**1.- ExampleInstrumentedTest.java**

package com.tarea2.ap1;

import android.content.Context;

import android.support.test.InstrumentationRegistry;

import android.support.test.runner.AndroidJUnit4;

import org.junit.Test;

import org.junit.runner.RunWith;

import static org.junit.Assert.\*;

/\*\*

\* Instrumented test, which will execute on an Android device.

\*

\* @see <a href="http://d.android.com/tools/testing">Testing documentation</a>

\*/

@RunWith(AndroidJUnit4.class)

public class ExampleInstrumentedTest {

@Test

public void useAppContext() {

// Context of the app under test.

Context appContext = InstrumentationRegistry.getTargetContext();

assertEquals("com.marcemma.ap1", appContext.getPackageName());

}

}

**2.- Androidmanifest.xml**

<?xmlversion="1.0"encoding="utf-8"?>

<manifestxmlns:android="http://schemas.android.com/apk/res/android"

package="com.tarea2.ap1"

android:versionCode="1"

android:versionName="1.0">

<uses-sdk

android:minSdkVersion="8"

android:targetSdkVersion="16"/>

<application

android:allowBackup="true"

android:icon="@drawable/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppTheme">

<activity

android:name="com.example.ap1.P1"

android:label="@string/app\_name">

<intent-filter>

<actionandroid:name="android.intent.action.MAIN"/>

<categoryandroid:name="android.intent.category.LAUNCHER"/>

</intent-filter>

</activity>

<activityandroid:name="P2"></activity>

<activityandroid:name="P3"></activity>

</application>

</manifest>

<!--Este archivo de manifiesto proporciona información esencial sobre tu aplicación al sistema Android, información que el sistema debe tener para poder ejecutar el código de la app.

Nombra el paquete de Java para la aplicación. El nombre del paquete sirve como un identificador único para la aplicación.

Describe los componentes de la aplicación, como las actividades, los servicios, los receptores de mensajes y los proveedores de contenido que la integran.

Determina los procesos que alojan a los componentes de la aplicación.-->

**3.- ExampleUnitTest.java**

package com.tarea2.ap1;

import org.junit.Test;

import static org.junit.Assert.\*;

/\*\*

\* Example local unit test, which will execute on the development machine (host).

\*

\* @see <a href="http://d.android.com/tools/testing">Testing documentation</a>

\*/

public class ExampleUnitTest {

@Test

public void addition\_isCorrect() {

assertEquals(4, 2 + 2);

}

}

**4.- curp.java**

package com.tarea2.ap1; /\*paquetería\*/

public class Curp { /\*nombre de la clase principal\*/

private String nombre; /\*representa cadenas de caracteres\*/

private String paterno; /\*representa cadenas de caracteres\*/

private String materno; /\*representa cadenas de caracteres\*/

private String sexo; /\*representa cadenas de caracteres\*/

private String estado; /\*representa cadenas de caracteres\*/

privateintdd;

privateintmm;

privateintyyyy;

/\*Arreglo de tipo String que contiene las

\* claves de las entidades federativas de México\*/

finalstatic String claves[] = {"AS","BS","BC","CC","CS","CH","CL",

"CM","DF","DG","GT","GR","HG","JC","MC","MN","MS","NT","NL","OC",

"PL","QT","QR","SP","SL","SR","TC","TS","TL","VZ","YN","ZS","NE"};

publicCurp(){

this.nombre = null; /\*es el contexto del estado actual de la aplicación / objeto\*/

this.paterno = null; /\*es el contexto del estado actual de la aplicación / objeto\*/

this.materno = null; /\*es el contexto del estado actual de la aplicación / objeto\*/

this.sexo = null; /\*es el contexto del estado actual de la aplicación / objeto\*/

this.estado = null; /\*es el contexto del estado actual de la aplicación / objeto\*/

this.mm = 0; /\*es el contexto del estado actual de la aplicación / objeto inicializado en 0\*/

this.dd = 0; /\*es el contexto del estado actual de la aplicación / objeto inicializado en 0\*/

this.yyyy = 0; /\*es el contexto del estado actual de la aplicación / objeto inicializado en 0\*/

