Formulario DAI

Android Studio:

**.xml:**

<LinearLayout android:orientation="vertical o horizontal"

<TextView android:text="…"/>

<EditText Android:inputType="text"/>

<Button android:onClick="nombreMetodoDeJava"/>

<WebView/> En project/app/src/main/AndroidManifest se pone <uses-permission android:name="android.permission.INTERNET"></uses-permission> entre package y application

<GridView android:columnWidth="90dp",android:numColumns="auto\_fit"/>

<Generales android:layout\_width="match\_parent", android:layout\_height= "wrap\_content", android:id="@id/nombre />

**.java:** Se crean todos los elementos como private al inicio y se asignan después en los métodos; todos los métodos excepto el onCreate llevan como parámetros View v; para mandar mensajes se hace:

Toast.makeText(this, "…", Toast.LENGTH\_SHORT).show();

En el onCreate se crean todos los valores del xml para manipularlos haciendo:

nombre = (nombreElemento)findViewById(R.id.idDelXML);

WebView: se hace en el onCreate

webView.getSettings().setJavaScriptEnabled(true);

webView.loadUrl("http://bahidora.com/");

GridView: se hace en el onCreate

gridView.setAdapter(new ImageAdapter(this));

Es necesario crear una clase en Java que se llame ImageAdapter y agregar imágenes en android/app/res/drawable

public class ImageAdapter extends BaseAdapter {

private Context mContext;

public ImageAdapter(Context c) {

mContext = c;

}

public int getCount() {

return mThumbIds.length;

}

public Object getItem(int position) {

return null;

}

public long getItemId(int position) {

return 0;

}

public View getView(int position, View convertView, ViewGroup parent) {

ImageView imageView;

if (convertView == null) {

imageView = new ImageView(mContext);

imageView.setLayoutParams(new GridView.LayoutParams(85, 85));

imageView.setScaleType(ImageView.ScaleType.CENTER\_CROP);

imageView.setPadding(8, 8, 8, 8);

} else {

imageView = (ImageView) convertView;

}

imageView.setImageResource(mThumbIds[position]);

return imageView;

}

private Integer[] mThumbIds = {

R.drawable.nombreArchivo1, R.drawable.nombreArchivo2

};

}

Intent: Se divide en 2 páginas y la primera usa el método del botón y la segunda está en el onCreate

Intent nombreIntent = new Intent(MainActivity.this, Main2Activity.class);

Bundle nombreBundle = new Bundle();

nombreBundle.putString("nombreTexto", nombreElemento.getText().toString()); //Nombre texto se asigna en este momento

nombreIntent.putExtras(nombreBundle);

startActivity(nombreIntent);

---

Bundle nombreBundle = this.getIntent().getExtras();

nombreElemento.setText("…"+bundle.get("nombreTexto"));//Poner el nombre de la página anterior

Bases de Datos: Es necesario crear una clase en java que se llame AdminSQLiteOpenHelper que es:

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 sqLiteDatabase) {

sqLiteDatabase.execSQL("create table nomTabla(nomColumna1 integer primary key, nomColumna2 text)");

}

@Override

public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

sqLiteDatabase.execSQL("drop table if exists nomTabla");

sqLiteDatabase.execSQL("create table nomTabla(nomColumna1 integer primary key, nomColumna2 text)");

}

}

Todos los métodos llevan:

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

SQLiteDatabase bd = admin.getWritableDatabase();

String cu = nombreElemento.getText().toString();//Aunque sea int

bd.update y bd.delete regresan 1 si se logran hacer, lo que sirve para mandar mensajes if (cant==1)

Y al final bd.close();

Para alta: ContentValues registro = new ContentValues();

registro.put("nombreColumna",cu);

bd.insert("nombreTabla", null, registro);

Para consulta: Cursor fila=bd.rawQuery("select … where cu="+cu, null);

If(fila.moveToFirst()){

nombreElemento.setText(fila.getString(0));

nombreElemento2.setText(fila.getString(1));} //Va regresando el dato 1, 2,etc.

Para baja: bd.delete("nombreTabla", "cu="+cu, null);

Para modificación: bd.update("nombreTabla", registro, "cu="+cu, null); //Hace falta hacer el registro otra vez.

Excel: Todos los If condición Then, Case, With los cierro al final con End y su nombre.

En el UserForm\_Intialize(), cargo cosas a un comboBox haciendo:

ComboBox.Clear //Esto lo hago siempre.

