Leo del futuro Recuerda.

Clave Root es 12345

Clave de admin 1234

Repositorio Git:

usaurio: LeoSan

clave: 010285jodete

------------------------------------------------------------------------------------------------------------------------

Pasos para futura Guia.

Resumen del curso.

- Curso Android Parte 1 -

Comenta de los requisitos que debe tener el equipo

- Curso Android Parte 2 -

Comenta como instalarlo JDK y el Visual Estudio.

Como Instalar Android Studio en Linux.

Paso 1:

Seguir los pasos de este link

https://medium.com/@aashimad1/install-android-studio-in-ubuntu-b8aed675849f

Paso 2: Componer java como variable de entorno

https://www.alvarolara.com/2015/08/24/como-saber-donde-esta-instalado-java-en-ubuntu/

Ruta -> de java en Linux -> /usr/lib/jvm/java-8-oracle/jre/bin

Paso 3: Intalar Android Studio

Lamentablemente se debe ejecutar el studio.sh para poder usar

/Descargas/android-studio/bin$ sh studio.sh

aqui direccciòn.

- Curso Android Parte 3 -

Comenta como crear nuestro primer proyecto android, se debe seleccionar que sistema operativo podra correr nuestra aplicaciòn.

Comenta que es un ACTIVITY y como se usa la interfaz de android estudio para poder desarrollar.

Activity:como su nombre lo indica es la actividad o tarea que se realizara en la APP, cada activity esta acompañada con dos componenentes su compornente xml que ira las especificaciones visuales y el componente .java que tendra la lògica y comportamiento de la tarea o actividad, podremos usar el modo diseño o texto para poder ir visualizando los cambios de nuestra app

- Curso Android Parte 4 -

Comenta como desplegar nuestro virtualizador de telefono para poder ver el funcionamiento de nuestra app.

- Curso Android Parte 5 -

Comenta como desplegar nuestro virtualizador directamente en un celular esto en caso que no podamos virtualizar en nuestra maquina.

- Curso Android Parte 6 -

CICLO DE VIDA DE UN ACTIVITY

Comenta el con un diagrama las funciones que tiene la logica de un activity

onCreate()-> con esto se invoca un activity y lo crea

onStar()-> inicia el activity que previamente se creo

onResume()-> muestra el activity que queremos mostrar

Distintos Caminos

Si el usurio minimiza la app:

Se ejecuta el onPause()-> pasa la app al segundo plano.

onStop()->Oculta elactivity si se pausa.

OnDestroy()-> Permite cerrar la aplicaciòn.

onRestart()-> permite iniciar el activity. -> onStar().

@Override

protected void onStart() {

super.onStart();

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

// La actividad está a punto de hacerse visible.

}

@Override

protected void onResume() {

super.onResume();

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

// La actividad se ha vuelto visible (ahora se "reanuda").

}

@Override

protected void onPause() {

super.onPause();

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

// Enfocarse en otra actividad (esta actividad está a punto de ser "detenida").

}

@Override

protected void onStop() {

super.onStop();

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

// La actividad ya no es visible (ahora está "detenida")

}

@Override

protected void onDestroy() {

super.onDestroy();

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

// La actividad está a punto de ser destruida.

}

- Curso Android Parte 7 -

Comenta los pasos para debuguear y usar los BreakPoint.

codigo Ejemplo.

package com.example.miprimerapp;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

int matematicas = 5;

int quimica = 5;

int fisica = 5;

int promedio = 0;

promedio = ( matematicas + quimica + fisica ) / 3;

if (promedio >= 6 ){

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

}else if(promedio <= 5){

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

}

}

}

- Curso Android Parte 7 -

Comenta como usar el toast.

Paso 1: se Importa la libreria

import android.widget.Toast;

Paso 2: Estructura del Toast

Toast.makeText( Lugar donde se ejecuta el toast , "string del mensaje", int Duracion).show();

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

- Curso Android Parte 8 -

- Curso Android Parte 9 -

Se realizará una pequeña calculadora para que solo sume dos valores.

Nota: Para resolver el error de permisos entra a este enlace

<https://www.youtube.com/watch?v=-ZKNmJFTOAk>

Solo debemos dar permiso a la carpeta con el siguiente comando.

Pero antes debemos ser-> sudo su -> 1234.123

root@leosan-Inspiron-5548:/dev# chmod 777 -R kvm

Diseñamos nuestra primera interfaz gráfica para hacer una calculadora

Nota Aprendizaje:

Existen dos Vistas de Diseño:

Desing Blue BluePrint: Que es como una radiografía de los componentes podemos jugar con las opciones de setear o fijar un componente en un layout se logra usando los puntos de cada componente y definir dónde iran fijo o fluides dentro del layoot.

Diseño Normal : la ventaja de esta es que podremos ver cómo el usuario final podrá ver nuestra aplicaciòn.

- Curso Android Parte 10 -

Se Explico como usar las dos librerías para manipular datos.

EditText->

TextView->

View->

Nota Investigar: Librerías para campos y para mostrar resultado.

**import** android.widget.EditText;

**import** android.widget.TextView;

**import** android.view.View;

Clase R: es una clase que android's studio genera de automaticamente sirve como puente entre la parte lógica y la parte gráfica. Ejemplo

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

- Curso Android Parte 11

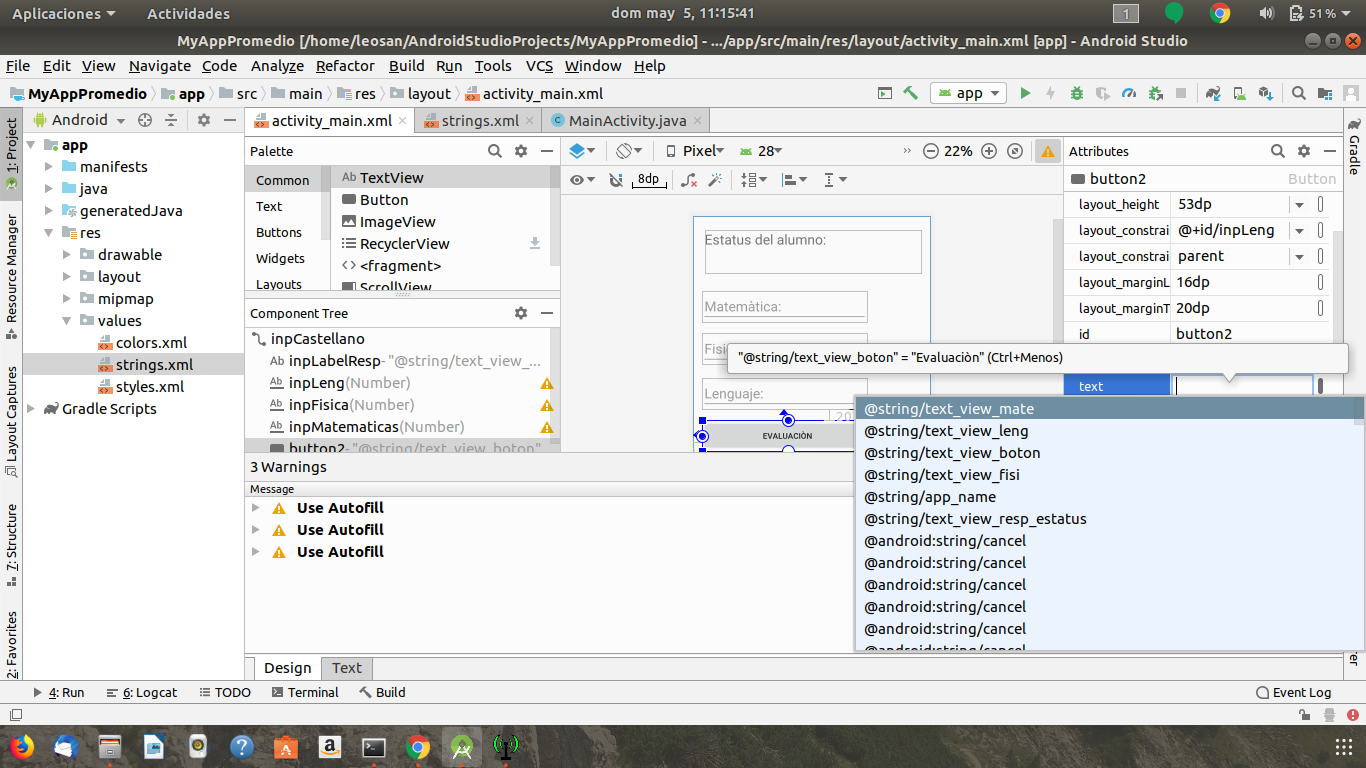
Creamos una app de promedio esta guardada en la carpeta 10 -11

- Curso Android Parte 12 -

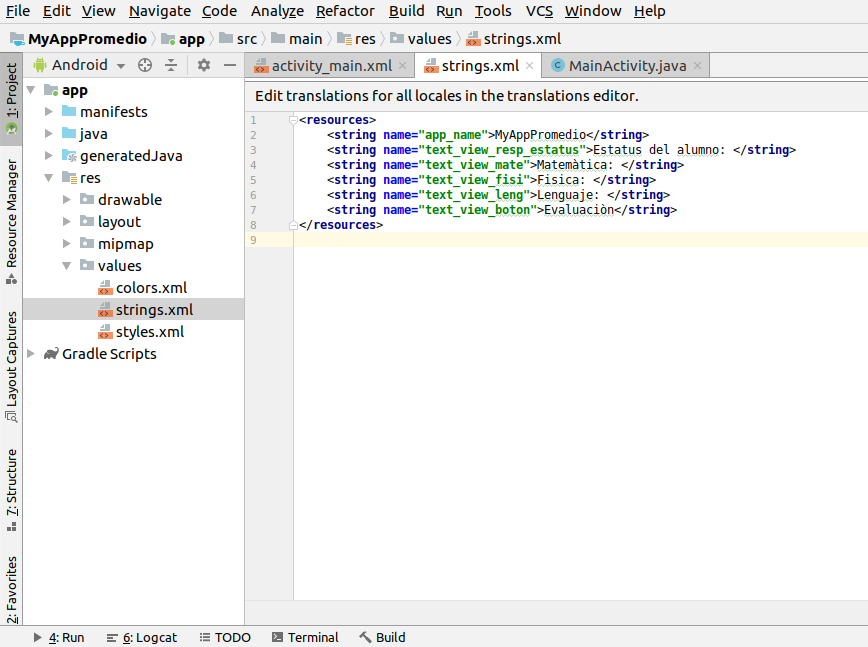
En esta parte se explica como declarar valores de variables de uso recurrente de la intefaz por ejemplo los nombre de los campos y botones.

Pasos:

Paso 1: como lo indica la imagen de android estudio se debe buscar en el repositorio res/values/strings.xml aqui se configura los string de nuestra app.