}

public String getNombre(){ /\*representa cadenas de caracteres del objeto getNombre\*/

returnthis.nombre; /\*retorna el valor según el estado de "nombre"\*/

}

publicvoidsetNombre(String n){

this.nombre = n; /\*es el contexto del estado actual del objeto\*/

}

public String getPaterno() /\*representa cadenas de caracteres del objeto getPaterno\*/

returnthis.paterno; /\*retorna el valor según el estado de "paterno"\*/

}

publicvoidsetPaterno(String n){

this.paterno = n; /\*es el contexto del estado actual del objeto\*/

}

public String getMaterno(){ /\*representa cadenas de caracteres del objeto getMaterno\*/

returnthis.materno; /\*retorna el valor según el estado de "materno"\*/

}

publicvoidsetMaterno(String n){

this.materno = n; /\*es el contexto del estado actual del objeto\*/

}

public String getSexo(){ /\*representa cadenas de caracteres del objeto getSexo\*/

returnthis.sexo; /\*retorna el valor según el estado de "sexo"\*/

}

publicvoidsetSexo(String n){

this.sexo = n; /\*es el contexto del estado actual del objeto\*/

}

public String getEstado(){ /\*representa cadenas de caracteres del objeto getEstado\*/

returnthis.estado; /\*retorna el valor según el estado de "estado"\*/

}

publicvoidsetEstado(String n){

this.estado = n; /\*es el contexto del estado actual del objeto\*/

}

publicintgetDD(){

returnthis.dd;

}

publicvoidsetDD(int n){

this.dd = n;

}

publicintgetMM(){

returnthis.mm;

}

publicvoidsetMM(int n){

this.mm = n;

}

publicintgetYYYY(){

returnthis.yyyy;

}

publicvoidsetYYYY(int n){

this.yyyy = n;

}

public String generaClave(int id){

String re = claves[id];

return re;

}

public String buscaVocales(String en){ /\*Arreglo de tipo String que contiene las vocales admitidas\*/

String re = "";

en.toUpperCase();

for (int i=1; i<en.length(); i ++){

if(en.charAt(i) == 'A' || en.charAt(i) == 'E'

|| en.charAt(i) == 'I' || en.charAt(i) == 'O'

|| en.charAt(i) == 'U'){

re=""+en.charAt(i);

break;

}

}

return re;

}

public String buscaConsonantes(String en){ /\*Arreglo de tipo String que contiene las consonantes admitidas\*/

String re = "";

en.toUpperCase();

for (int i=1; i<en.length(); i ++){

if(en.charAt(i) == 'B' || en.charAt(i) == 'C'

|| en.charAt(i) == 'D' || en.charAt(i) == 'F'

|| en.charAt(i) == 'G' || en.charAt(i) == 'H'

|| en.charAt(i) == 'J' || en.charAt(i) == 'K'

|| en.charAt(i) == 'L' || en.charAt(i) == 'M'

|| en.charAt(i) == 'N' || en.charAt(i) == 'Ñ'

|| en.charAt(i) == 'P' || en.charAt(i) == 'Q'

|| en.charAt(i) == 'R' || en.charAt(i) == 'S'

|| en.charAt(i) == 'T' || en.charAt(i) == 'V'

|| en.charAt(i) == 'W' || en.charAt(i) == 'X'

|| en.charAt(i) == 'Y' || en.charAt(i) == 'Z'){

re=""+en.charAt(i);

break;

}

}

return re;

}

public String generaCurp(String clave){ /\*Arreglo de tipo String que crea la CURP del usuario\*/

String ddd,mmm,yyy;

yyy = ""+this.yyyy;

if(this.dd<10){

ddd = "0"+this.dd;

}else{ddd = ""+this.dd;}

if(this.mm<10){

mmm = "0"+this.mm;

}else{mmm = ""+this.mm;}

String re =""+this.paterno.charAt(0)+buscaVocales(this.paterno)+

this.materno.charAt(0)+this.nombre.charAt(0)+yyy.charAt(2)+

yyy.charAt(3)+mmm+ddd+this.sexo.charAt(0)+

clave+buscaConsonantes(this.paterno)+

buscaConsonantes(this.materno)+buscaConsonantes(this.nombre);

return re;

}

}

**5.- P1.java**

package com.tarea2.ap1; /\*paquetería\*/

/\*importaciones iniciales\*/

importandroid.app.Activity;

importandroid.content.Intent;

importandroid.os.Bundle;

importandroid.view.Menu;

importandroid.view.MenuItem;

publicclass P1 extends Activity {

@Override

protectedvoidonCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_p1);

/\*

\* el parametro R.layout.activity\_p1 corresponde al

\* fichero activity\_p1.xml y a los objetos contenidos

\* en el.