With Combobox

.AddItem "…"

.AddItem "…"

En métodos normales, para consultar el valor de un comboBox hago

If ComboBox.Values = "…" Then

Si es un parámetro es

Select Case ComboBox

Case "…"

Suma = suma + …

Case "…"

En las funciones los parámetros se dan como nombre1 As String, nombre2 As Boolean; se pone al final el nombre de la función = lo que se quiera regresar, si lo que se quiere regresar es un String pero esta en int se hace Cstr(valor); además, las variables se declaran como Dim nombreVariable As Double.

Para imprimir cosas se usan los label y se pone label.Caption=".." o para llamar a una función se hace label.Caption=Format(nombreFuncion(parametro1, parametro2))

Visual Studio:

**WPF:** MessageBox.Show("");

Conexión:

public static SqlConnection agregarConexion(){

SqlConnection con = new SqlConnection("Data Source = PC; Initial Catalog = Prueba1; User ID = sa; Password = sqladmin");

con.Open();

return con;

}

public void llenarCombo(ComboBox cb){

SqlConnection con = Conexion.agregarConexion();

SqlCommand cmd = new SqlCommand("select cu from alum", con);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read()){

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

}

dr.Close();

}

Clase:

Public int nombre {get;set;}

Constructor (con id de nombre distinto)

Public static int nombreMetodo(nombreClase a){

SqlConnection con = Conexión.agregarConexion();

SqlCommand cmd = new SqlCommand (String.format("insert into nombreTabla(nombreColumna1, nombreColumna2) values ('{0}','{1}')", a.nombre, a.nombre2), con);

int res = cmd.ExecuteNonQuery();

con.close();

return res;}

MainWindow:

loaded{

Conexion c = new Conexion

c.llenarCombo();}

botónCambioPágina{

this.hide();

nomPag w = new NomPag();

w.show();}

botonSobreClase(){

SqlConnection con = Conexion.agregarConexion();

SqlCommand cmd = new SqlCommand("", con);

Object o = cmd.executeScalar();

aux=o.toString();

Clase a = new Clase(int.parse(cb.text/cb.selectedItem.toString(), textBox.text, datePicker.text);

int res = Clase.nombreMetodo(a);

if(res>o)…}

**Sitio web:** Todas las páginas llevan el conectarBD() y para consultar la localización de una cookie se va a opciones de internet/general/configuración/ archivos temporales de internet/ver archivos

Protected OdbcConnection conectarBD(){

OdbcConnection con = new OdbcConnection("Driver={SQL Server Native Client 11.0};Server=CC102-18;Uid=sa;Pwd=sqladmin;Database=Juegos");

con.open();

label.text="Conexión exitosa";

return con;}

protected void botonCambioPágina(){

OdbcConnection con = conectarBD();

OdbcCommand cmd = new OdbcCommand(""),con

OdbcDataReader dr = cmd.ExecuteReader();

If(dr.hasRows){

dr.read()}

Session["nombreColumna"]=dr.getInt32(0).toString();//dr.getString(0)

Response.Redirect("nombrePagina2.aspx");

dr.close();

con.close();}

En la próxima página lo recupero en donde quiera declararlo con un Session["nombreColumna"].toString();

Los GridView y DropDownList se cargan cuando hago el loaded o DropDownList\_SelectedIndexChanged

OdbcConnection con = conectarBD();

If(!IsPostBack){//Solo se hace si es loaded

OdbcCommand cmd=new OdbcCommand("",con);

OdbcDataReader dr = cmd.ExecuteReader();

nombreGridview.DataSource=dr; //GridView

nombregridview.dataBind();

nombreDropDownList.Items.Clean(); //DropDownList

while(dr.read()){

nombreDropDownList.Items.Add(dr.getString(0));}

dr.read(); //Label

nombreLabel.text=dr.getString(0);

dr.close();

con.close();}

SQL:

update Prof set NomProf = 'Miau', Categoría = 'tc' where idProf=1

insert into Prof values (10, 'Guau', 'tc')

delete from Prof where IdProf=10

select avg(IdProf) from Prof

alter table Prof (add/drop Column) Prueba1 int

select \* from prof where IdProf (in (1,3,7)/between 1 and 7)

select \* from Prof where NomProf (= 'Miau'/like'M%')

select \* from Prof order by NomProf asc/desc

select distinct nomMat from Mater, Grupo, Prof where Mater.ClaveM = Grupo.ClaveM and Grupo.IdProf = Prof.IdProf and NomProf = 'Miau'