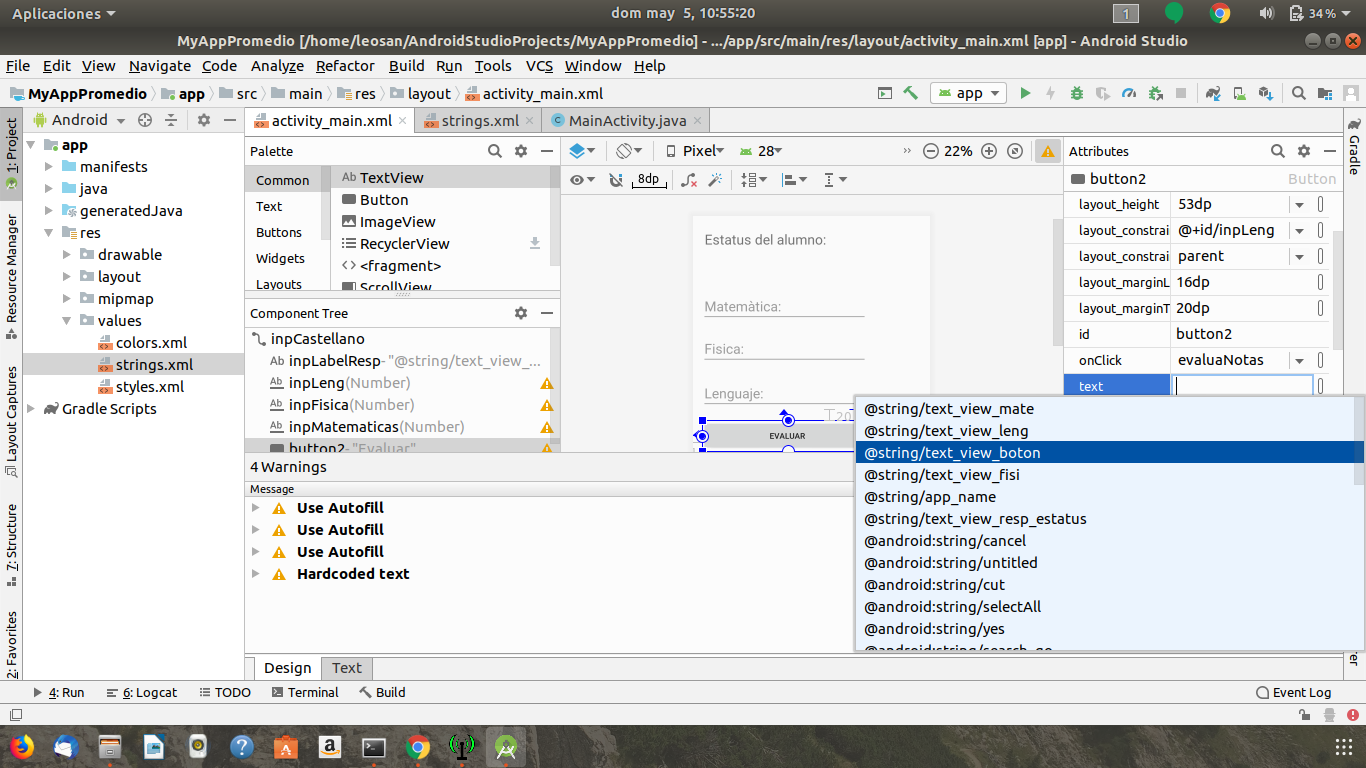


Paso 2: se crea cada string usando la plantilla xml como se muestra a continuaciòn



Paso 3: Del lado del panel del atributo de cada componente podremos hacer la referencia del string se muestra a continuaciòn.

buscamos en atributos el hint ò text de los componentes damos control + espacio y nos mostrará la lista de todos los string creados. ya con esto podremos eliminar los alert que nos indica que nos falta algun texto.



- Curso Android Parte 13 - 14 - Desarrollo del diseño de una app que use radio button

NOTA: LEONARD ES IMPORTANTE RECORDAR QUE CUANDO SE CREA UN NUEVO PROYECTO Y LO EMULAS ESTE SE DEBE CREAR OTRO DISPOSITIVO YA QUE CUANDO SE CREA OTRO DISPOSITIVO EL CREA UN DIRECTORIO UNICO DE SDK PARA EJECUTAR ESE EMULADOR, por eso cuando se mular salia las app anteriores.

En esta secciòn se nos explica como generar un radio button en android como administrarlo y como llegar a sus valores aprendimos, lo siguiente:

* Objeto (**private** RadioButton **RadioButonA**, **RadioButonB**;
* ) Tipo RadioButton: Que es el que manipula al componente:
* el metodo isCheckout() preguntamos si esta true para saber si esta marcado o no.
* en el modo diseño antes de crear un componente radio button hay que crear un Grupo de de radios este se encuentra en la secciòn componentes de nuestra paleta.

The path

'/root/Android/Sdk'

does not belong to a directory.

Android Studio will use this Android SDK instead:

'/home/leosan/Android/Sdk'

and will modify the project's local.properties file.

Failed to change read-only status for the following files: /home/leosan/AndroidStudioProjects/MyAppPromedio/.idea/libraries/Gradle\_\_android\_arch\_core\_runtime\_1\_1\_1\_aar.xml

-- Curso Android Parte 15 - 16 --

Practica para hacer una calculadora seleccionando los radio button

En esta practica tuve problemas con el emulador no me agarraba los cambios más recientes por lo que buscando en la web me tope con esto.

<https://es.stackoverflow.com/questions/103739/error-al-instalar-apk-con-android-studio-delete-failed-internal-error>

traducciòn aquí

2

Si alguien sigue teniendo este problema, les digo la solución que se me ocurrió (luego de horas de búsqueda sin éxito). Como Android Studio intenta instalar un APK para correr el emulador, se me ocurrió eliminar el APK existente del proyecto.

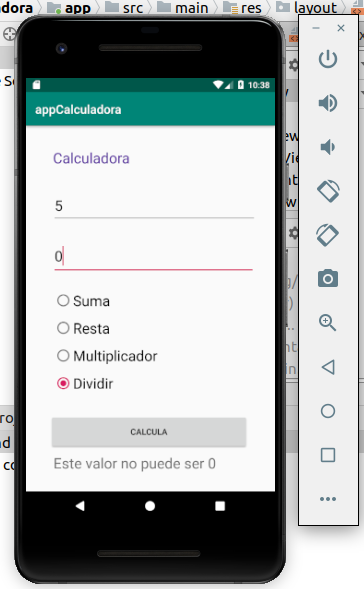
Fui a la carpeta del proyecto, donde está localizado el archivo "app-debug.apk"; (en mi caso "\NOMBRE\_DEL\_PROYECTO\app\build\outputs\apk\debug"), y para no perder ningún dato (ya que soy nueva en esto), modifiqué el archivo "app-debug.apk" por "app-debug.apk.txt" (de forma que Android Studio no lo reconozca e intente instalar nuevamente el APK). Luego cerré el emulador, y le di "Run" de nuevo. A la primera no me funcionó del todo bien, ya que estaba instalando el APK mientras corría el emulador, por lo cual tuve que cerrar el emulador de nuevo, y correrlo otra vez. Ahora está funcionando sin problemas!

Estoy trabajando con la siguiente configuración: Mi AVD es un Nexus 4 corriendo Android Oreo (API 27), cuya configuración gráfica es "Software - GLES 2.0" (por defecto, al instalar AS viene como "Automática", y eso genera otros problemas al correr la app). Además estoy en un Windows 10, utilizando Android Studio 3.1.4.

Espero a alguien le sea de ayuda!

y la ruta fue : /home/leosan/AndroidStudioProjects/appCalculadora/app/build/outputs/apk

Resultado



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**package** com.example.appcalculadora;

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

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.EditText;

**import** android.widget.RadioButton;

**import** android.widget.TextView;

**import** android.widget.Toast;

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

**private** TextView msjResp, textResultado;

**private** RadioButton rbSuma, rbResta, rbMulti, rbDiv;

**private** EditText inpNum1, inpNum2;

@Override

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

**super**.onCreate(savedInstanceState);

*//todo Aqui obtenemos todo lo relacionado con la vista Diseño pero este objeto lo llamamos R*

setContentView(R.layout.activity\_main);

*//Definimos los valores que vamos a recibir del Layout R que es la vista diseño.*

inpNum1 = (EditText)findViewById(R.id.inpNumPri);

inpNum2 = (EditText)findViewById(R.id.inpNumDos);

msjResp = (TextView)findViewById(R.id.respTitulo);

textResultado = (TextView)findViewById(R.id.textResultado);

rbSuma = (RadioButton)findViewById(R.id.radioSum);

rbResta = (RadioButton)findViewById(R.id.radioResta);

rbMulti = (RadioButton)findViewById(R.id.radioMul);

rbDiv = (RadioButton)findViewById(R.id.radioDiv);

}

*// Creamos el metodo de inicio Aqui se validan los campos y se validan los radio button*

**public void** procesaCalculo(View view){

*// todo lo que nos devuelve la view es texto. asi que declaro otras variables tipo String para recibir*

String strValorA = inpNum1.getText().toString();

String strValorB = inpNum2.getText().toString();

*// Metodo para validar Campos -- Todo - Esta Forma de llamado de funciones ->*

evaluaCampos(strValorA, **"Primer Campo vacio."** );

evaluaCampos(strValorB, **"Segundo Campo vacio."** );

*// Metodo que valida el radio si uno de los dos es true suma o resta*

procesoEvaluacion(strValorA, strValorB, rbSuma, rbResta, rbMulti, rbDiv );

}

*// Metodo para Validar.*

**public void** evaluaCampos(String valor, String msj){

*//Valido los inputs*

**if** (valor.length() == 0 ){

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, msj, Toast.LENGTH\_SHORT).show();

*//Setea la variables tipo TextView que definimos en el diseño*

textResultado.setText(msj);

}

}

*// Metodo para Validar.*

**public void** procesoEvaluacion(String strValorA, String strValorB, RadioButton rbSum, RadioButton rbRes, RadioButton rbRMulti, RadioButton rbDiv ){

*//valido para entrar al proceso*

**if** (strValorA.length() > 0 && strValorB.length() > 0 ){

*//como es un calculo debemos pasar los valores a enteros*

**int** inVAlorA = Integer.parseInt(strValorA);

**int** inVAlorB = Integer.parseInt(strValorB);

**int** intResp;

String strResul;

**if** (rbSum.isChecked() == **true** ){

intResp = inVAlorA + inVAlorB;

strResul = String.valueOf(intResp);

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, **"Total:"**+strResul, Toast.LENGTH\_SHORT).show();

*//Setea la variables tipo TextView que definimos en el diseño*

textResultado.setText(strResul);

}

**if** (rbRes.isChecked() == **true** ){

intResp = inVAlorA - inVAlorB;

strResul = String.valueOf(intResp);

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, **"Total"**+strResul, Toast.LENGTH\_SHORT).show();

*//Setea la variables tipo TextView que definimos en el diseño*

textResultado.setText(strResul);

}

**if** (rbRMulti.isChecked() == **true** ){

intResp = inVAlorA \* inVAlorB;

strResul = String.valueOf(intResp);

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, **"Total"**+strResul, Toast.LENGTH\_SHORT).show();

*//Setea la variables tipo TextView que definimos en el diseño*

textResultado.setText(strResul);

}

**if** (rbDiv.isChecked() == **true** ){

**if** (inVAlorB > 0){

intResp = inVAlorA / inVAlorB;

strResul = String.valueOf(intResp);

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, **"Total"**+strResul, Toast.LENGTH\_SHORT).show();

*//Setea la variables tipo TextView que definimos en el diseño*

textResultado.setText(strResul);

}**else**{

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, **"Este valor no puede ser 0"**, Toast.LENGTH\_SHORT).show();

*//Setea la variables tipo TextView que definimos en el diseño*

textResultado.setText(**"Este valor no puede ser 0"**);

textResultado.setHighlightColor(255);

}

}

}

}*// fin del procesoEvalua*

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*<?***xml version="1.0" encoding="utf-8"***?>*

<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

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

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

**tools:context=".MainActivity"**>

<**TextView**

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

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

**android:text="Hello World!"**

**app:layout\_constraintBottom\_toBottomOf="parent"**

**app:layout\_constraintLeft\_toLeftOf="parent"**

**app:layout\_constraintRight\_toRightOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"** />

</**android.support.constraint.ConstraintLayout**>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Curso Android Parte 17 CLASE DE RADIO BOX - SELECCIÓN MULTIPLE --

haremo un ejercicio para aprender esta práctica.

