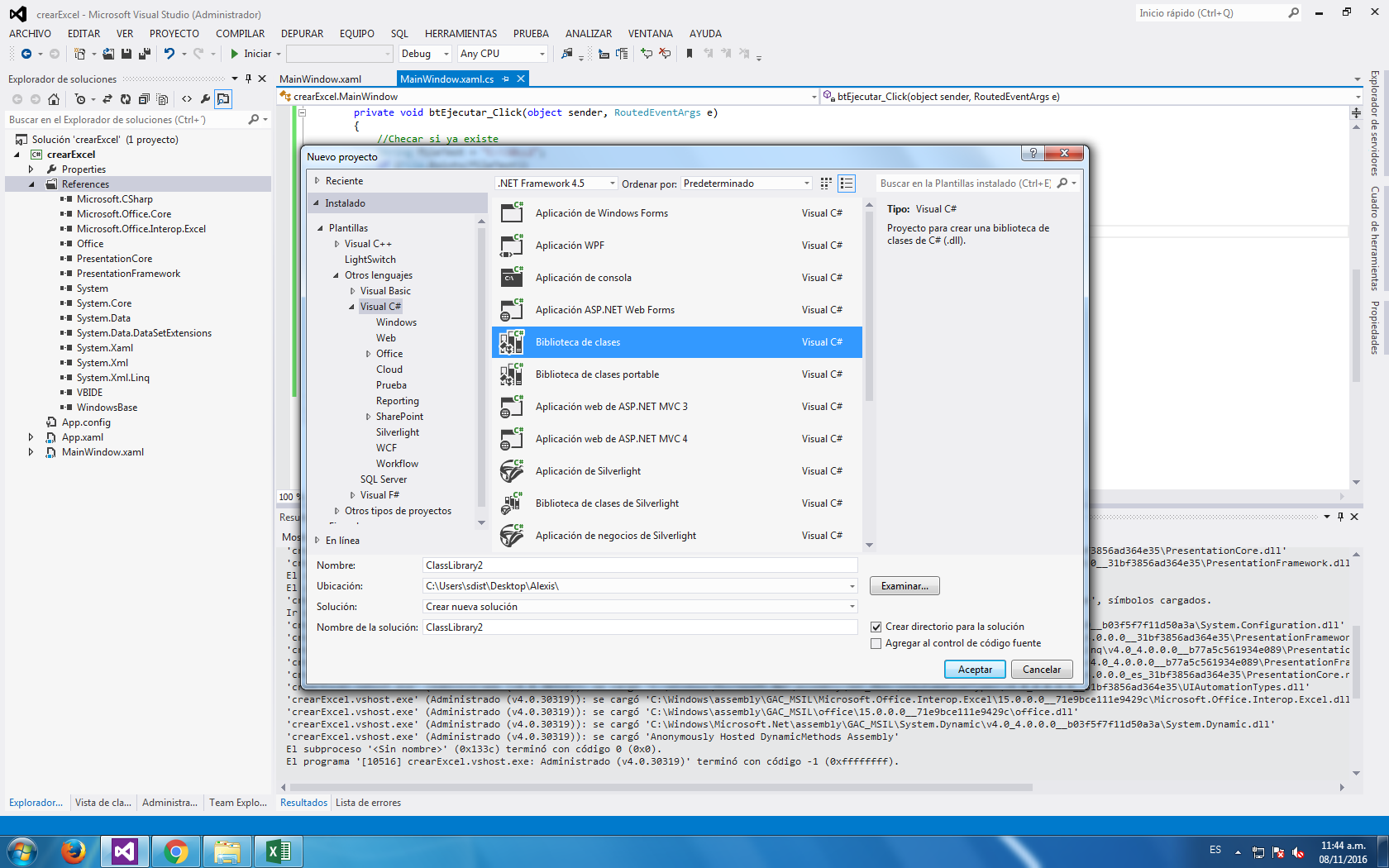
## Desarrollador

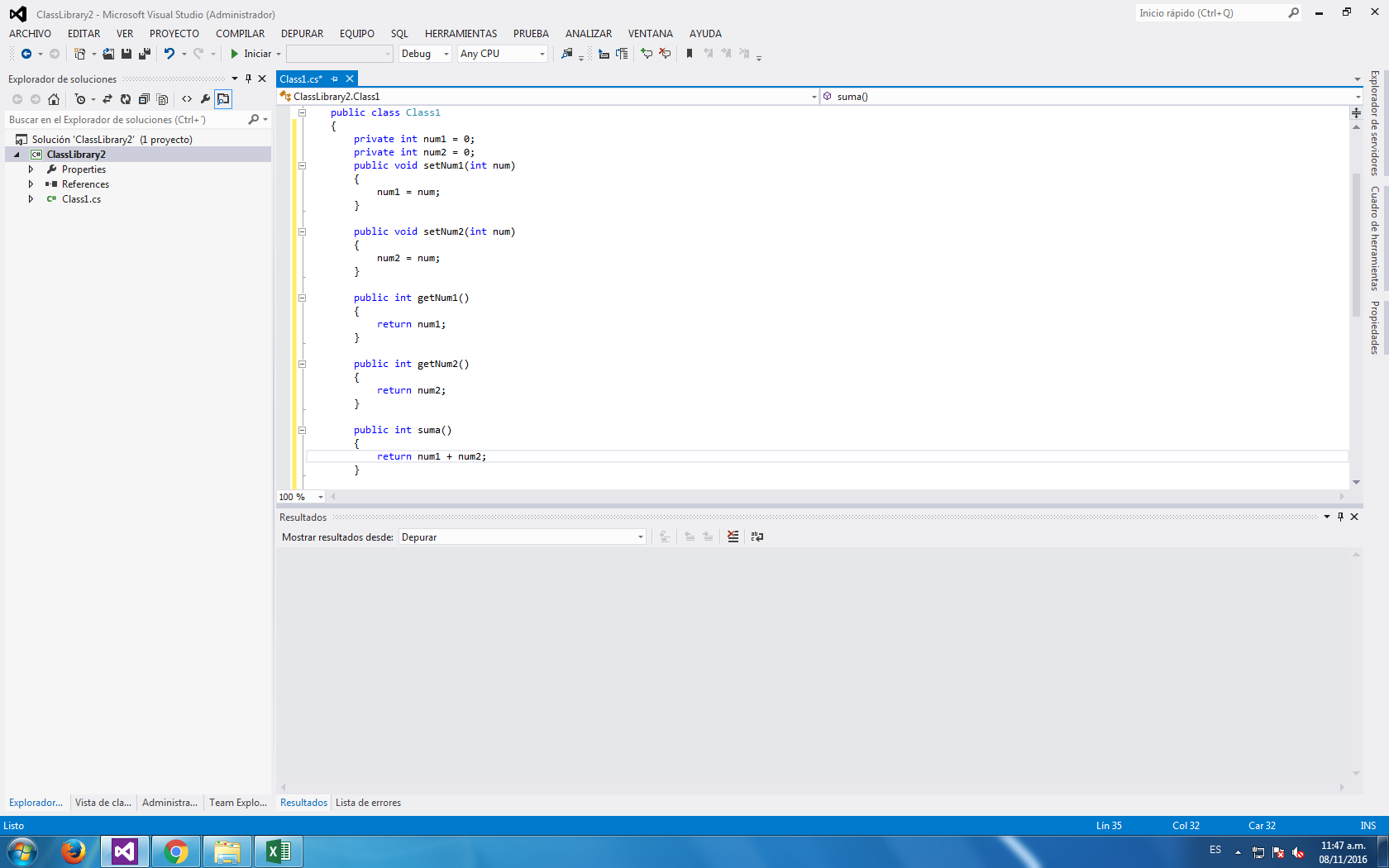


# DLL

## Visual studio

Class Library





public class Class1

{

private int num1 = 0;

private int num2 = 0;

public void setNum1(int num)

{

num1 = num;

}

public void setNum2(int num)

{

num2 = num;

}

public int getNum1()

{

return num1;

}

public int getNum2()

{

return num2;

}

public int cT()

{

int res;

if(num1== num2){

res = 0;

}

else{

if(num1>num2){

res= 1;

}

else{

res= 0;

}

return res;