\*/

}

@Override

publicbooleanonCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.activity\_p1, menu);

returntrue;

}

//metodo que llama a otro Activity

publicbooleanonOptionsItemSelected(MenuItem mi){

//evalua la opcionelejida al presionar la tecla menu

switch(mi.getItemId()){

caseR.id.menu\_settings:

/\*si la opcion seleccionada corresponde al parametro

\* R.id.menu\_settings , entonces crea un nuevo

\* Intent el cual recibe dos parametros de entrada,

\* el primero corresponde a la clase actual P1.this, y

\* el segundo a la clase a la que se desea acceder

\* P2.class, la cual es un Activity\*/

Intent miIntent1 = new Intent(P1.this,P2.class);

/\*este metodo inicia la ejecucion del Activity, y

\* recibe como parametro de entrada un Intent\*/

startActivity(miIntent1);

break;

}

return true;

}

}

**6 P2.java**

package com.tarea2.ap1; /\*paquetería\*/

/\*importaciones iniciales\*/

importandroid.app.Activity;

importandroid.content.Intent;

importandroid.os.Bundle;

importandroid.view.View;

importandroid.widget.Button;

importandroid.widget.EditText;

importandroid.widget.Toast;

publicclass P2 extends Activity {

private Button bNext; /\*creación del botón\*/

privateEditTexttNombre;

privateEditTexttPaterno;

privateEditTexttMaterno;

private String datos[] = {null,null,null};

protectedvoidonCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_p2);

/\*se realiza un vinculo con los objetos contenidos

\* dentro del fichero activity\_p2.xml\*/

bNext = (Button)findViewById(R.id.buttonNext);

tNombre = (EditText)findViewById(R.id.editTextNombre);

tPaterno = (EditText)findViewById(R.id.editTextPaterno);

tMaterno = (EditText)findViewById(R.id.editTextMaterno);

/\*el metodosetOnClickListener recibe como parametro

\* un eventoOnClick\*/

bNext.setOnClickListener(newView.OnClickListener() {

@Override

publicvoidonClick(View v) {

/\*Aqui se escriben las acciones que se realizan

\* al pulsar sobre el boton.

\*/

datos[0] =tNombre.getText().toString().toUpperCase();

datos[1] =tPaterno.getText().toString().toUpperCase();

datos[2] =tMaterno.getText().toString().toUpperCase();

/\*se evalua que el contenido del Array datos sea igual

\* a null\*/

if((datos[0].equals("")) || (datos[1].equals(""))

|| (datos[2].equals(""))){

/\*el objeto Toast es un mensaje emergente de

\* cortaduracion en pantalla.\*/

Toast.makeText(getApplicationContext(),

"Faltaron campos por llenar",

1).show();

/\*los parametros del metodomakeText,

\* corresponden al contexto, al texto

\* y al tiempo que dura el texto en

\* pantalla cuando se llama al metodo

\* show()\*/

}else{

/\*si el contenido del Array datos no es

\* igual a null se crea un Intent y

\* se envian los datos al siguente

\* Activity\*/

Intent miIntent2 = new Intent(P2.this,P3.class);

/\*el metodoputExtra sirve para enviar datos

\* a otro Activity,recibe dos parametros, el

\* primero es de tipo String y es una llave

\* o nombre con el cual identificar al Intent

\* que envio los datos; el segundo parametro

\* corresponde a los datos que seran enviados,

\* pueden ser cadenas, enteros,

\* flotantes o booleanos.

\*/

miIntent2.putExtra("datosP2",datos);

startActivity(miIntent2);

}

}

});

}

}

**7.- P3.java**

package com.tarea2.ap1; /\*paquetería\*/

/\*importaciones iniciales\*/

importandroid.app.Activity;

importandroid.app.AlertDialog;

importandroid.os.Bundle;

importandroid.view.View;

importandroid.widget.AdapterView;

importandroid.widget.ArrayAdapter;

importandroid.widget.Button;

importandroid.widget.DatePicker;

importandroid.widget.Spinner;

importandroid.widget.AdapterView.OnItemSelectedListener;

publicclass P3 extends Activity {

privateDatePickerdp;

private Spinner sp1; /\*representa cadenas de caracteres\*/

private Spinner sp2; /\*representa cadenas de caracteres\*/

private Button bConsultar; /\*representa al botón\*/

privateintfecha[] = newint [3];

private String pe3[];

private String estados[];

protectedvoidonCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_p3);

/\*el objeto Bundle pe2 sirve como receptor del Intent

\* del Activity anterior, el metodogetIntent()

\* optiene al Intent, el metodogetExtras()

\* identifica los datos enviados\*/

Bundle pe2 = getIntent().getExtras();