Para esta práctica lo nuevo fue el objeto que controla los checkbox llamado

**private** CheckBox **checkSuma**, **checkResta**;

**import** android.widget.CheckBox;

Dejo los resultados :

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**package** com.example.myappradiobox;

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

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.CheckBox;

**import** android.widget.EditText;

**import** android.widget.TextView;

**import** android.widget.Toast;

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

**private** TextView **textTitulo**, **respText**;

**private** EditText **inpValor1**, **inpValor2**;

**private** CheckBox **checkSuma**, **checkResta**;

@Override

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

**super**.onCreate(savedInstanceState);

*//****todo Aqui obtenemos todo lo relacionado con la vista Diseño pero este objeto lo llamamos R***

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

*//Definimos los valores que vamos a recibir del Layout R que es la vista diseño.*

**inpValor1** = (EditText)findViewById(R.id.***inpValor1***);

**inpValor2** = (EditText)findViewById(R.id.***inpValor2***);

**checkResta** = (CheckBox)findViewById(R.id.***checkBoxResta***);

**checkSuma** = (CheckBox)findViewById(R.id.***checkBoxSuma***);

**textTitulo** = (TextView)findViewById(R.id.***textTitulo***);

**respText** = (TextView)findViewById(R.id.***respApp***);

}

*// Creamos el metodo de inicio Aqui se validan los campos y se validan los radio button*

**public void** procesaCalculo(View view){

*//* ***todo lo que nos devuelve la view es texto. asi que declaro otras variables tipo String para recibir***

String strValorA = **inpValor1**.getText().toString();

String strValorB = **inpValor2**.getText().toString();

*// Metodo para validar Campos --* ***Todo - Esta Forma de llamado de funciones ->***

evaluaCampos(strValorA, **"Primer Campo vacio."** );

evaluaCampos(strValorB, **"Segundo Campo vacio."** );

*// Metodo que valida el radio si uno de los dos es true suma o resta*

procesoEvaluacion(strValorA, strValorB, **checkSuma**, **checkResta**);

}

*// Metodo para Validar.*

**public void** evaluaCampos(String valor, String msj){

*//Valido los inputs*

**if** (valor.length() == 0 ){

*//Muestra mensaje como un echo de PHP*

Toast.*makeText*(**this**, msj, Toast.***LENGTH\_SHORT***).show();

*//Setea la variables tipo TextView que definimos en el diseño*

**respText**.setText(msj);

}

}

**public void** procesoEvaluacion( String strValorA, String strValorB, CheckBox radioBoxSuma, CheckBox radioBoxResta ) {

**int** inVAlorA = Integer.*parseInt*(strValorA);

**int** inVAlorB = Integer.*parseInt*(strValorB);

**int** intRespSuma, intRespResta ;

String strResul =**""**;

*//valido para entrar al proceso*

**if** (strValorA.length() > 0 && strValorB.length() > 0 ) {

**if** ( radioBoxResta.isChecked() == **true** ){

intRespResta = inVAlorA - inVAlorB;

strResul = **"La Resta es "** + String.*valueOf*(intRespResta);

*//Muestra mensaje como un echo de PHP*

Toast.*makeText*(**this**, strResul, Toast.***LENGTH\_SHORT***).show();

*//Setea la variables tipo TextView que definimos en el diseño*

*//respText.setText(strResulRest);*

}

**if** (radioBoxSuma.isChecked() == **true** ){

intRespSuma = inVAlorA + inVAlorB;

strResul = strResul + **" // La Suma es :"** +String.*valueOf*(intRespSuma);

*//Muestra mensaje como un echo de PHP*

Toast.*makeText*(**this**, strResul, Toast.***LENGTH\_SHORT***).show();

*//Setea la variables tipo TextView que definimos en el diseño*

*//respText.setText(strResulSuma);*

}

**respText**.setText(strResul);

}**else**{

**respText**.setText(**"Debe Ingresar algun Valor."**);

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*<?***xml version="1.0" encoding="utf-8"***?>*

<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

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

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

**tools:context=".MainActivity"**>

<**TextView**

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

**android:layout\_width="341dp"**

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

**android:layout\_marginStart="28dp"**

**android:layout\_marginLeft="28dp"**

**android:layout\_marginTop="32dp"**

**android:text="@string/app\_titulo"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"** />

<**EditText**

**android:id="@+id/inpValor1"**

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

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

**android:layout\_marginStart="28dp"**

**android:layout\_marginLeft="28dp"**

**android:layout\_marginTop="28dp"**

**android:ems="10"**

**android:hint="@string/app\_Valor1"**

**android:inputType="number"**

**android:textColorHint="#673AB7"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/textTitulo"** />

<**EditText**

**android:id="@+id/inpValor2"**

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

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

**android:layout\_marginStart="28dp"**

**android:layout\_marginLeft="28dp"**

**android:layout\_marginTop="24dp"**

**android:ems="10"**

**android:hint="@string/app\_Valor2"**

**android:inputType="number"**

**android:textColorHint="#9C27B0"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpValor1"** />

<**CheckBox**

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

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

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

**android:layout\_marginStart="8dp"**

**android:layout\_marginLeft="8dp"**

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

**android:layout\_marginEnd="8dp"**

**android:layout\_marginRight="8dp"**

**android:text="@string/app\_rb\_suma"**

**android:textSize="24sp"**

**app:layout\_constraintEnd\_toStartOf="@+id/checkBoxResta"**

**app:layout\_constraintHorizontal\_bias="0.482"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpValor2"** />

<**CheckBox**

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

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

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

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

**android:layout\_marginEnd="76dp"**

**android:layout\_marginRight="76dp"**

**android:text="@string/app\_rb\_resta"**

**android:textSize="24sp"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpValor2"** />

<**Button**

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

**style="@android:style/Widget.Button"**

**android:layout\_width="331dp"**

**android:layout\_height="48dp"**

**android:layout\_marginStart="8dp"**

**android:layout\_marginLeft="8dp"**

**android:layout\_marginTop="36dp"**

**android:layout\_marginEnd="40dp"**

**android:layout\_marginRight="40dp"**

**android:onClick="procesaCalculo"**

**android:text="@string/app\_boton"**

**android:textSize="24sp"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**app:layout\_constraintHorizontal\_bias="1.0"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/checkBoxSuma"** />

<**TextView**

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

**android:layout\_width="318dp"**

**android:layout\_height="61dp"**

**android:layout\_marginStart="8dp"**

**android:layout\_marginLeft="8dp"**

**android:layout\_marginTop="36dp"**

**android:layout\_marginEnd="8dp"**

**android:layout\_marginRight="8dp"**

**android:text="@string/app\_resultado"**

**android:textSize="24sp"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**app:layout\_constraintHorizontal\_bias="0.363"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/button"** />

</**android.support.constraint.ConstraintLayout**>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

---- Curso Android Parte 18 CLASE DE SELECT -

haremo un ejercicio para aprender esta práctica.

Para esta práctica lo nuevo fue el objeto que controla los checkbox llamado

**import** android.widget.Spinner;

**private Spinner inpSelect;**

**inpSelect = (Spinner)findViewById(R.id.*inpSelect*);**

Dejo los resultados :

*<?***xml version="1.0" encoding="utf-8"***?>*

<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

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

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

**tools:context=".MainActivity"**>

<**EditText**

**android:id="@+id/inpValor1"**

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

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

**android:layout\_marginStart="32dp"**

**android:layout\_marginLeft="32dp"**

**android:layout\_marginTop="28dp"**

**android:ems="10"**

**android:hint="@string/app\_valor1"**

**android:inputType="number"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/textTitulo"** />

<**EditText**

**android:id="@+id/inpValor2"**

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

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

**android:layout\_marginStart="32dp"**

**android:layout\_marginLeft="32dp"**

**android:layout\_marginTop="32dp"**

**android:ems="10"**

**android:hint="@string/app\_valor2"**

**android:inputType="number"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpValor1"** />

<**TextView**

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

**android:layout\_width="275dp"**

**android:layout\_height="45dp"**

**android:layout\_marginStart="32dp"**

**android:layout\_marginLeft="32dp"**

**android:layout\_marginTop="36dp"**

**android:lineSpacingExtra="8sp"**

**android:text="@string/app\_titulo"**

**android:textColor="#FFC107"**

**android:textIsSelectable="true"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"** />

<**Button**

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

**android:layout\_width="280dp"**

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

**android:layout\_marginStart="36dp"**

**android:layout\_marginLeft="36dp"**

**android:layout\_marginTop="20dp"**

**android:onClick="procesaCalculo"**

**android:text="@string/app\_boton"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpSelect"** />

<**Spinner**

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

**android:layout\_width="284dp"**

**android:layout\_height="62dp"**

**android:layout\_marginStart="32dp"**

**android:layout\_marginLeft="32dp"**

**android:layout\_marginTop="36dp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpValor2"** />

<**TextView**

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

**android:layout\_width="278dp"**

**android:layout\_height="77dp"**

**android:layout\_marginStart="32dp"**

**android:layout\_marginLeft="32dp"**

**android:layout\_marginTop="28dp"**

**android:text="@string/app\_resp"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/button"** />

</**android.support.constraint.ConstraintLayout**>

**package** com.example.myappselect;

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

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.ArrayAdapter;