}

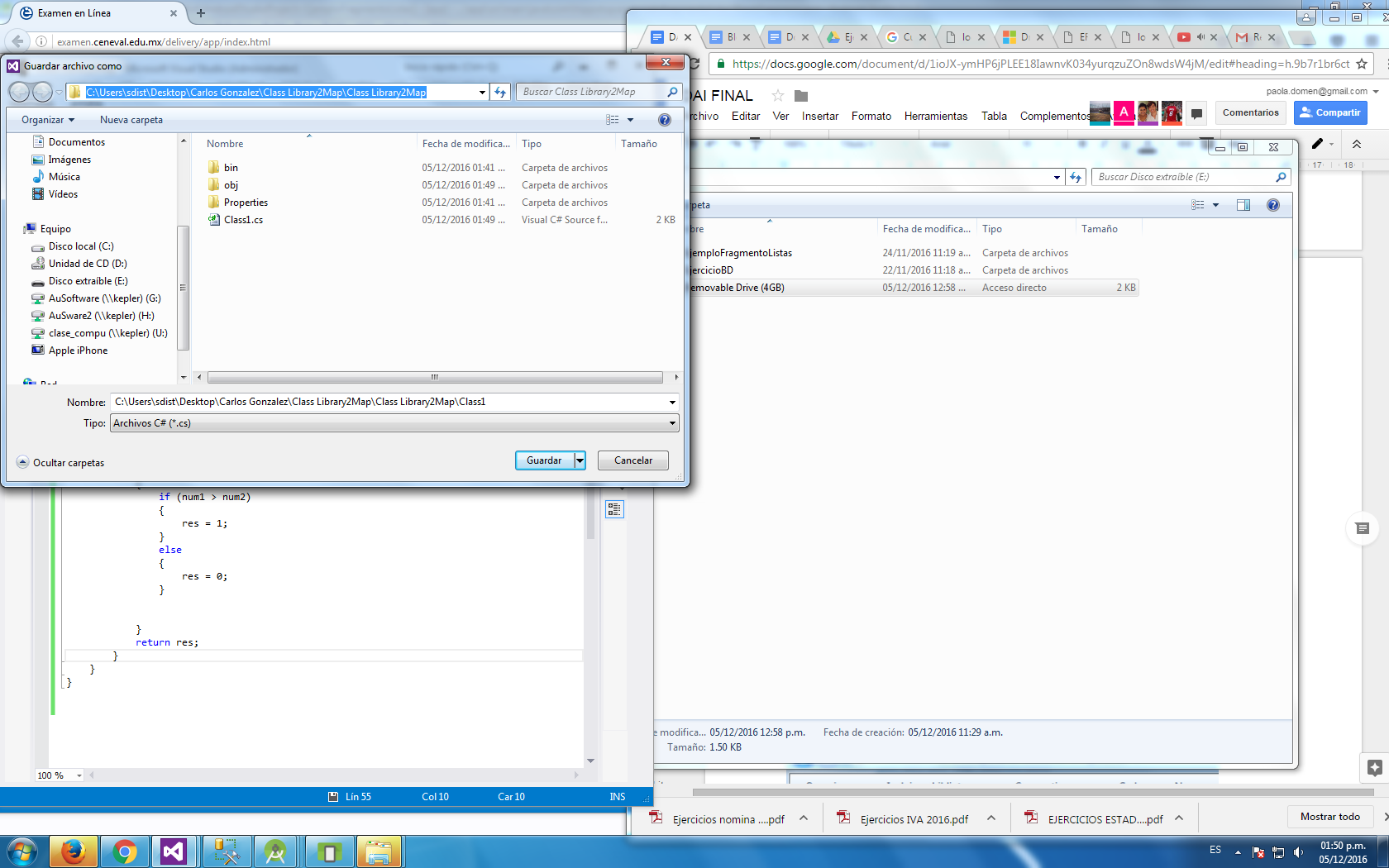
}

# 

# 

# 

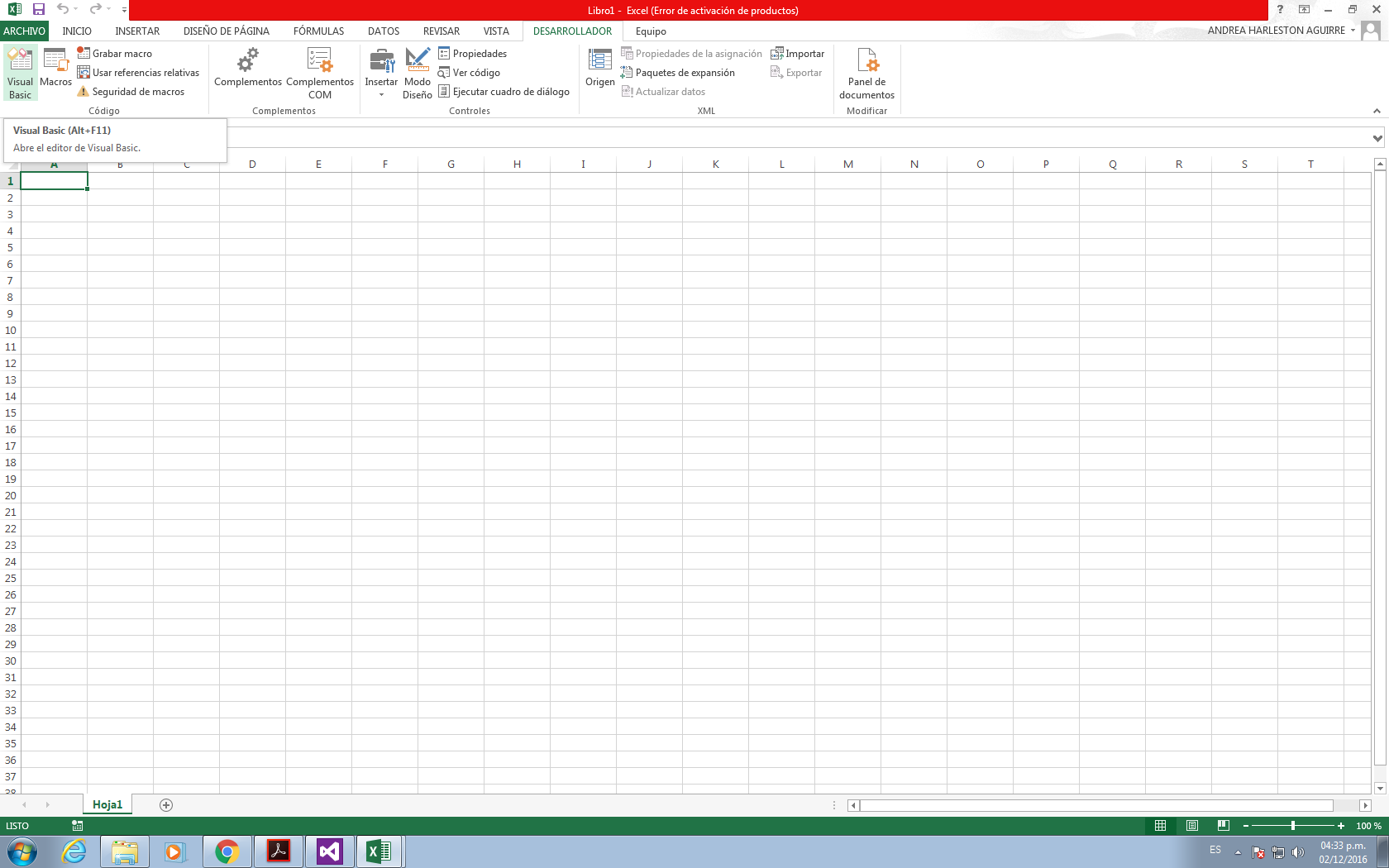
# 

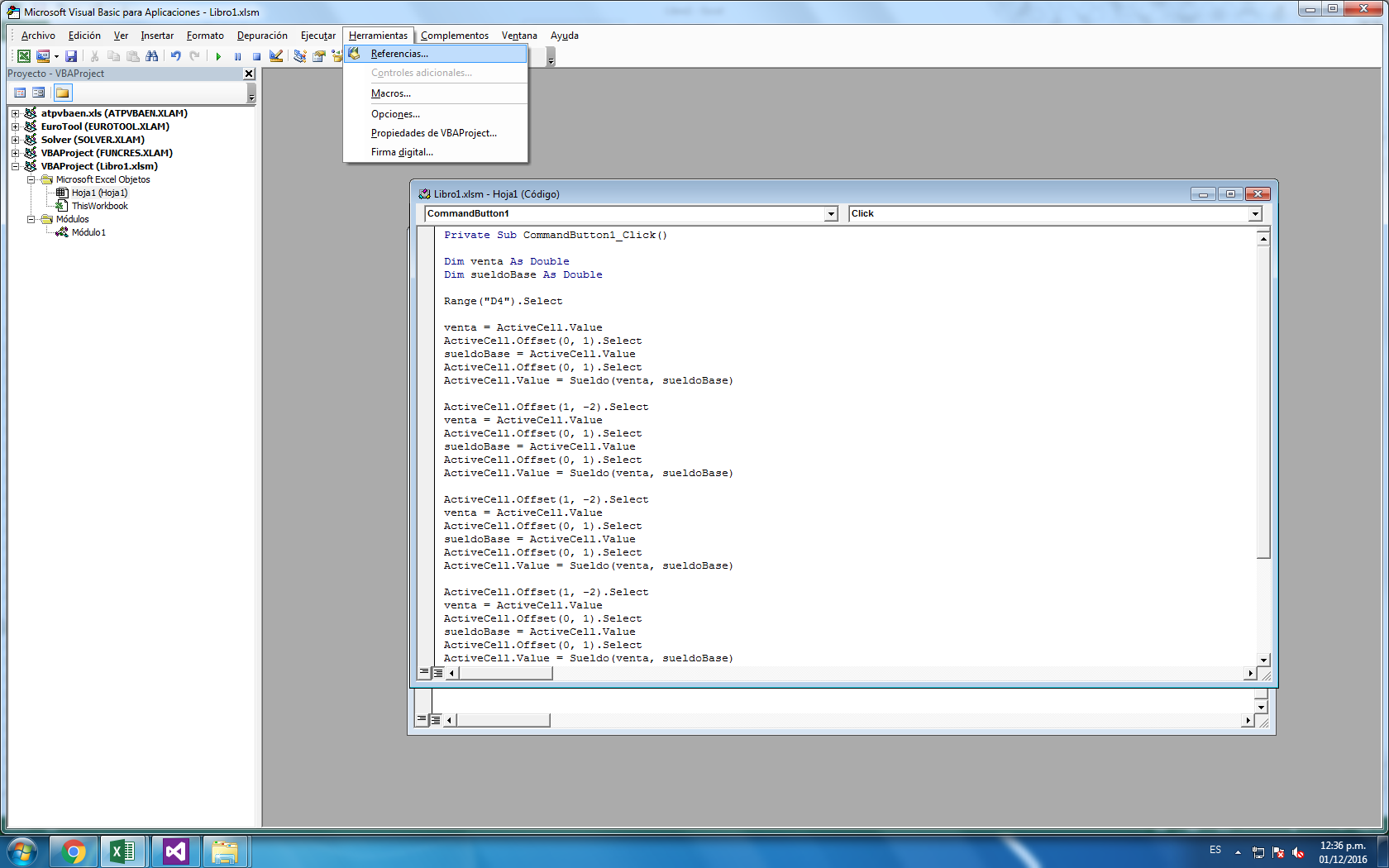