/\*pe3 es un Array de tipo String, que

\* es igual resultado obtenido por el

\* metodogetStringArray(), el cual

\* recibe como parametro un String, el

\* cual coresponde a una llave o identificador

\* de un Intent lanzado en un Activity

\* anterior\*/

pe3 = pe2.getStringArray("datosP2");

bConsultar = (Button)findViewById(R.id.buttonConsultar);

sp1 = (Spinner)findViewById(R.id.spinnerSex);

sp2 = (Spinner)findViewById(R.id.spinnerEstado);

dp = (DatePicker)findViewById(R.id.datePickerFecha);

/\*este es un Arreglo estatico; podria ser llenado de forma

\* dinamica y luego ser cargado dentro de un ArrayAdapter

\* de tipo String para luego ser visualizado dentro de un

\* Spinner o un ListView\*/

estados = newString[] {"Aguscalientes",

"Baja California Sur","Baja California",

"Campeche","Chiapas","Chihuahua","Coahuila",

"Colima","DistritoFederal","Durango",

"Guadalajara","Guerrero","Hidalgo",

"Jalisco","México","Michoacan","Morelos",

"Nayarid","Nuevo Leon","Oaxaca","Puebla",

"Queretaro","Quintana Roo","San Luis Potosi",

"Sinaloa","Sonora","Tabasco","Tamaulipas",

"Tlaxcala","Veracruz","Yucatán","Zacatecas",

"Nacido en el Extranjero"};

/\*Un ArrayAdapter de tipo String, sirve para cargar

\* un arreglo de tipo String a un Spinner, en este caso,

\* no obstante es posible cargarlo dentro de un

\* ListView, este ArrayAdapter recibe como parametros

\* de entrada a la propia clase, a un objeto preconstruido

\* o un spinner propio personalizado, y a un arreglo de

\* tipo String; cargar elementos a un Spinner de este

\* modo es muy util cuando los datos dentro del Spinner

\* se desea que se agreguen de forma dinamica\*/

ArrayAdapter<String>adaptador = newArrayAdapter<String>

(this,android.R.layout.simple\_spinner\_item,estados);

adaptador.setDropDownViewResource

(android.R.layout.simple\_spinner\_dropdown\_item);

/\*se carga el adaptador dentro del Spinner.\*/

sp2.setAdapter(adaptador);

/\*el metodosetOnItemSelectedListener, sirve para

\* llamar a un Elemento dentro de un Spinner\*/

sp2.setOnItemSelectedListener

(newAdapterView.OnItemSelectedListener() {

publicvoidonItemSelected

(AdapterView<?> parent,

android.view.Viewvis,intpos,long id){

}

@Override

publicvoidonNothingSelected(AdapterView<?> arg0) {

/\*Aqui se escribe el codigo que se desea ejecutar

\* cuando se elije una opcion en el objeto

\* Spinner, esto es util cuando se desea usar

\* al objeto Spinner como un menu de opciones\*/

}

});

bConsultar.setOnClickListener(newView.OnClickListener() {

@Override

publicvoidonClick(View v) {

/\*se optiene la ultimaposicion visible en

\* el Spinner, gracias al metodo

\* getLastVisiblePosition()\*/

int position = sp1.getLastVisiblePosition();

/\*se optiene el valor en el objeto Spinner

\* mediant su posicion con el metodo

\* getItemAtPosition(), el cual recibe un

\* valor entero\*/

String sex =

sp1.getItemAtPosition(position).toString().toUpperCase();

int position2 = sp2.getLastVisiblePosition();

String est =

sp2.getItemAtPosition(position2).toString().toUpperCase();

fecha[0]=dp.getDayOfMonth();//se obtiene el dia

/\*se obtiene el mes y se le suma 1 debido a

\* que el rango obtenido es de 0 a 11\*/

fecha[1]=dp.getMonth()+1;

fecha[2]=dp.getYear();();//se obtiene el año

CurpmiCurp = newCurp();

miCurp.setNombre(pe3[0]);

miCurp.setPaterno(pe3[1]);

miCurp.setMaterno(pe3[2]);

miCurp.setSexo(sex);

miCurp.setEstado(est);

miCurp.setDD(fecha[0]);

miCurp.setMM(fecha[1]);

miCurp.setYYYY(fecha[2]);

dialogo(miCurp.generaCurp(miCurp.generaClave(position2)));

}

});

}

/\*se define un metodo propio el

\* cual recibe un String como parametro\*/

publicvoiddialogo(String mensage){

/\*se crea un nuevo objeto de tipo

\*AlertDialog.Builder el cual recibe

\*como parametro la propia clase \*/

AlertDialog.Builderrc = newAlertDialog.Builder(P3.this);

rc.setTitle("C.U.R.P.");

rc.setMessage(mensage);

rc.setPositiveButton("Aceptar", null);

rc.show();

}

}