**import** android.widget.CheckBox;

**import** android.widget.EditText;

**import** android.widget.Spinner;

**import** android.widget.TextView;

**import** android.widget.Toast;

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

**private** Spinner inpSelect;

**private** EditText inpValor1, inpValor2;

**private** TextView respText;

@Override

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

**super**.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

*// Todo Permite la comunicaciòn con la parte grafica*

inpValor1 = (EditText)findViewById(R.id.inpValor1);

inpValor2 = (EditText)findViewById(R.id.inpValor2);

respText = (TextView)findViewById(R.id.respValor);

inpSelect = (Spinner)findViewById(R.id.inpSelect);

*// Todo aqui se muestra como crear un componente tipo Select*

*//Arreglo con las opciones*

String [] opciones = {**"Sumar"**, **"Restar"**, **"Multiplicar"**, **"Dividir"**};

*//Debemos declarar un adaptare parapoder comunicar el arreglo con el componente*

ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, android.R.layout.simple\_list\_item\_1, opciones);

*// Aqui agrego el objeto tipo adapte al componente*

inpSelect.setAdapter(adapter);

}

*// Creamos el metodo de inicio Aqui se validan los campos y se validan los radio button*

**public void** procesaCalculo(View view){

*// todo lo que nos devuelve la view es texto. asi que declaro otras variables tipo String para recibir*

String strValorA = inpValor1.getText().toString();

String strValorB = inpValor2.getText().toString();

String seleccion = inpSelect.getSelectedItem().toString();

*// Metodo para validar Campos -- Todo - Esta Forma de llamado de funciones ->*

**if** ( evaluaCampos(strValorA, **"Primer Campo vacio."** ) == **true** && evaluaCampos(strValorB, **"Segundo Campo vacio."** ) == **true** ){

*// Metodo que valida el radio si uno de los dos es true suma o resta*

procesoEvaluacion(strValorA, strValorB, seleccion);

}**else**{

respText.setText(**"Debe Ingresar algun Valor."**);

}

}

*// Metodo para Validar.*

**public boolean** evaluaCampos(String valor, String msj){

*//Valido los inputs*

**if** (valor.length() == 0 ){

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, msj, Toast.LENGTH\_SHORT).show();

*//Setea la variables tipo TextView que definimos en el diseño*

respText.setText(msj);

**return false**;

}**else**{

**return true**;

}

}

**public void** procesoEvaluacion(String strValorA, String strValorB, String seleccion) {

**int** inVAlorA = Integer.parseInt(strValorA);

**int** inVAlorB = Integer.parseInt(strValorB);

**int** intRespSuma, intRespResta ;

String strResul = **""**;

*//valido para entrar al proceso*

**if** (strValorA.length() > 0 && strValorB.length() > 0 ) {

**if** ( seleccion.equals(**"Sumar"**) ){

intRespResta = inVAlorA + inVAlorB;

strResul = String.valueOf(intRespResta);

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, **" La suma es "**+ strResul, Toast.LENGTH\_SHORT).show();

}

**if** ( seleccion.equals(**"Restar"**) ){

intRespResta = inVAlorA - inVAlorB;

strResul = String.valueOf(intRespResta);

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, **" La Restar es "**+ strResul, Toast.LENGTH\_SHORT).show();

}

**if** ( seleccion.equals(**"Multiplicar"**) ){

intRespResta = inVAlorA \* inVAlorB;

strResul = String.valueOf(intRespResta);

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, **" La Multiplicar es "**+ strResul, Toast.LENGTH\_SHORT).show();

}

**if** ( seleccion.equals(**"Dividir"**) ){

**if** (inVAlorB!=0){

intRespResta = inVAlorA / inVAlorB;

strResul = String.valueOf(intRespResta);

*//Muestra mensaje como un echo de PHP*

Toast.makeText(**this**, **" La Dividir es "**+ strResul, Toast.LENGTH\_SHORT).show();

}**else**{

Toast.makeText(**this**, **" No se puede dividir entre 0 "**, Toast.LENGTH\_SHORT).show();

strResul = **"No se Puede dividir entre 0"**;

}

}

*//Setea la variables tipo TextView que definimos en el diseño*

respText.setText(strResul);

}**else**{

respText.setText(**"Debe Ingresar algun Valor."**);

}

}

}

spinner\_item\_view

@Override

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

**super**.onCreate(savedInstanceState);

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

*//* ***Todo Permite la comunicaciòn con la parte grafica***

**inpValor1** = (EditText)findViewById(R.id.***inpValor1***);

**inpValor2** = (EditText)findViewById(R.id.***inpValor2***);

**respText** = (TextView)findViewById(R.id.***respValor***);

**inpSelect** = (Spinner)findViewById(R.id.***inpSelect***);

*//* ***Todo aqui se muestra como crear un componente tipo Select***

*//Arreglo con las opciones*

String [] opciones = {**"Sumar"**, **"Restar"**, **"Multiplicar"**, **"Dividir"**};

*//Debemos declarar un adaptare para poder comunicar el arreglo con el componente*

ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, **R.layout.*spinner\_item\_view***, opciones);

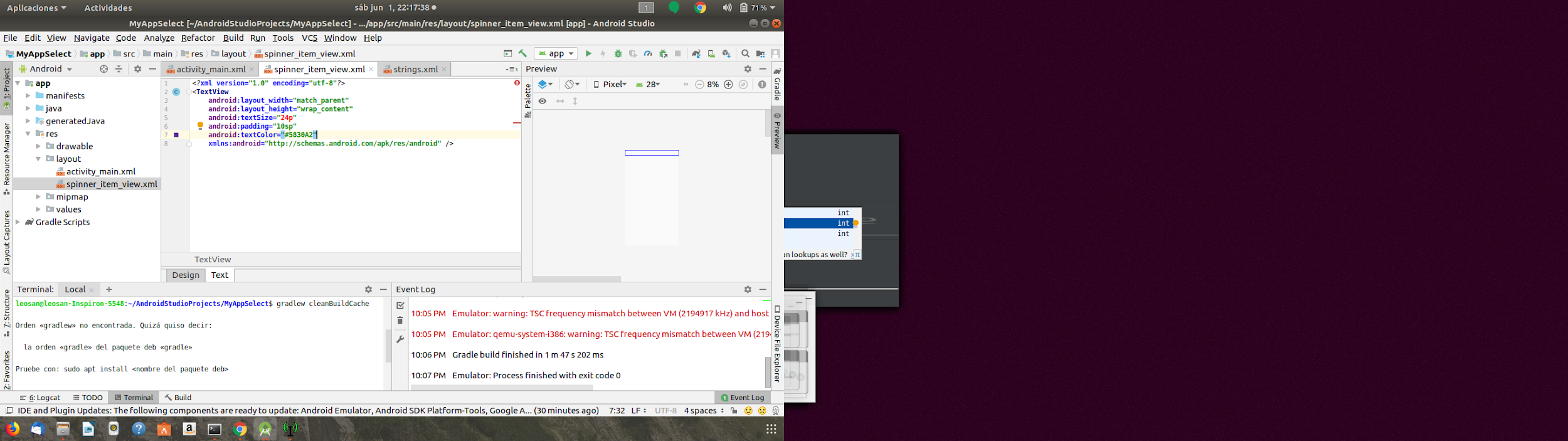
*// Aqui agrego el objeto tipo adapte al componente*

**inpSelect**.setAdapter(adapter);

}

Parte muy importante podemos crear nuestros layout y asignarlos a la parte lógica

Aqui se hace el layout



---- Curso Android Parte 20 -- LIST VIEW ---

Para esta práctica aprenderemos hacer un list view dejo codigo fuente.

Notas:

Para esta versiòn de android studio esta el LisView en [ Legacy. ]

**package** com.example.applisview;

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

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.AdapterView;

**import** android.widget.ArrayAdapter;

**import** android.widget.ListView;

**import** android.widget.TextView;

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

**private** TextView **tv1**;

**private** ListView **lv1**;

**private** String **arrNombres** [] = {**"Samuel"**, **"Valentina"**, **"Jose"**, **"Maria"** };

**private** String **arrEdad** [] = {**"18"**, **"25"**, **"20"**, **"18"**};

@Override

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

**super**.onCreate(savedInstanceState);

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

**tv1** = (TextView)findViewById(R.id.***inpResp***);

**lv1** = (ListView)findViewById(R.id.***inpList***);

*// Detalle importante*

ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, R.layout.***list\_item\_leo***, **arrNombres**);

**lv1**.setAdapter(adapter);

*//Recibe clases anonima tema avanzado de android*

**lv1**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {

@Override

**public void** onItemClick(AdapterView<?> AdapterView, View view, **int** position, **long** id) {

**tv1**.setText(**"La edad de "**+**lv1**.getItemAtPosition(position) + **"es "** + **arrEdad**[position]+**" años"**);

}

});

}

}

-----------------------------------------------------------------------------------------------------------------

*<?***xml version="1.0" encoding="utf-8"***?>*

<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

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

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

**tools:context=".MainActivity"**>

<**TextView**

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

**android:layout\_width="367dp"**

**android:layout\_height="50dp"**

**android:layout\_marginStart="8dp"**

**android:layout\_marginLeft="8dp"**

**android:layout\_marginTop="36dp"**

**android:text="TextView"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"**

**tools:text="@string/textView"** />

<**ListView**

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

**android:layout\_width="364dp"**

**android:layout\_height="269dp"**

**android:layout\_marginStart="16dp"**

**android:layout\_marginLeft="16dp"**

**android:layout\_marginTop="20dp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpResp"**>

</**ListView**>

</**android.support.constraint.ConstraintLayout**>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*<?***xml version="1.0" encoding="utf-8"***?>*

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

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

**android:background="#FF1368BD"**

**android:textSize="24sp"**

**android:padding="10sp"**

**android:textColor="#FFFFFF"**

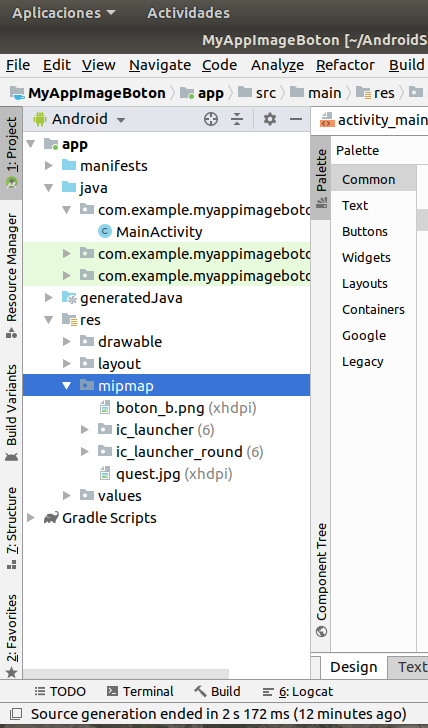
**xmlns:android="http://schemas.android.com/apk/res/android"** />

---- Curso Android Parte 21 -- Boton Image ---

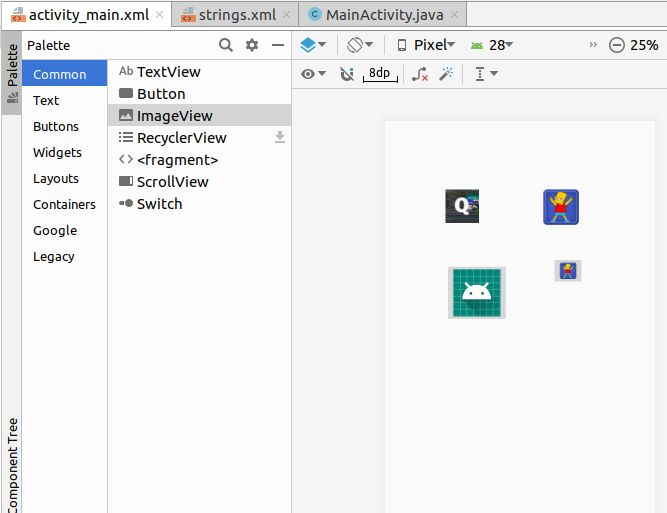
Paso 1: se debe ubicar las imágenes que se van a utilizar y copiar en esta ruta.