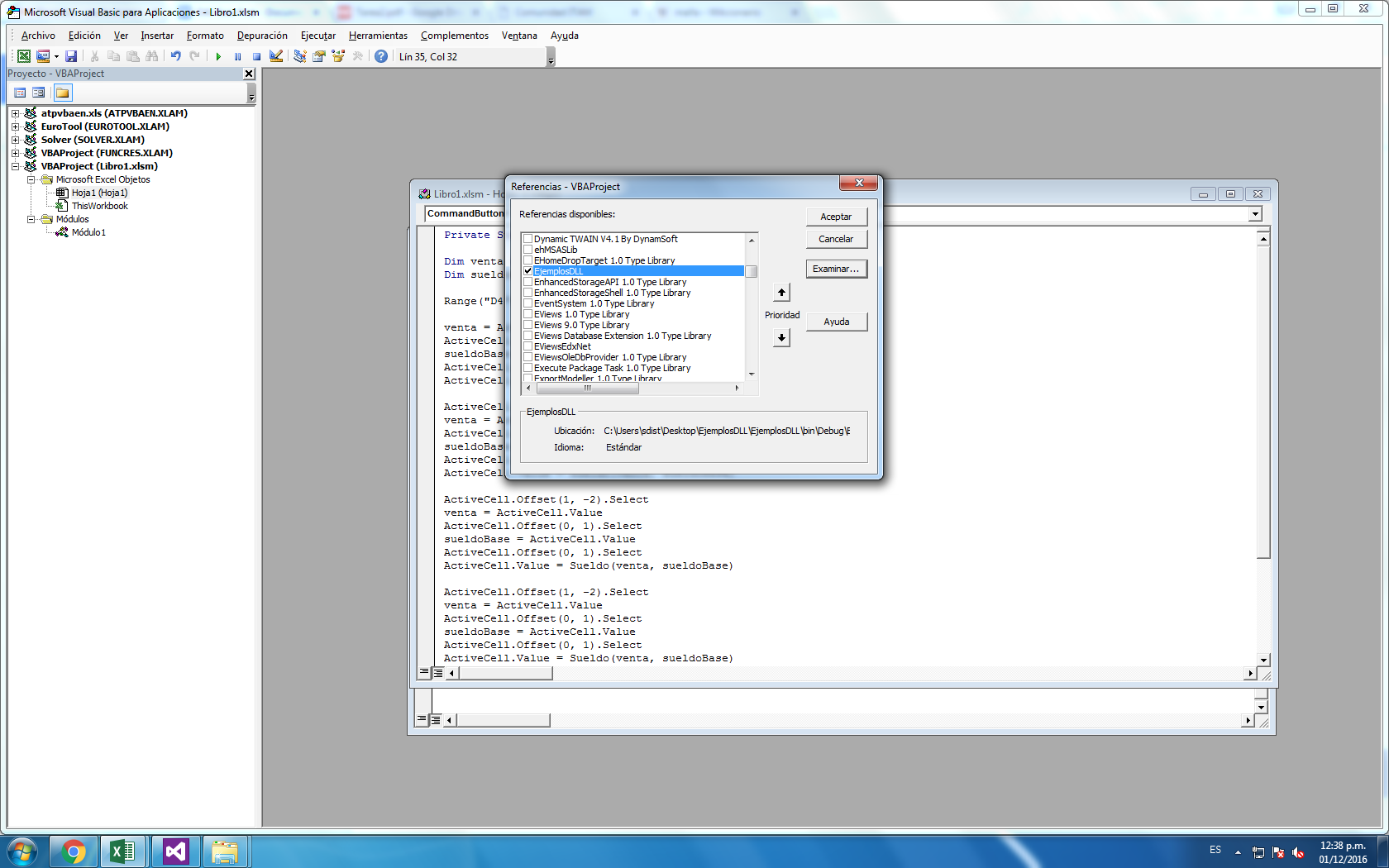
# 

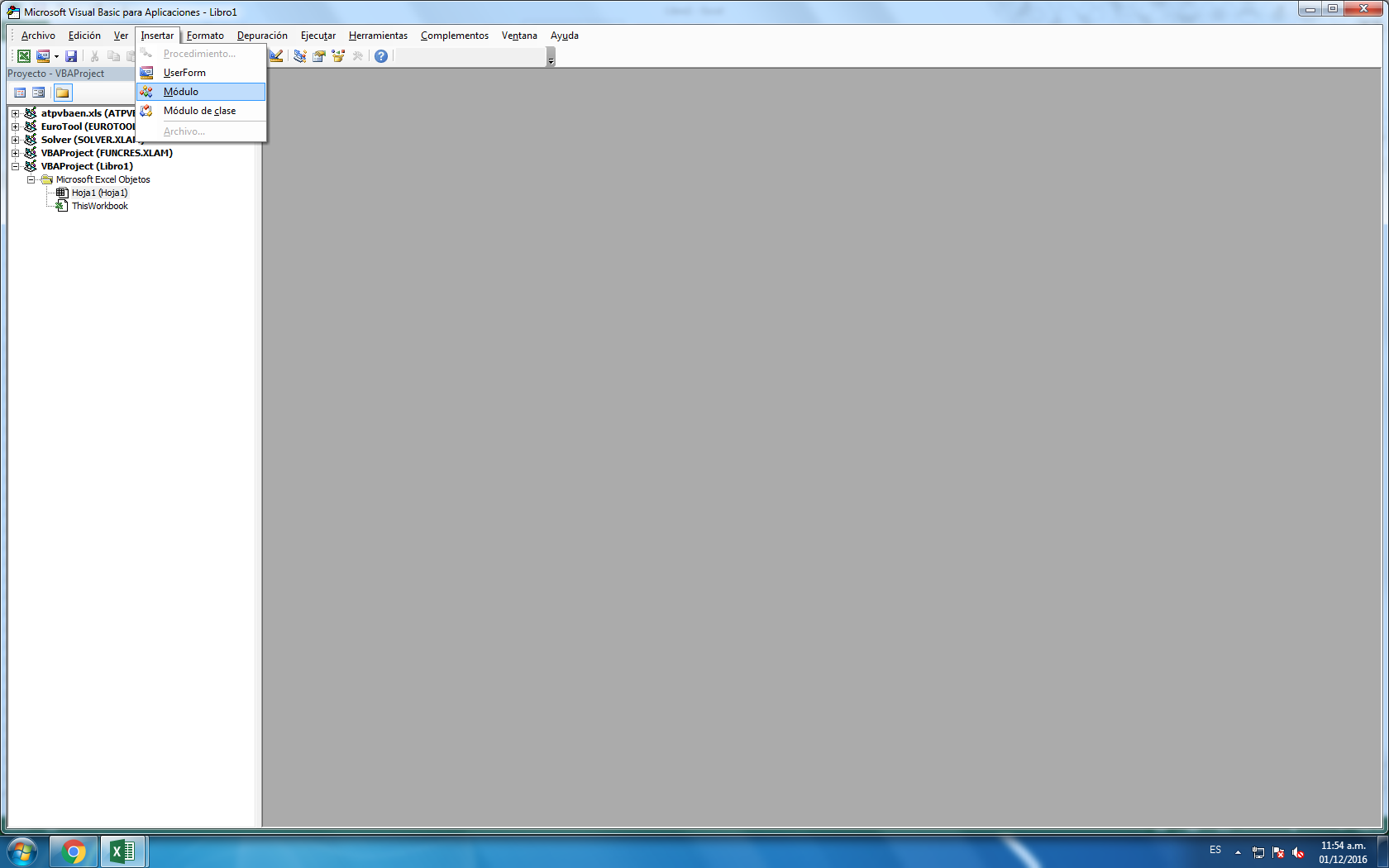
# 

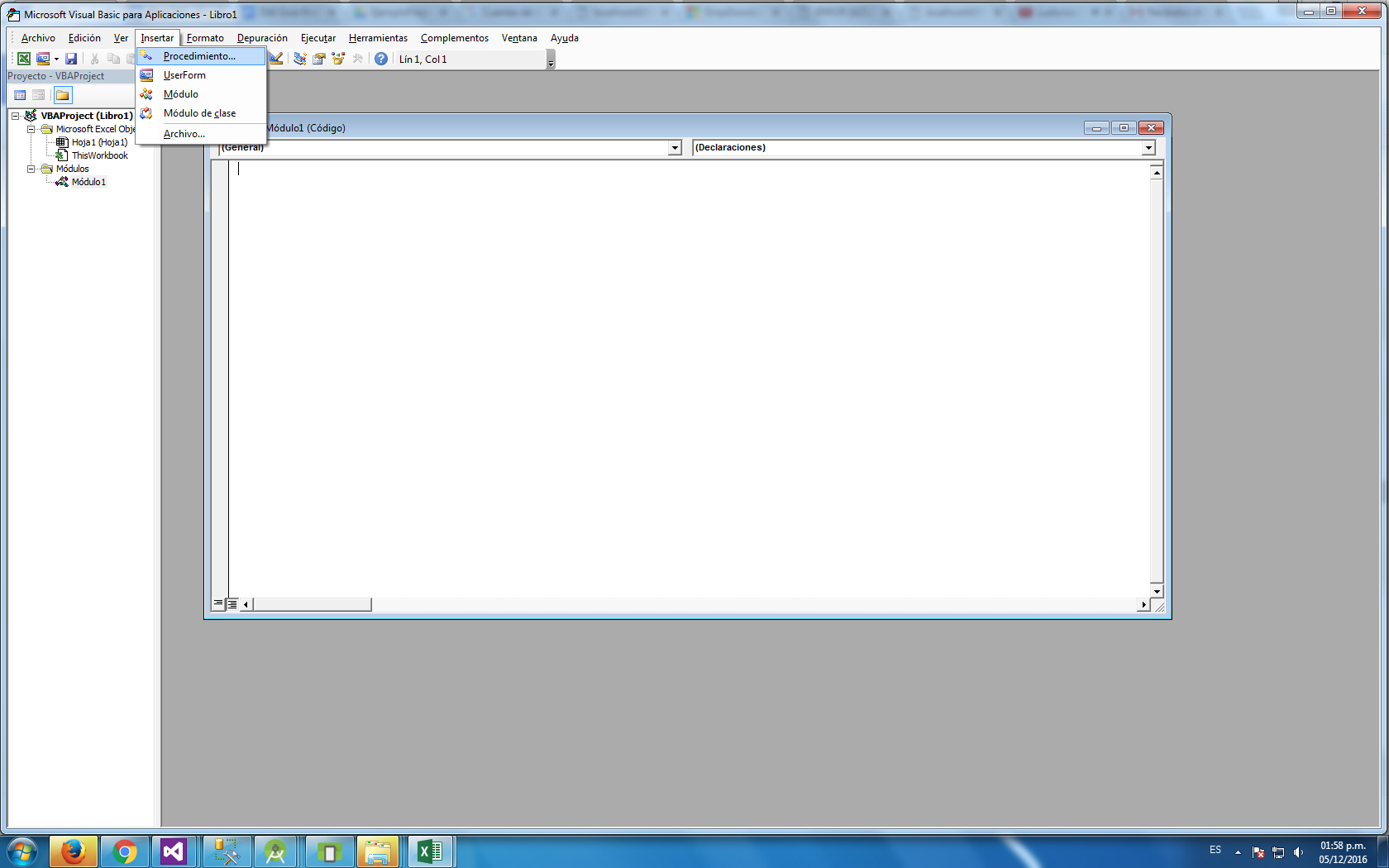
Pegar en System 32