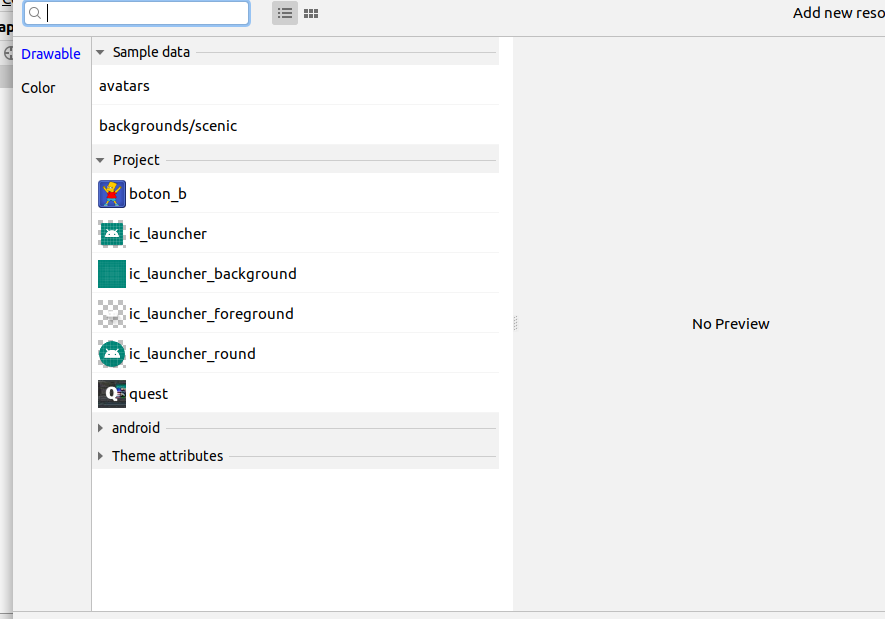
app/res/mipmap

Se pega en esta ruta y se debe dar ok en dos interfaces.

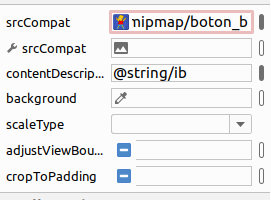


Paso 2: luego seleccionamos el componente





Notas: si hay fallas debes revisar el diseño blue y contentDescription ya que debe tener un texto relacionado

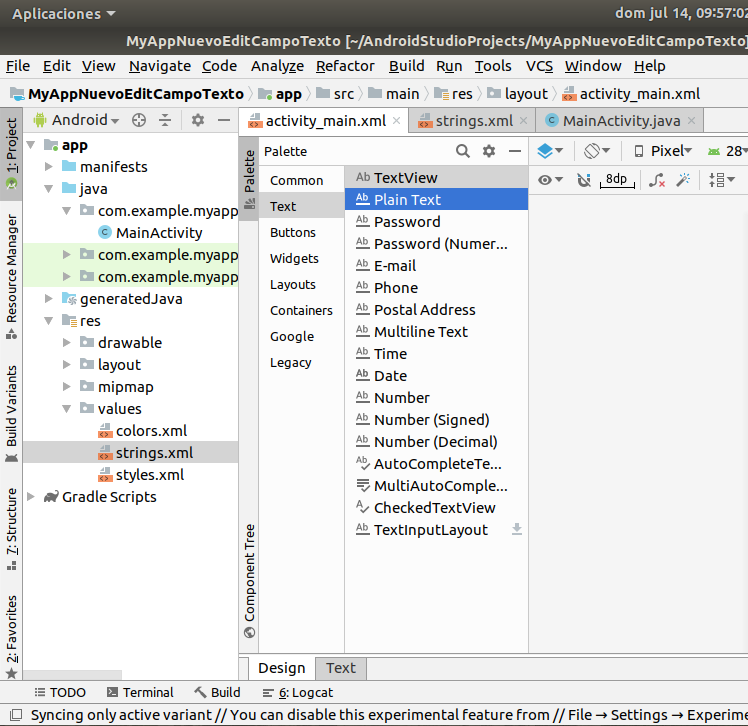


---- Curso Android Parte 22 -- Editar Nuevo Campo de texto ---

En esta capítulo aprenderemos como usar otro editor de texto a diferencia de los anteriores este podra recibir texto.

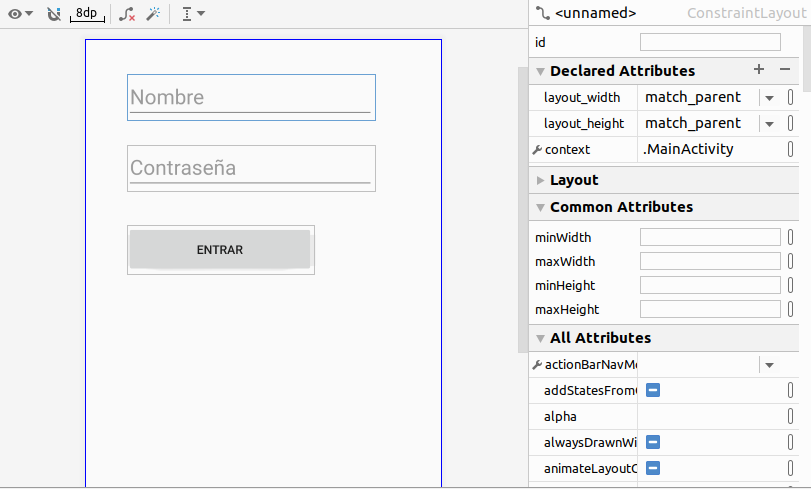
Paso 1: creamos proyecto y creamos una interfaz para hacer la practica.

Paso 2: ubicamos donde están los componentes:



Paso 3: creamos nuestra interfaz para probar.

Dejo Còdigo XML y print de pantalla:



Codigo: Nota recuerda crear su strig value ..

*<?***xml version="1.0" encoding="utf-8"***?>*

<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

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

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

**tools:context=".MainActivity"**>

<**EditText**

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

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

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

**android:layout\_marginStart="48dp"**

**android:layout\_marginLeft="48dp"**

**android:layout\_marginTop="28dp"**

**android:ems="10"**

**android:hint="@string/inpClave"**

**android:inputType="textPassword"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpNombre"** />

<**EditText**

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

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

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

**android:layout\_marginStart="48dp"**

**android:layout\_marginLeft="48dp"**

**android:layout\_marginTop="40dp"**

**android:ems="10"**

**android:hint="@string/inpNombre"**

**android:inputType="textPersonName"**

**android:textSize="24sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"** />

<**Button**

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

**android:layout\_width="216dp"**

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

**android:layout\_marginStart="48dp"**

**android:layout\_marginLeft="48dp"**

**android:layout\_marginTop="40dp"**

**android:onClick="Registrar"**

**android:text="@string/textBoton"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpClave"** />

</**android.support.constraint.ConstraintLayout**>

Paso 4: Dejo còdigo java:

**package** com.example.myappnuevoeditcampotexto;

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

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.EditText;

**import** android.widget.Toast;

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

**private** EditText **etn**, **etp**;

@Override

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

**super**.onCreate(savedInstanceState);

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

**etn** = (EditText)findViewById(R.id.***inpNombre***);

**etp** = (EditText)findViewById(R.id.***inpClave***);

}

**public void** Registrar(View view){

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

String clave = **etp**.getText().toString();

}

**public void** Validar(String valNombre, String valClave){

**if** (valNombre.length() == 0 ){

Toast.*makeText*(**this**, **"Debes Ingresar un nombre"**, Toast.***LENGTH\_LONG***).show();

}

**if** (valClave.length() == 0 ){

Toast.*makeText*(**this**, **"Debes Ingresar una clave"**, Toast.***LENGTH\_LONG***).show();

}

**if** (valNombre.length() != 0 && valClave.length() != 0 ){

Toast.*makeText*(**this**, **"Registro en proceso..."**, Toast.***LENGTH\_LONG***).show();

}

}

}

---- Curso Android Parte 23 -- ---

---- Curso Android Parte 24 -- —

---- Curso Android Parte 25 -- —

---- Curso Android Parte 26 -- —

---- Curso Android Parte 27 -- —

---- Curso Android Parte 28 -- App de Bitacora —

---- Curso Android Parte 29 -- Almacenamiento Externo --

Creamos un proyecto nuevo para explicar el funcionamiento

Paso 1: Creamos la interfaz:

*<?***xml version="1.0" encoding="utf-8"***?>*

<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

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

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

**tools:context=".MainActivity"**>

<**EditText**

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

**android:layout\_width="319dp"**

**android:layout\_height="68dp"**

**android:layout\_marginStart="8dp"**

**android:layout\_marginLeft="8dp"**

**android:layout\_marginTop="32dp"**

**android:layout\_marginEnd="64dp"**

**android:layout\_marginRight="64dp"**

**android:ems="10"**

**android:hint="@string/appBuscar"**

**android:inputType="textPersonName"**

**android:textColor="#4CAF50"**

**android:textSize="18sp"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**app:layout\_constraintHorizontal\_bias="1.0"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"** />

<**EditText**

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

**android:layout\_width="372dp"**

**android:layout\_height="212dp"**

**android:layout\_marginStart="16dp"**

**android:layout\_marginLeft="16dp"**

**android:layout\_marginTop="16dp"**

**android:layout\_marginEnd="8dp"**

**android:layout\_marginRight="8dp"**

**android:background="#9C27B0"**

**android:ems="10"**

**android:gravity="start|top"**

**android:hint="@string/appValores"**

**android:inputType="textMultiLine"**

**android:textColor="#F5F2F2"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**app:layout\_constraintHorizontal\_bias="0.0"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpBuscar"** />

<**Button**

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

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

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

**android:layout\_marginStart="88dp"**

**android:layout\_marginLeft="88dp"**

**android:layout\_marginTop="76dp"**

**android:onClick="metodoGuardar"**

**android:text="@string/btnGuardar"**

**android:textSize="18sp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpDatos"** />

<**Button**

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

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

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

**android:layout\_marginStart="8dp"**

**android:layout\_marginLeft="8dp"**

**android:layout\_marginTop="72dp"**

**android:layout\_marginEnd="48dp"**

**android:layout\_marginRight="48dp"**

**android:onClick="metodoConsultar"**

**android:text="@string/btnConsultar"**

**android:textSize="18sp"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**app:layout\_constraintStart\_toEndOf="@+id/btnGuardar"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpDatos"** />

</**android.support.constraint.ConstraintLayout**>

Paso 2: Creamos los permisos.