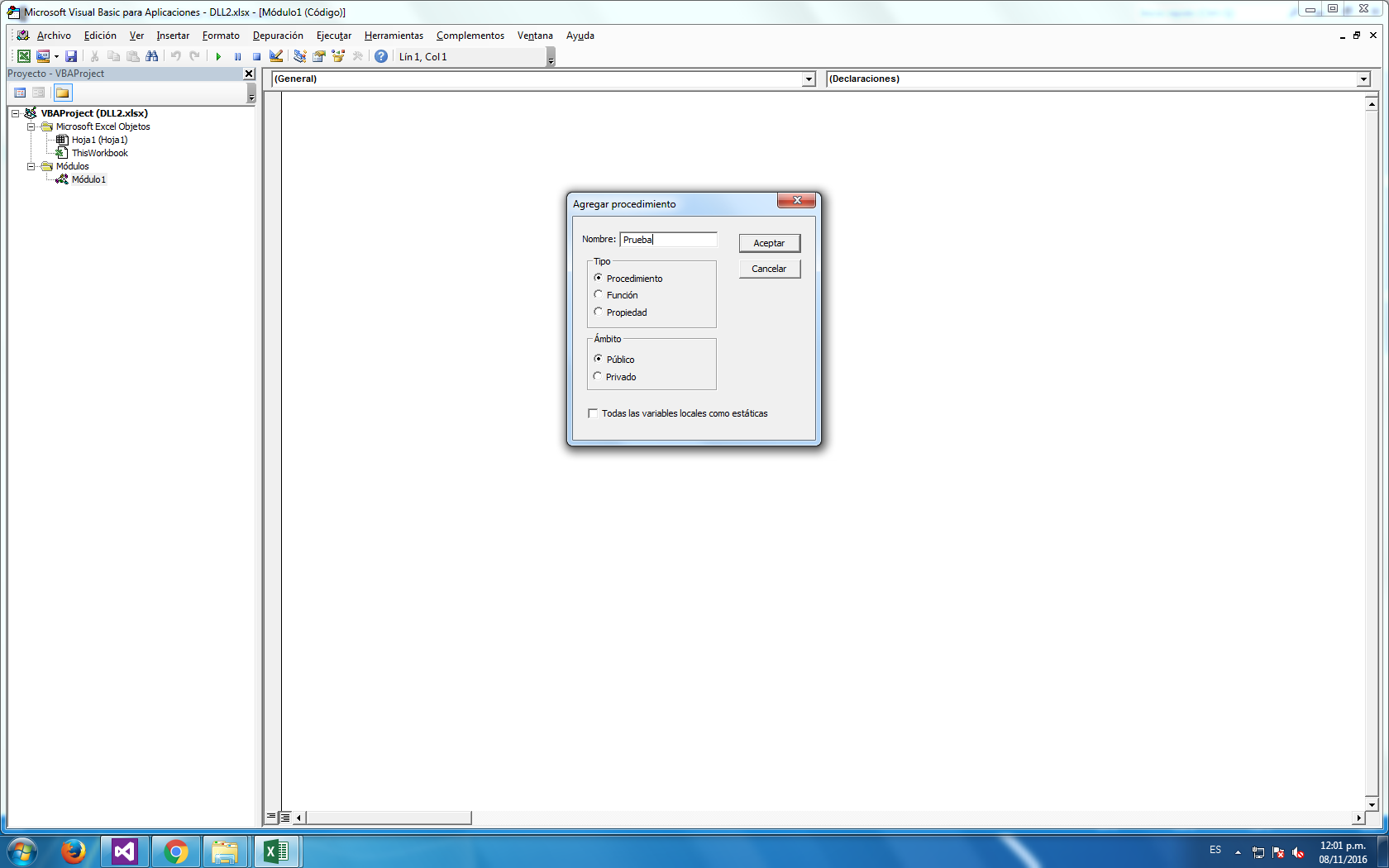


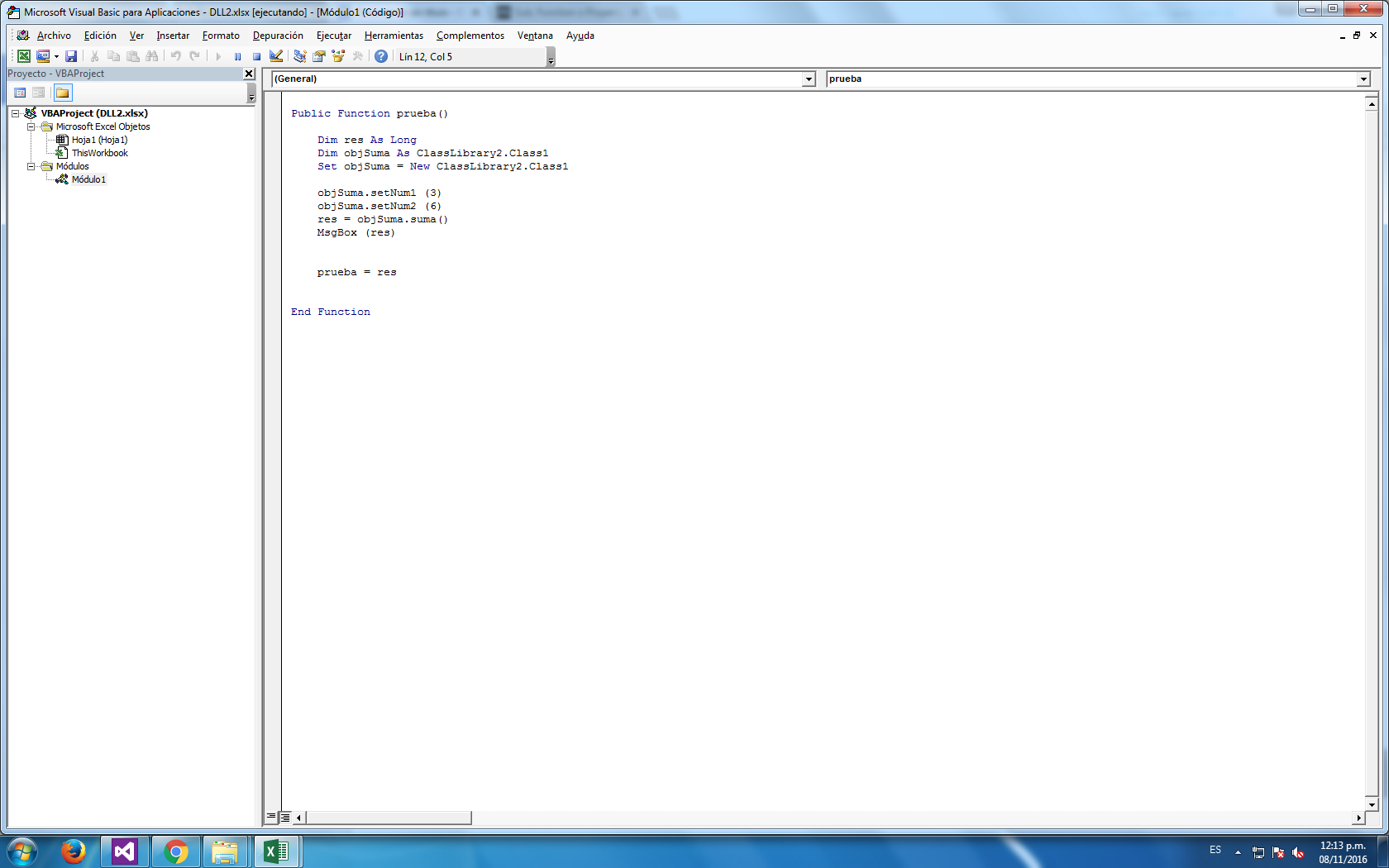












### código visual basic

Public Function prueba()