Los permisos estan ubicados en el siguiente path:

/home/leosan/AndroidStudioProjects/MyAppAlmacenExterno/app/src/main/AndroidManifest.xml

Se agrega esta línea de còdigo que da el permiso:

<**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"**/>

El còdigo final queda asi:

*<?***xml version="1.0" encoding="utf-8"***?>*

<**manifest xmlns:android="http://schemas.android.com/apk/res/android"**

**package="com.example.myappalmacenexterno"**>

<**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"**/>

<**application**

**android:allowBackup="true"**

**android:icon="@mipmap/ic\_launcher"**

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

**android:roundIcon="@mipmap/ic\_launcher\_round"**

**android:supportsRtl="true"**

**android:theme="@style/AppTheme"**>

<**activity android:name=".MainActivity"**>

<**intent-filter**>

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

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

</**intent-filter**>

</**activity**>

</**application**>

</**manifest**>

Paso 3: Creamos la interactividad. Aqui como leemos y escribimos un Archivo.

**package** com.example.myappalmacenexterno;

**import** android.app.Activity;

**import** android.os.Environment;

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

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.EditText;

**import** android.widget.Toast;

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.io.OutputStreamWriter;

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

EditText **inpBuscar**, **inpDatos**;

@Override

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

**super**.onCreate(savedInstanceState);

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

**inpBuscar** = (EditText)findViewById(R.id.***inpBuscar***);

**inpDatos** = (EditText)findViewById(R.id.***inpDatos***);

}

**public void** metodoGuardar(View view){

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

String contenido = **inpDatos**.getText().toString();

**try** {

File tarjetaSD = Environment.*getExternalStorageDirectory*();

Toast.*makeText*(**this**, tarjetaSD.getPath(), Toast.***LENGTH\_SHORT*** ).show();

File rutaArchivo = **new** File(tarjetaSD.getPath(), nombre);

OutputStreamWriter crearArchivo = **new** OutputStreamWriter( openFileOutput(nombre, Activity.***MODE\_PRIVATE***) );

crearArchivo.write(contenido);

crearArchivo.flush();

crearArchivo.close();

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

**inpBuscar**.setText(**" "**);

**inpDatos**.setText(**" "**);

}**catch** (IOException e){

Toast.*makeText*(**this**, **"No se pudo guardar"**, Toast.***LENGTH\_SHORT***).show();

}

}

**public void** metodoConsultar(View view){

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

**try** {

File tarjetaSD = Environment.*getExternalStorageDirectory*();

File rutaArchivo = **new** File(tarjetaSD.getPath(), nombre);

*//Abre el archivo*

InputStreamReader abrirArchivo = **new** InputStreamReader(openFileInput(nombre));

*//leermos el archivo*

BufferedReader leerArchivo = **new** BufferedReader(abrirArchivo);

String ln = leerArchivo.readLine();

String contenidoCompleto = **""**;

**while**(ln != **null**){

contenidoCompleto = contenidoCompleto + ln + **"\n"**;

ln = leerArchivo.readLine();

}

leerArchivo.close();

abrirArchivo.close();

**inpDatos**.setText(contenidoCompleto);

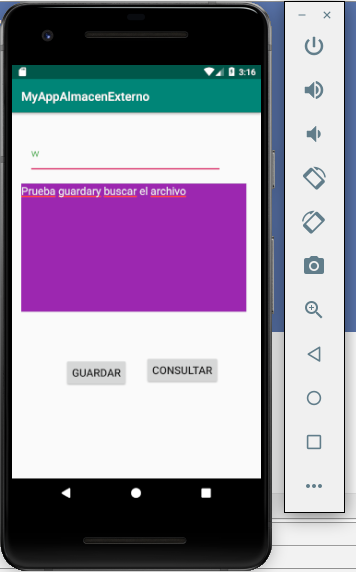
}**catch** (IOException e){

Toast.*makeText*(**this**,**"Error al leer "**, Toast.***LENGTH\_SHORT***).show();

}

}

}



---- Curso Android Parte 30, 31, 32, 33 --Bases de datos ---

Aquí el tópico nos muestra cómo hacer las siguientes Actividades:

* crear una base de datos. SQL
* conectarnos a una Bases de datos SQL
* Registrar datos en una BD.
* Buscar datos en una BD.
* Eliminar datos Bd.
* Modificar datos en una Bd.
* Validar campos.

Teoria:

Se usa en bases de datos SQLite,

SQLite es una biblioteca escrita en leguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración. El código de SQLite es de dominio público y libre para cualquier uso, ya sea comercial o privado. Actualmente es utilizado en gran cantidad de aplicaciones incluyendo algunas desarrolladas como proyectos de alto nivel.

La clase que nos permite usar las funciones de bases de datos es llamada

SQLiteOpenHelper. podremos ver sus metodos aqui.

<https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper>

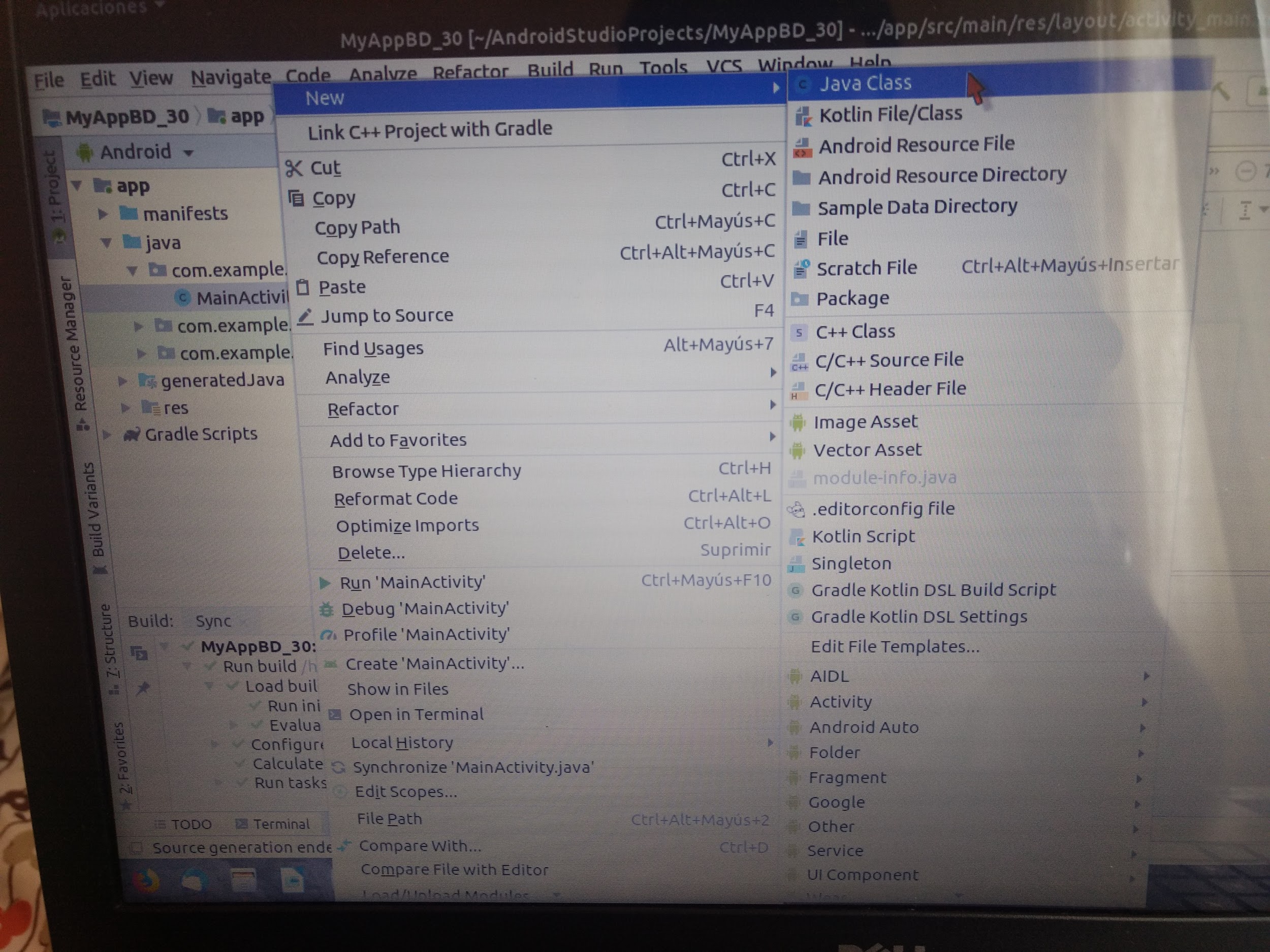
Ejemplo: este ejemplo explica de manera dinámica todas las operaciones básicas que se pueden hacer en la base de datos.