Dim n As Integer

Dim n1 As Integer

Dim objeto As EjemploDLL.Class1

Set objn = New EjemploDLL.Class1

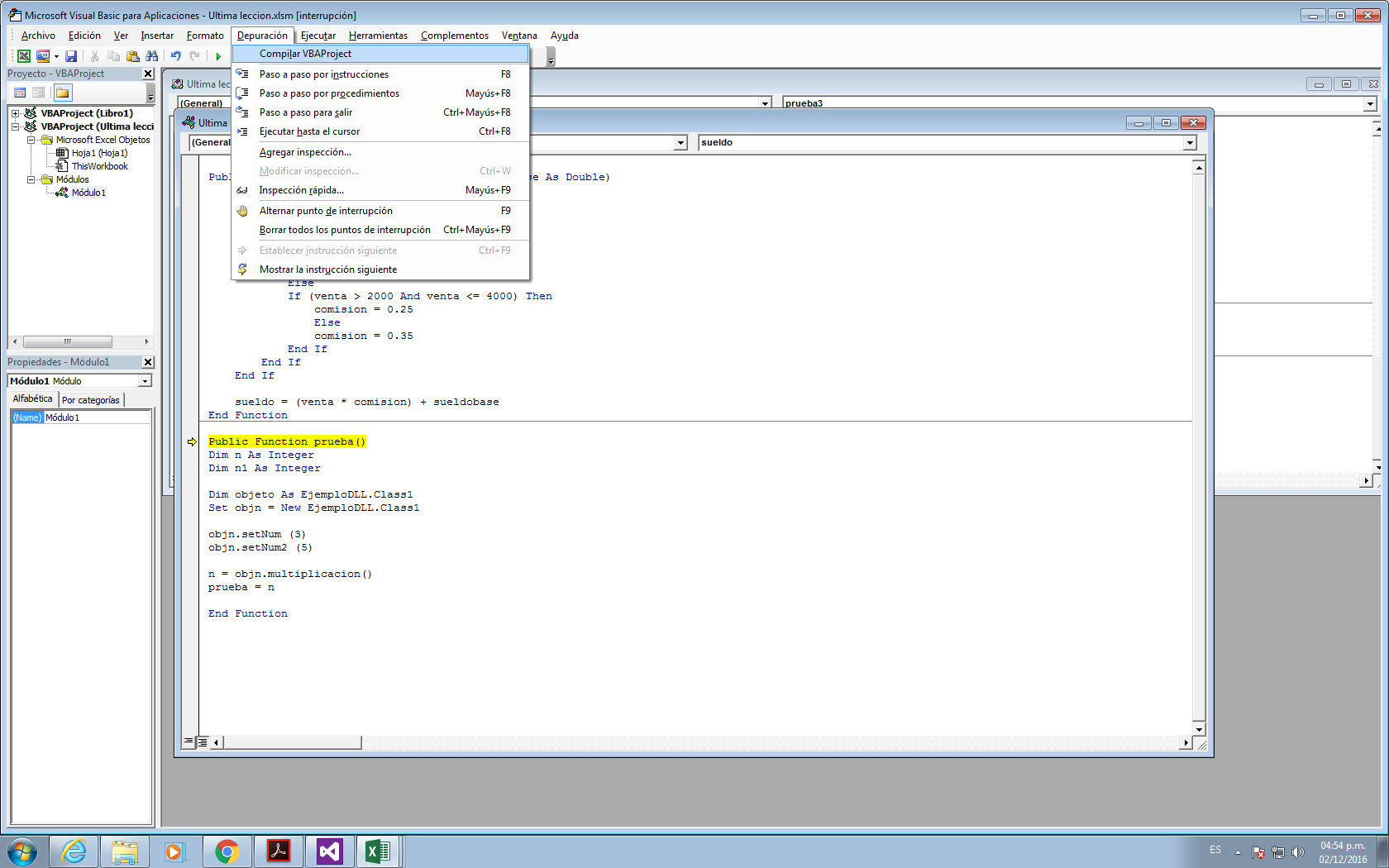
objn.setNum (3)

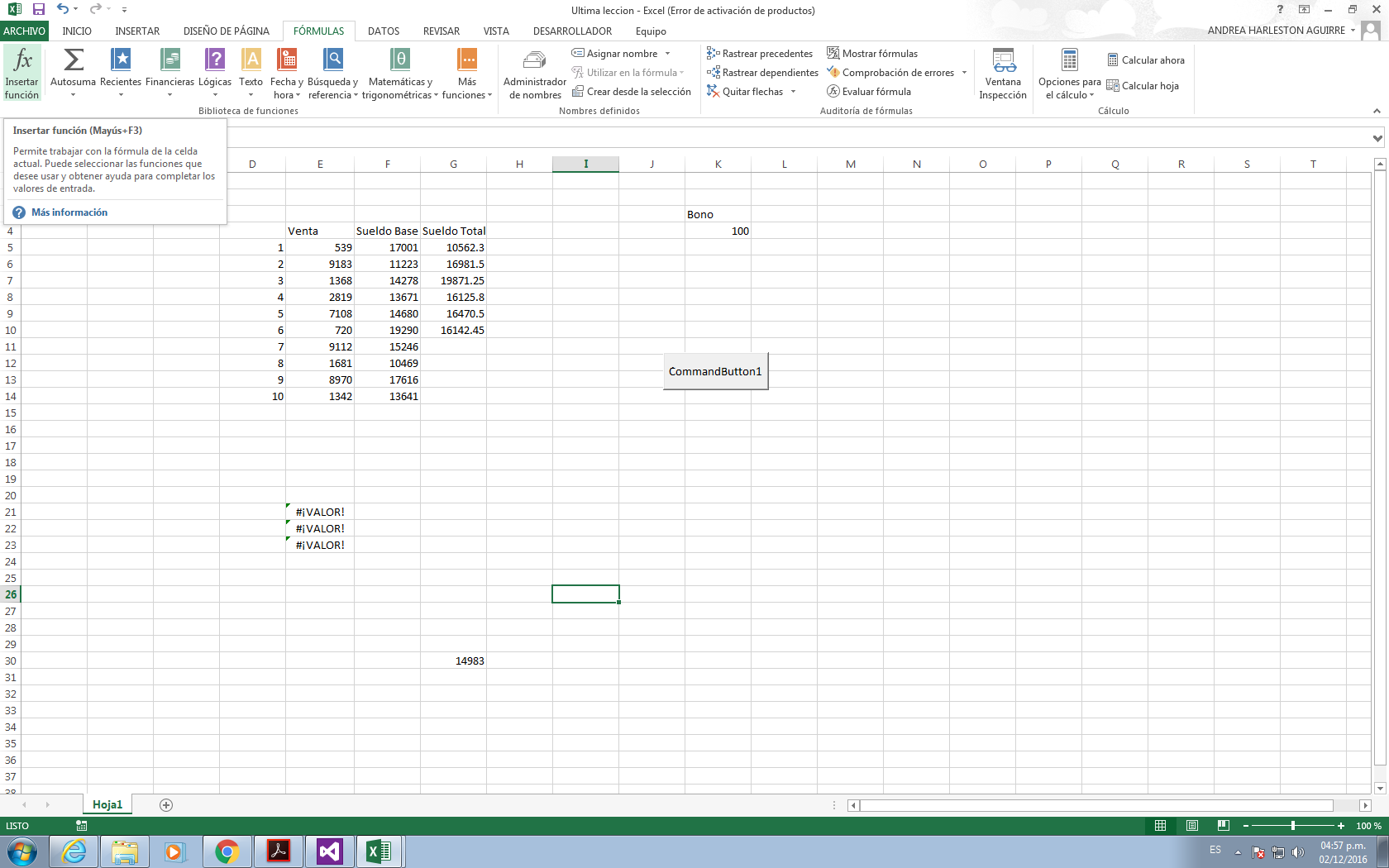
objn.setNum2 (5)

n = objn.multiplicacion()