Paso 1: Creamos nuestros proyecto para este caso para el ejemplo se denominará

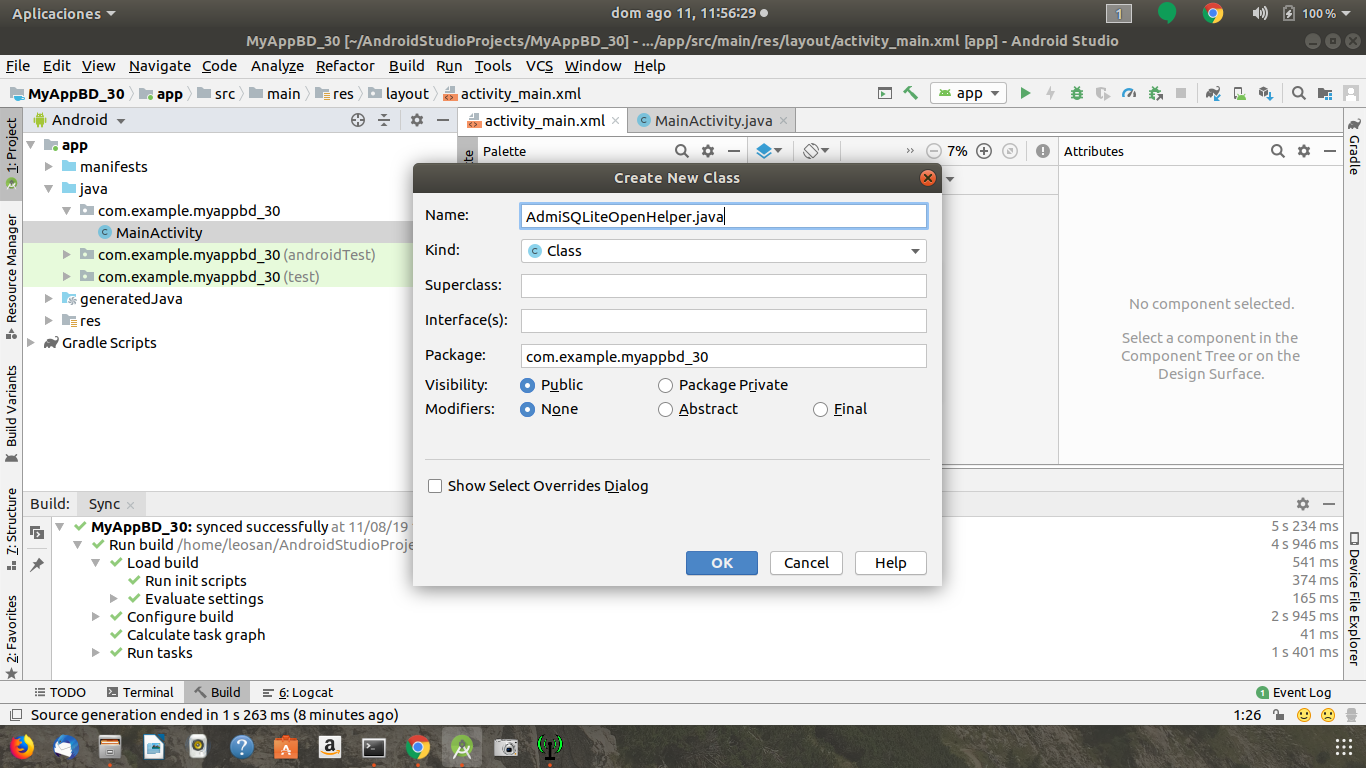
com.example.myappbd\_30

Paso 2: Creamos el entorno para la bases de datos, creamos una clase AdmiSQLiteOpenHelper.java, como hacemos esto nos ubicamos en el proyecto clic derecho y creamos una clase imagen adjunta nos indicará.

Clic derecho en: MainActivity/New/JavaClass



Luego nos abrirá otra interfaz donde debemos colocar el nombre de nuestra nueva clase,



clic en Ok.

Esto generará una clase que nos ayudará a controlar y admnistrar todo lo relacionado con nuestra bases de datos, ahora es importante resaltar que al crearla debemos importar las librerías correspondientes y generar su herencia.

Paso 3: importamos

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

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

Paso 4: extendemos la herencia

**public class** AdmiSQLiteOpenHelper **extends** SQLiteOpenHelper {

Paso 5: debemos incorporar los métodos abstractos oncreate() y onUpgrade() estos métodos nos permite manipular la bases de datos pero debemos adaptarlas para nuestros proyectos.

@Override

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

}

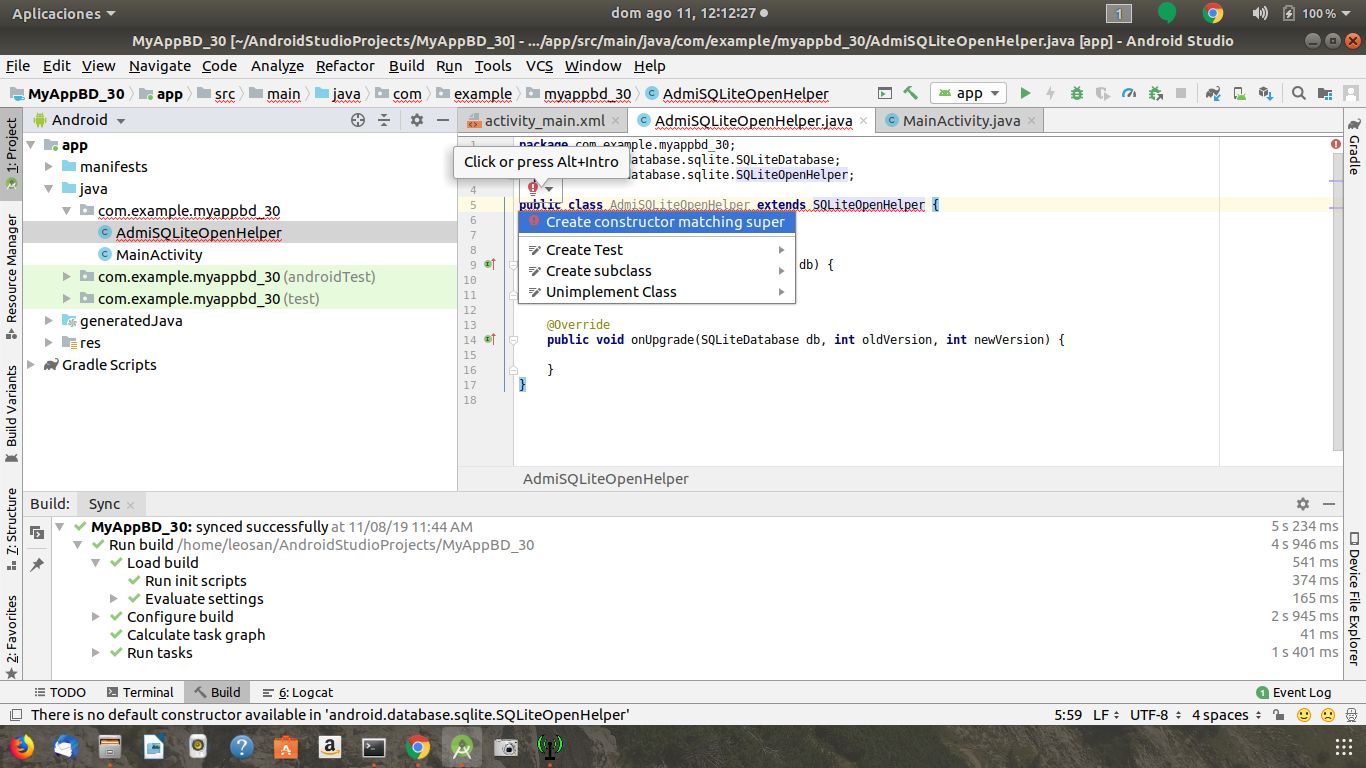
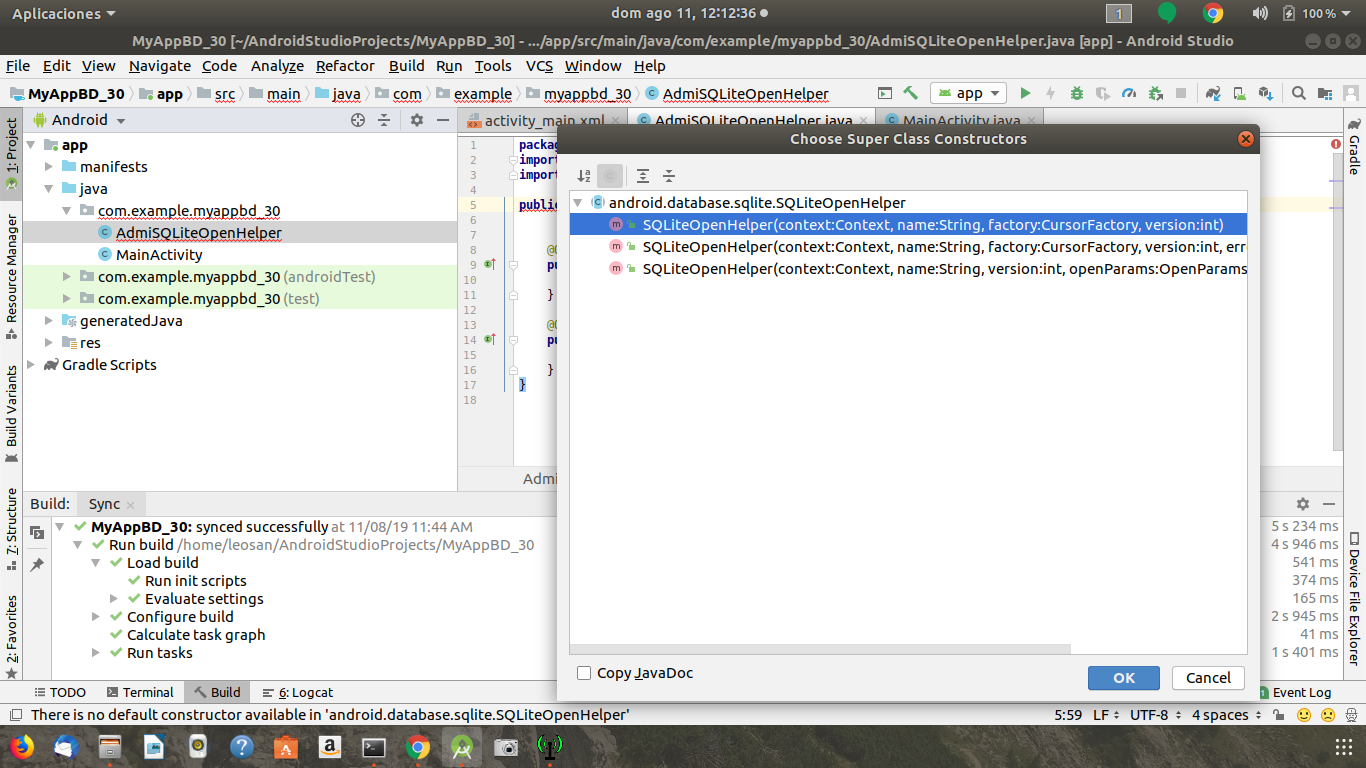
@Override

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

}



Paso 6: Creamos nuestro constructor



Paso 7: al finalizar la clase creamos los métodos para crear nuestra BD, resalto y colocó en letras grandes como crear una BD.