prueba = n

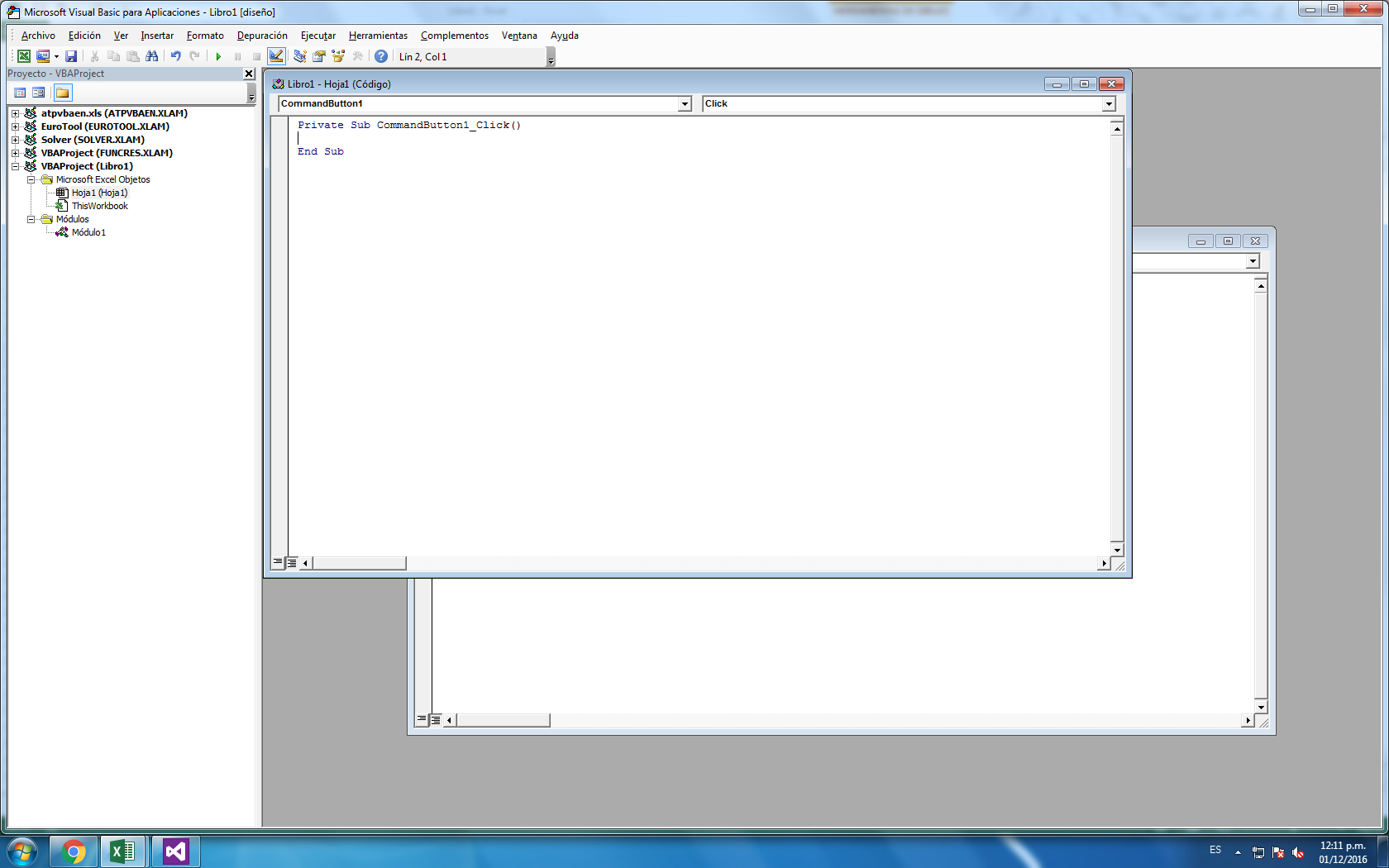
End Function

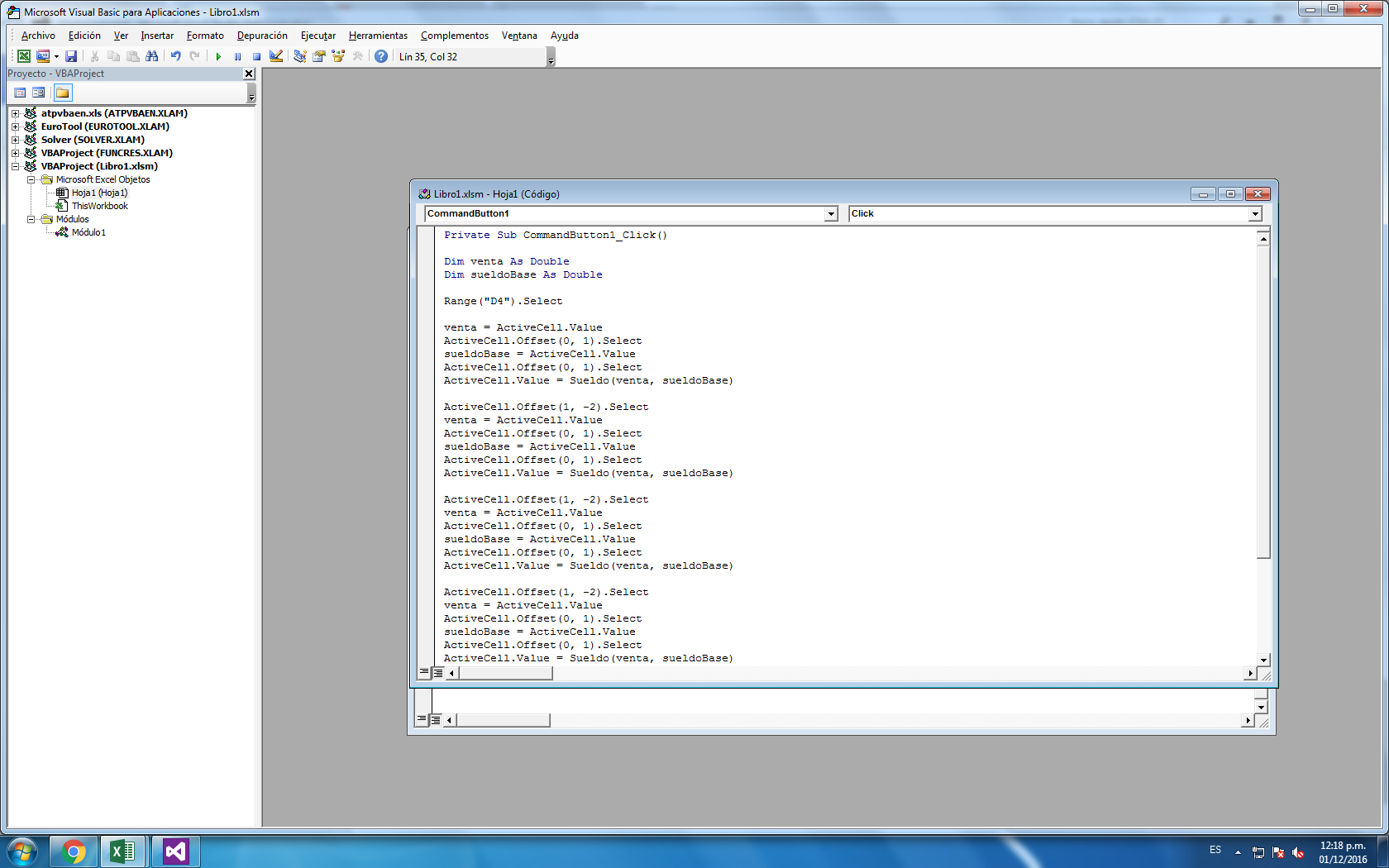




### 

1. Formulario, graba macros
2. Como código





Selección

Private Sub CommandButton1\_Click()

Dim num1 As Double

Dim num2 As Double

Range("D4").Select

num1 = ActiveCell.Value

ActiveCell.Offset(1, 0).Select

num2 = ActiveCell.Value

ActiveCell.Offset(1, 1).Select

ActiveCell.Value = num1

End Sub

# 

# 

# Android

## general

### conexión

**import** android.content.Context;

**import** android.database.sqlite.SQLiteDatabase;

**import** android.database.sqlite.SQLiteOpenHelper;

**public class** ConexionBD **extends** SQLiteOpenHelper{

String **cadenaCreate** = **"CREATE TABLE if not exists tablaprueba(\_id integer primary autoincrement, datos text not null);"**;

**public** ConexionBD(Context context) {

**super**(context, **"prueba.db"**, **null**, 1);

}

**public void** onCreate(SQLiteDatabase db) {

db.execSQL(**cadenaCreate**);

}

**public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {

String cadenaUpdate = **" DROP TABLE if exists tablaprueba;"**;

db.execSQL(cadenaUpdate);

onCreate(db);

}

}

### InterfazBD

**import** android.content.ContentValues;

**import** android.content.Context;

**import** android.database.Cursor;

**import** android.database.sqlite.SQLiteDatabase;

**import** android.database.sqlite.SQLiteException;

**public class** InterfazBD {

ConexionBD **con**;

SQLiteDatabase **db**;

**public** InterfazBD(Context context) {

**con** = **new** ConexionBD(context);

}

**public void** open() **throws** SQLiteException{

**db**=**con**.getWritableDatabase();

}

**public void** close() **throws** SQLiteException{

**con**.close();

}

**public long** insertarDatos(String dato) {

ContentValues valores;

open();

valores = **new** ContentValues();

valores.put(**"datos"**,dato);

**long** clave = **db**.insert(**"tablaprueba"**,**null**,valores);

close();

**return** clave;

}

**public void** insertarDatosPrueba() {

ContentValues valores;

open();

valores = **new** ContentValues();

valores.put(**"datos"**,**"Carlos"**);

**db**.insert(**"tablaprueba"**,**null**,valores);

valores = **new** ContentValues();

valores.put(**"datos"**,**"Alfredo"**);

**db**.insert(**"tablaprueba"**,**null**,valores);

valores = **new** ContentValues();

valores.put(**"datos"**,**"Gonzalez"**);

**db**.insert(**"tablaprueba"**,**null**,valores);

valores = **new** ContentValues();

valores.put(**"datos"**,**"ITAM"**);

**db**.insert(**"tablaprueba"**,**null**,valores);

close();

}

**public** Cursor traerDatos() {

Cursor res = **null**;

open();

String cadena = **"SELECT \* FROM tablaprueba;"**;

res = **db**.rawQuery(cadena,**null**);

**if**(res.getCount()==0) {

insertarDatosPrueba();

res=**db**.rawQuery(cadena,**null**);

}

**return** res;

}

}

## Bases de datos

### AdminSQLiteOpenHelper

Crea base de datos y tablas. Mi base de datos

**import** android.content.Context;

**import** android.database.sqlite.SQLiteDatabase;

**import** android.database.sqlite.SQLiteOpenHelper;

**public class** AdminSQLiteOpenHelper **extends** SQLiteOpenHelper {

**public** AdminSQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, **int** version) {

**super**(context, name, factory, version);

}

@Override

**public void** onCreate(SQLiteDatabase db) {

db.execSQL(**"CREATE TABLE alumnos (cu integer primary key, nombre text, carrera text, universidad text)"**);

}

@Override

**public void** onUpgrade(SQLiteDatabase db, **int** i, **int** i1) {

db.execSQL(**"drop table if exists alumnos"**);

db.execSQL(**"CREATE TABLE alumnos (cu integer primary key, nombre text, carrera text, universidad text)"**);

}

}

### Opción 2 (Ambas funcionan)

**import** android.content.Context;

**import** android.database.sqlite.SQLiteDatabase;

**import** android.database.sqlite.SQLiteOpenHelper;

**public class** MiniDBMS **extends** SQLiteOpenHelper {

*//crear la base de datos*

**public** MiniDBMS(Context context){

*//el 1 es el numero de version*

**super**(context, **"chafimon.db"**, **null**,1);

}

*//solo se ejecuta si la tabla no existe en el dispositivo*

@Override

**public void** onCreate(SQLiteDatabase db) {

String strCreateDB = **"create table pokemon(id integer primary key, "** +

**"nombre text, foto text, descripcion text);"**;

db.execSQL(strCreateDB);

}

@Override

**public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {

String strUpdate= **"drop table if exists pokemon;"**;

db.execSQL(strUpdate);

onCreate(db);

}

}

## Main Activity

**public class** MainActivity **extends** AppCompatActivity {

#### Calculadora

*//atributos*

**private** EditText **oper1**, **oper2**;

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

**oper1**=(EditText)findViewById(R.id.***editText***);

**oper2**=(EditText)findViewById(R.id.***editText2***);

}

**public void** sumar(View v){

**int** num1, num2, res;

num1=Integer.*parseInt*(**oper1**.getText().toString());

num2=Integer.*parseInt*(**oper2**.getText().toString());

res=num1+num2;

mostrar(res);

}

#### Login

**public class** MainActivity **extends** AppCompatActivity {

**private** EditText **et1**, **et2**, **et3**, **et4**;

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

**et1** = (EditText)findViewById(R.id.***claveU***);

**et2** = (EditText)findViewById(R.id.***editText***);

**et3** = (EditText)findViewById(R.id.***editText2***);

**et4** = (EditText)findViewById(R.id.***editText3***);

}

### Alta

**public void** alta(View v) {

AdminSQLiteOpenHelper admin = **new** AdminSQLiteOpenHelper(**this**, **"administracion"**, **null**, 1);

SQLiteDatabase db = admin.getWritableDatabase();

String cu = **et1**.getText().toString();

String nombre = **et2**.getText().toString();

String carrera = **et3**.getText().toString();

String universidad = **et4**.getText().toString();

ContentValues registro = **new** ContentValues();

registro.put(**"cu"**,cu);

registro.put(**"nombre"**,nombre);

registro.put(**"carrera"**, carrera);

registro.put(**"universidad"**,universidad);

db.insert(**"alumnos"**,**null**,registro);

db.close();

**et1**.setText(**""**);

**et2**.setText(**""**);

**et3**.setText(**""**);

**et4**.setText(**""**);

Toast.*makeText*(**this**, **"Se cargaron los datos adecuadamente"**, Toast.***LENGTH\_SHORT***).show();

}

#### Alta simple

*// Agregar pokemon*

**public void** inserta( PokeItem pokeItem) {

SQLiteDatabase db = **this**.getWritableDatabase();

*//que funcion tiene ContentValues?*

ContentValues values = **new** ContentValues();

values.put(**"nombre"**, pokeItem.getNombre());

values.put(**"descripcion"**, pokeItem.getDesc());

values.put(**"foto"**, pokeItem.getFoto());

*// inserta los campos values en la tabla pokemon, regresa la llave primaria*

db.insert(**"pokemon"**, **null**, values);

db.close();

}

### Consulta

**public void** consulta (View v) {

AdminSQLiteOpenHelper admin = **new** AdminSQLiteOpenHelper(**this**, **"administracion"**, **null**, 1);

SQLiteDatabase db = admin.getWritableDatabase();

String cu = **et1**.getText().toString();

Cursor file = db.rawQuery(**"select nombre, carrera,"**+**"universidad FROM alumnos WHERE cu="**+cu,**null**);

**if**(file.moveToFirst()){

**et2**.setText(file.getString(0));

**et3**.setText(file.getString(1));

**et4**.setText(file.getString(2));

}

**else** {

Toast.*makeText*(**this**, **"No existe este CU"**, Toast.***LENGTH\_SHORT***).show();

}

db.close();

}

### Baja

**public void** baja (View v) {

AdminSQLiteOpenHelper admin = **new** AdminSQLiteOpenHelper(**this**, **"administracion"**, **null**, 1);

SQLiteDatabase db = admin.getWritableDatabase();

String cu = **et1**.getText().toString();

**int** cant = db.delete(**"alumnos"**, **"cu="**+cu, **null**);

db.close();

**et1**.setText(**""**);

**et2**.setText(**""**);

**et3**.setText(**""**);

**et4**.setText(**""**);

**if**(cant == 1){

Toast.*makeText*(**this**, **"Se borro el alumno"**, Toast.***LENGTH\_SHORT***).show();

}

**else**

Toast.*makeText*(**this**, **"No existe el alumna"**, Toast.***LENGTH\_SHORT***).show();

}

### Modificar

**public void** modificar (View v) {

AdminSQLiteOpenHelper admin = **new** AdminSQLiteOpenHelper(**this**, **"administracion"**, **null**, 1);

SQLiteDatabase db = admin.getWritableDatabase();

String cu = **et1**.getText().toString();

String nombre = **et2**.getText().toString();

String carrera = **et3**.getText().toString();

String universidad = **et4**.getText().toString();

ContentValues registro = **new** ContentValues();

registro.put(**"cu"**,cu);

registro.put(**"nombre"**,nombre);

registro.put(**"carrera"**, carrera);

registro.put(**"universidad"**,universidad);

db.update(**"alumnos"**,registro, **"cu="**+cu, **null**);

db.close();

**et1**.setText(**""**);

**et2**.setText(**""**);

**et3**.setText(**""**);

**et4**.setText(**""**);

Toast.*makeText*(**this**, **"Se modificaron los datos correctamente"**, Toast.***LENGTH\_SHORT***).show();

}

### limpiar

**public void** limpiar (View v) {

**et1**.setText(**""**);

**et2**.setText(**""**);

**et3**.setText(**""**);

**et4**.setText(**""**);

### lista

*// Obtener una lista con todos los pokemones en la base*

**public** ArrayList<PokeItem> obtieneTodos() {

ArrayList<PokeItem> res = **new** ArrayList<PokeItem>();

*//modificar si quiero tmb el id*

String strQuery = **"select nombre, foto, descripcion from pokemon;"**;

SQLiteDatabase db = **this**.getReadableDatabase();

Cursor cursor = db.rawQuery(strQuery, **null**);

*// recorrer el cursor para sacar cada item*

**if** (cursor.moveToFirst()) {

**do** {

PokeItem item = **new** PokeItem();

item.setNombre(cursor.getString(0));

item.setFoto(cursor.getString(1));

item.setDesc(cursor.getString(2));

*// agregar a mi arreglo*

res.add(item);

} **while** (cursor.moveToNext());

}

*// regresar el arreglo*

**return** res;

}

## Toast

Toast.*makeText*(**this**, **"Se cargaron los datos adecuadamente"**, Toast.***LENGTH\_SHORT***).show();

### Opción 2

Context context = this; // o tmb getApplicationContext();

String text = "Capturado!";

int duration = Toast.LENGTH\_SHORT;

Toast toast = Toast.makeText(context, text, duration);

toast.show();

finish();

## listView

### layout

<**ListView**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:id="@+id/listView"**

**android:layout\_gravity="left|top"** />

### 

### clase

**import** android.database.Cursor;

**import** android.os.Bundle;

**import** android.support.v4.app.Fragment;

**import** android.support.v4.widget.SimpleCursorAdapter;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.ListView;

**import** android.widget.Toast;

*/\*\**

*\* A simple {****@link*** *Fragment} subclass.*

*\*/*

**public class** FragmentoLista **extends** Fragment {

ListView **lv**;

InterfazBD **iBD**;

**public** FragmentoLista() {

*// Required empty public constructor*

}

@Override

**public** View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

*// Inflate the layout for this fragment*

Cursor res;

*//adaptador*

SimpleCursorAdapter sca;

View v=**super**.onCreateView(inflater,container,savedInstanceState);

String []arregloColumnas={**"\_id"**, **"datos"**};

**int** []to={R.id.***text1***, R.id.***text2***};

*//crear la conexion la bd*

**iBD** = **new** InterfazBD(**this**.getActivity());

*//cursor de la bd con los resultados de la tabla*

res = **iBD**.traerDatos();

*//pasarle el cursor a la actividad y crear el adpatador para mostrar los datos*

sca= **new** SimpleCursorAdapter(**this**.getActivity(),R.layout.***formato\_renglon***,res,arregloColumnas,to,0);

**this**.setListAdapter(sca);

**return** v;

}

**public void** onListItemClick (ListView l, View v, **int** position, **long** id){

**if**(**iBD**==**null**)

**iBD** = **new** InterfazBD(**this**.getActivity());

String s=**iBD**.traerDatos(id);

Toast t = Toast.*makeText*(getActivity(),s, Toast.***LENGTH\_LONG***);

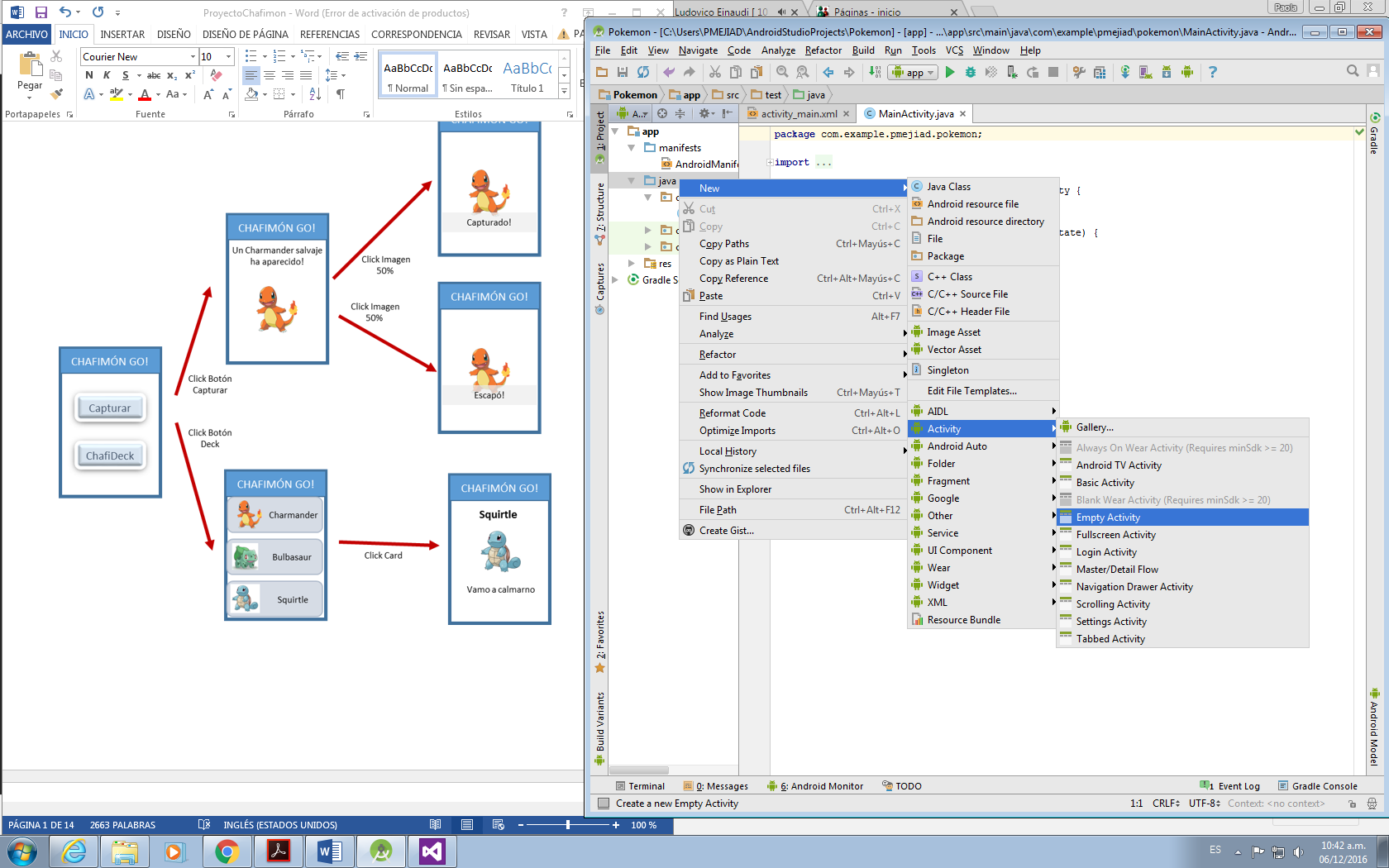
t.show();

}

}

## Nueva Actividad

En la carpeta donde están las clases principales, hacer clic derecho y luego New, activity, Empty Activity. Posteriormente, agregar este método en la actividad actual e incorporársel al botón que lanzará al nuevo Activity.



**public void** nuevaVentana(View v){

Intent intent;

//Cambiar Main2Activity por el nombre de la nueva actividad

intent = **new** Intent(**this**,Main2Activity.**class**);

startActivityForResult(intent,0);

}

**public void** abreCaptura(View view)

{

Intent intent = **new** Intent(**this**, Captura.**class**);

startActivity(intent);

}

En boton

**android:onClick="nuevaVentana"**

## fragmentos

### layout fragmento

<**RelativeLayout**

**android:layout\_width="match\_parent"**

**android:layout\_height="130dp"**

**android:background="#0066cc"**

**android:id="@+id/fragmento\_agregar"**

**android:layout\_alignParentTop="true"**

**android:layout\_alignParentLeft="true"**

**android:layout\_alignParentStart="true"**></**RelativeLayout**>

<**RelativeLayout**

**android:layout\_width="match\_parent"**

**android:layout\_height="200dp"**

**android:background="#cdcdcd"**

**android:layout\_below="@+id/fragmento\_agregar"**

**android:layout\_marginTop="5dp"**

**android:layout\_alignParentLeft="true"**

**android:id="@+id/actividadPrincipal"**

**android:layout\_alignParentStart="true"**></**RelativeLayout**>

### crear nuevo fragmento

**import** android.app.FragmentManager;

**import** android.app.FragmentTransaction;

**import** android.os.Bundle;

**import** android.support.v4.app.Fragment;

**import** android.view.LayoutInflater;

**import** android.view.View;

**import** android.view.ViewGroup;

**import** android.widget.Button;

**import** android.widget.EditText;

**import** android.widget.Toast;

*/\*\**

*\* A simple {****@link*** *Fragment} subclass.*

*\*/*

**public class** BlankFragment **extends** Fragment {

Button **agregar**;

EditText **dato**;

InterfazBD **iBD**;

FragmentManager **fm**;

**public** BlankFragment() {

*// Required empty public constructor*

}

@Override

**public** View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View v = inflater.inflate(R.layout.***fragment\_blank***,container, **false**);

**agregar**=(Button)v.findViewById(R.id.***btAgregar***);

**dato**= (EditText)v.findViewById(R.id.***datoUsuario***);

**fm**= **this**.getActivity().getFragmentManager();

**agregar**.setOnClickListener(**new** View.OnClickListener() {

@Override

**public void** onClick(View v) {

String s=**dato**.getText().toString();

*//Instanciar objeto para conexion a bd*

**iBD**=**new** InterfazBD(v.getContext());

*//Insertar dato en la bd - ver el codigo de insertarDatos en la clase*

//InterfazBD

**long** clave=**iBD**.insertarDatos(s);

*//Mostrar la llave primaria en un toast*

Toast.*makeText*(v.getContext(),**"Llave: "**+clave,Toast.***LENGTH\_SHORT***).show();

FragmentTransaction ft=**fm**.beginTransaction();

android.app.Fragment fl=**new** FragmentoLista();

ft.replace(R.id.***actividadPrincipal***,fl);

ft.commit();

}

});

**return** v;

}

}

### agregar fragmentos en Main Activity

**import** android.support.v4.app.Fragment;

**import** android.support.v4.app.FragmentManager;

**import** android.support.v4.app.FragmentTransaction;

**import** android.support.v7.app.AppCompatActivity;

**import** android.os.Bundle;

**public class** MainActivity **extends** AppCompatActivity {

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

FragmentManager fm = getFragmentManager();

FragmentTransaction ft = fm.beginTransaction();

Fragment f1 = **new** BlankFragment();

ft.add(R.id.***fragmento\_agregar***,f1);

ft.commit();

Fragment f2 = **new** FragmentoLista();

ft=fm.beginTransaction();

ft.add(R.id.***actividadPrincipal***,f2);

ft.commit();

}

### Fragmento Lista

**public class** FragmentoLista **extends** Fragment {

ListView **lv**;

InterfazBD **iBD**;

**public** FragmentoLista() {

*// Required empty public constructor*

}

@Override

**public** View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

*// Inflate the layout for this fragment*

Cursor res;

*//adaptador*

SimpleCursorAdapter sca;

View v=**super**.onCreateView(inflater,container,savedInstanceState);

String []arregloColumnas={**"\_id"**, **"datos"**};

**int** []to={R.id.***text1***, R.id.***text2***};

*//crear la conexion la bd*

**iBD** = **new** InterfazBD(**this**.getActivity());

*//cursor de la bd con los resultados de la tabla*

res = **iBD**.traerDatos();

*//pasarle el cursor a la actividad y crear el adpatador para mostrar los datos*

sca= **new** SimpleCursorAdapter(**this**.getActivity(),R.layout.***formato\_renglon***,res,arregloColumnas,to,0);

**this**.setListAdapter(sca);

**return** v;

}

**public void** onListItemClick (ListView l, View v, **int** position, **long** id){

**if**(**iBD**==**null**)

**iBD** = **new** InterfazBD(**this**.getActivity());

String s=**iBD**.traerDatos(id);

Toast t = Toast.*makeText*(getActivity(),s, Toast.***LENGTH\_LONG***);

t.show();

}

}

## Blank Fragment

**public class** BlankFragment **extends** Fragment {

Button **agregar**;

EditText **dato**;

InterfazBD **iBD**;

FragmentManager **fm**;

**public** BlankFragment() {

*// Required empty public constructor*

}

@Override

**public** View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View v = inflater.inflate(R.layout.***fragment\_blank***,container, **false**);

**agregar**=(Button)v.findViewById(R.id.***btAgregar***);

**dato**= (EditText)v.findViewById(R.id.***datoUsuario***);

**fm**= **this**.getActivity().getFragmentManager();

**agregar**.setOnClickListener(**new** View.OnClickListener() {

@Override

**public void** onClick(View v) {

String s=**dato**.getText().toString();

*//Instanciar objeto para conexion a bd*

**iBD**=**new** InterfazBD(v.getContext());

*//Insertar dato en la bd - ver el codigo de insertarDatos en la clase*

*//InterfazBD*

**long** clave=**iBD**.insertarDatos(s);

*//Mostrar la llave primaria en un toast*

Toast.*makeText*(v.getContext(),**"Llave: "**+clave,Toast.***LENGTH\_SHORT***).show();

FragmentTransaction ft=**fm**.beginTransaction();

Fragment fl=**new** FragmentoLista();

ft.replace(R.id.***actividadPrincipal***,fl);

ft.commit();

}

});

**return** v;

}

}

## Layout

### botón

<**LinearLayout**

**android:orientation="horizontal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**>

<**Button**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Captura"**

**android:onClick="abreCaptura"**

**android:id="@+id/Captura"** />

</**LinearLayout**>

### textView

<**LinearLayout**

**android:orientation="horizontal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**>

<**TextView**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Numero 1:"**

**android:id="@+id/textView"** />

<**EditText**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:id="@+id/editText"**

**android:inputType="number"**

**android:layout\_weight="1"** />

</**LinearLayout**>

<**LinearLayout**

**android:orientation="horizontal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="56dp"**>

<**TextView**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Numero 2:"**

**android:id="@+id/textView2"** />

<**EditText**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:id="@+id/editText2"**

**android:inputType="number"**

**android:layout\_weight="1"** />

</**LinearLayout**>

### Login

<**LinearLayout**

**android:orientation="vertical"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**android:layout\_alignParentTop="true"**

**android:layout\_alignParentLeft="true"**

**android:layout\_alignParentStart="true"**

**android:weightSum="1"**>

<**LinearLayout**

**android:orientation="horizontal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**>

<**TextView**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Clave Unica:"**

**android:id="@+id/textView"** />

</**LinearLayout**>

<**LinearLayout**

**android:orientation="horizontal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**>

<**EditText**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:id="@+id/claveU"**

**android:inputType="number"** />

</**LinearLayout**>

<**LinearLayout**

**android:orientation="horizontal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**>

<**TextView**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Nombre:"**

**android:id="@+id/textView2"**/>

</**LinearLayout**>

<**LinearLayout**

**android:orientation="horizontal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**>

<**EditText**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:id="@+id/editText"** />

</**LinearLayout**>

<**LinearLayout**

**android:orientation="vertical"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**>

<**TextView**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Carrera:"**

**android:id="@+id/textView3"**/>

</**LinearLayout**>

<**LinearLayout**

**android:orientation="vertical"**

**android:layout\_width="match\_parent"**

**android:layout\_height="42dp"**

**android:layout\_gravity="center\_horizontal"**>

<**EditText**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:id="@+id/editText2"**/>

</**LinearLayout**>

<**LinearLayout**

**android:orientation="vertical"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"** >

<**TextView**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Universidad:"**

**android:id="@+id/textView4"**/>

</**LinearLayout**>

<**LinearLayout**

**android:orientation="vertical"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**>

<**EditText**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:id="@+id/editText3"**/>

</**LinearLayout**>

<**LinearLayout**

**android:orientation="horizontal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="44dp"**

**android:layout\_gravity="center\_horizontal"**>

<**Button**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="AGREGAR"**

**android:id="@+id/btAgregar"**

**android:onClick="alta"**/>

<**Button**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Eliminar"**

**android:onClick="baja"**

**android:id="@+id/btEliminar"** />

</**LinearLayout**>

<**LinearLayout**

**android:orientation="horizontal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**>

<**Button**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="CONSULTA CU"**

**android:id="@+id/btConsulta"**

**android:onClick="consulta"**/>

<**Button**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="MODIFICAR"**

**android:onClick="modificar"**

**android:id="@+id/btModificar"** />

</**LinearLayout**>

<**LinearLayout**

**android:orientation="horizontal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:layout\_gravity="center\_horizontal"**>

<**Button**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="LIMPIAR"**

**android:id="@+id/btLimpiar"**

**android:onClick="limpiar"**/>

</**LinearLayout**>

</**LinearLayout**>

//

<**Button**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Regresar"**

**android:id="@+id/btRegresar"**

**android:layout\_gravity="center\_horizontal"**

**android:onClick="RegresarPantalla"**/>