Codigo quedaria algo asi:

**package** com.example.myappbd\_30;

**import** android.content.Context;

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

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

**public class** AdmiSQLiteOpenHelper **extends** SQLiteOpenHelper {

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

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

}

@Override

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

**BasesDeDatos.execSQL("Create table t\_articulos(codigo int primary Key, nomArticulo text, precio real)");**

}

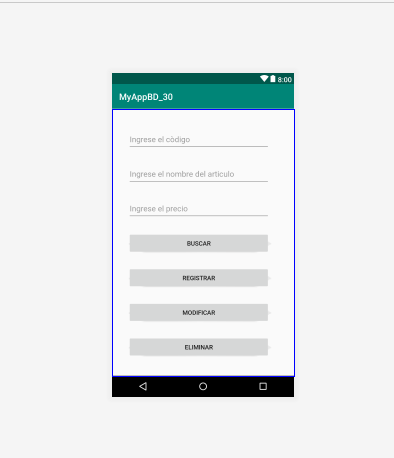
@Override

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

}

}

Paso 8: Creamos nuestra interfaz quedando algo asi:



Código Interfaz:

*<?***xml version="1.0" encoding="utf-8"***?>*

<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

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

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

**tools:context=".MainActivity"**>

<**TextView**

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

**android:layout\_width="320dp"**

**android:layout\_height="50dp"**

**android:layout\_marginStart="36dp"**

**android:layout\_marginLeft="36dp"**

**android:layout\_marginTop="16dp"**

**android:text="@string/text\_Titulo"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"** />

<**EditText**

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

**android:layout\_width="320dp"**

**android:layout\_height="50dp"**

**android:layout\_marginStart="36dp"**

**android:layout\_marginLeft="36dp"**

**android:layout\_marginTop="12dp"**

**android:ems="10"**

**android:hint="@string/text\_Codigo"**

**android:inputType="number"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/textResp"** />

<**EditText**

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

**android:layout\_width="320dp"**

**android:layout\_height="50dp"**

**android:layout\_marginStart="36dp"**

**android:layout\_marginLeft="36dp"**

**android:layout\_marginTop="12dp"**

**android:ems="10"**

**android:hint="@string/text\_Nombre"**

**android:inputType="textPersonName"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpCodigo"** />

<**EditText**

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

**android:layout\_width="320dp"**

**android:layout\_height="50dp"**

**android:layout\_marginStart="36dp"**

**android:layout\_marginLeft="36dp"**

**android:layout\_marginTop="12dp"**

**android:ems="10"**

**android:hint="@string/text\_Precio"**

**android:inputType="numberDecimal"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpNombre"** />

<**Button**

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

**android:layout\_width="320dp"**

**android:layout\_height="50dp"**

**android:layout\_marginStart="36dp"**

**android:layout\_marginLeft="36dp"**

**android:layout\_marginTop="12dp"**

**android:onClick="metodoBuscar"**

**android:text="@string/text\_btnBuscar"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/inpPrecio"** />

<**Button**

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

**android:layout\_width="320dp"**

**android:layout\_height="50dp"**

**android:layout\_marginStart="36dp"**

**android:layout\_marginLeft="36dp"**

**android:layout\_marginTop="12dp"**

**android:onClick="metodoRegistrar"**

**android:text="@string/text\_btnRegistrar"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/btnBuscar"** />

<**Button**

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

**android:layout\_width="320dp"**

**android:layout\_height="50dp"**

**android:layout\_marginStart="36dp"**

**android:layout\_marginLeft="36dp"**

**android:layout\_marginTop="12dp"**

**android:onClick="metodoModificar"**

**android:text="@string/text\_btnModificar"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/btnRegistrar"** />

<**Button**

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

**android:layout\_width="320dp"**

**android:layout\_height="50dp"**

**android:layout\_marginStart="36dp"**

**android:layout\_marginLeft="36dp"**

**android:layout\_marginTop="12dp"**

**android:onClick="metodoEliminar"**

**android:text="@string/text\_btnEliminar"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/btnModificar"** />

</**android.support.constraint.ConstraintLayout**>

Paso 8: Creamos la interactividad MainActivity

Nota: aquì esta toda la lógica de la interactividad:

* Como registrar
* como modificar
* cómo eliminar
* cómo buscar

Código ejemplo:

**package** com.example.myappbd\_30;

**import** android.content.ContentValues;

**import** android.database.Cursor;

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

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

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.EditText;

**import** android.widget.TextView;

**import** android.widget.Toast;

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

**private** EditText **inpCodigo**, **inpNombre**,**inpPrecio**;

**private** TextView **inpResp**;

@Override

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

**super**.onCreate(savedInstanceState);

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

*//Creamos la relacion con la interfaz*

**inpCodigo** = (EditText)findViewById(R.id.***inpCodigo***);

**inpNombre** = (EditText)findViewById(R.id.***inpNombre***);

**inpPrecio** = (EditText)findViewById(R.id.***inpPrecio***);

**inpResp** = (TextView)findViewById(R.id.***textResp***);

}

*/\**

*\* Description : Metodo que permite registrar los valores en la bases de datos.*

*\**

*\* \*/*

**public void** metodoRegistrar(View view){

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

*// Abre la base de datos en modo lectura y escritura*

SQLiteDatabase BasesDeDatos = admin.getWritableDatabase();

String codigo = **inpCodigo**.getText().toString();

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

String precio = **inpPrecio**.getText().toString();

**boolean** valCodigo, valNombre, valPrecio;

valCodigo = validaCampos(codigo, **"Registrar"**, **inpResp**);

valNombre = validaCampos(nombre, **"Registrar"**, **inpResp**);

valPrecio = validaCampos(precio, **"Registrar"**, **inpResp**);

**if** (valCodigo == **true** && valNombre == **true** && valPrecio == **true**){

ContentValues registro = **new** ContentValues(); *// Aqui intaciamos el insert para la tabla*

registro.put(**"codigo"**, codigo); *// Aqui asociamos los atributos de la tabla con los valores de los campos (Recuerda el campo de la tabla debe ser igual aqui)*

registro.put(**"nomArticulo"**, nombre); *// Aqui asociamos los atributos de la tabla con los valores de los campos (Recuerda el campo de la tabla debe ser igual aqui)*

registro.put(**"precio"**, precio); *// Aqui asociamos los atributos de la tabla con los valores de los campos (Recuerda el campo de la tabla debe ser igual aqui)*

*//Conectamos con la base datos insertamos.*

BasesDeDatos.insert(**"t\_articulos"**, **null**, registro);

BasesDeDatos.close();

*//limpiamos los valores de los campos*

**inpCodigo**.setText(**" "**);

**inpNombre**.setText(**" "**);

**inpPrecio**.setText(**" "**);

**inpResp**.setText(**"Registro Exitoso!"**);

Toast.*makeText*(**this**, **"Registro Exitoso!"**, Toast.***LENGTH\_SHORT***).show();

}

}*//Fin del metodo registrar en tabla*

*/\**

*\* Description : Metodo que permite consultar*

*\**

*\* \*/*

**public void** metodoBuscar(View view){

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

SQLiteDatabase BaseDedatos = admin.getReadableDatabase();

String codigo = **inpCodigo**.getText().toString();

Boolean valCodigo;

valCodigo = validaCampos(codigo,**"Buscar"**, **inpResp**);

**if** (valCodigo == **true**){

Cursor fila = BaseDedatos.rawQuery(**"SELECT nomArticulo, precio FROM t\_articulos WHERE codigo ="** + codigo, **null**);

**if** ( fila.moveToFirst() ){*//Muestra los valores encontrados en la consulta*

**inpNombre**.setText(fila.getString(0)); *// Siempre debe ponerle cero //* ***todo nota***

**inpPrecio**.setText(fila.getString(1)); *// Siempre debe ponerle cero //* ***todo nota***

}**else**{

Toast.*makeText*(**this**, **"No hay valores encontrados con este código ->"**+codigo , Toast.***LENGTH\_SHORT***).show();

**inpResp**.setText(**"No hay valores encontrados"**);

*//limpiamos los valores de los campos*

**inpCodigo**.setText(**" "**);

**inpNombre**.setText(**" "**);

**inpPrecio**.setText(**" "**);

}

}

BaseDedatos.close();

}*//fin del metodo Buscar*

*/\**

*\* Description : Metodo que ermite validar un campo que no este vacio*

*\**

*\* \*/*

**public boolean** validaCampos(String valor,String Tipo, TextView inpResp){

**boolean** valida = **true**; *//true ==* ***todo Ok, todo bien***

String mensaje = **" "**;

*//Configuro el tipo de mensaje*

**if** (Tipo == **"Registrar"**){ mensaje = **" Para registrar un articulo, todos los campos son obligatorios"**; }

**if** (Tipo == **"Buscar"**){ mensaje = **" Debe ingresar un código para buscar "**; }

**if** (Tipo == **"Eliminar"**){ mensaje = **" Debe ingresar un código para Eliminar un registro"**; }

**if** (Tipo == **"Modificar"**){ mensaje = **" Debe ingresar valores para modificar"**; }

*//Valido si el campo esta vacio*

**if** ( valor.isEmpty() ){

Toast.*makeText*(**this**, mensaje, Toast.***LENGTH\_SHORT***).show();

inpResp.setText(mensaje);

valida = **false**;

}

**return** valida;

}

*/\**

*\* Description : Metodo que permite eliminar*

*\**

*\* \*/*

**public void** metodoEliminar(View view){

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

SQLiteDatabase BaseDedatos = admin.getReadableDatabase();*// abre la base de datos en modo lectura y escritura*

String codigo = **inpCodigo**.getText().toString();

Boolean valCodigo;

valCodigo = validaCampos(codigo,**"Eliminar"**, **inpResp**);

**if** (valCodigo == **true**){

**int** cantidad = BaseDedatos.delete(**"t\_articulos"**, **"codigo="** + codigo, **null**);

**if** ( cantidad > 0 ){*//si Encontro algo muestra la cantidad de elementos eliminados*

*//limpiamos los valores de los campos*

**inpCodigo**.setText(**" "**);

**inpNombre**.setText(**" "**);

**inpPrecio**.setText(**" "**);

**inpResp**.setText(**"Articulo Eliminado exitosamente"**);

}**else**{

Toast.*makeText*(**this**, **"No hay valores encontrados"** , Toast.***LENGTH\_SHORT***).show();

**inpResp**.setText(**"No hay valores encontrados"**);

}

}

BaseDedatos.close();*// hay que cerrar cada vez que se abre*

}*//fin del metodo Eliminar*

*/\**

*\* Description : Metodo que permite eliminar*

*\**

*\* \*/*

**public void** metodoModificar(View view){

*/\* Inicio para conectar una bases de datos, Nota: debemos crear la clase AdminSqliteOpenHelper* ***todo*** *\*/*

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

SQLiteDatabase BaseDedatos = admin.getReadableDatabase();*// abre la base de datos en modo lectura y escritura*

String codigo = **inpCodigo**.getText().toString();

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

String precio = **inpPrecio**.getText().toString();

**boolean** valCodigo, valNombre, valPrecio;

valCodigo = validaCampos(codigo, **"Modificar"**, **inpResp**);

valNombre = validaCampos(nombre, **"Modificar"**, **inpResp**);

valPrecio = validaCampos(precio, **"Modificar"**, **inpResp**);

**if** (valCodigo == **true** && valNombre == **true** && valPrecio == **true**){

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

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

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

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

*//Metodo para actualizar recibe como parametro (NombreBd, Arreglo con los valores para modificar, condicion, null -> Obligatorio)*

**int** cantidad = BaseDedatos.update(**"t\_articulos"**, registro, **"codigo="** + codigo, **null**);

BaseDedatos.close();*// hay que cerrar cada vez que se abre*

**if** ( cantidad == 1 ){*//si Encontro algo muestra la cantidad de elementos eliminados*

*//limpiamos los valores de los campos*

**inpCodigo**.setText(**" "**);

**inpNombre**.setText(**" "**);

**inpPrecio**.setText(**" "**);

Toast.*makeText*(**this**, **"Artículo Modificado correctamente"** , Toast.***LENGTH\_SHORT***).show();

**inpResp**.setText(**"Artículo Modificado correctamente"**);

}**else**{

Toast.*makeText*(**this**, **"No se encontro valores con ese código"** , Toast.***LENGTH\_SHORT***).show();

**inpResp**.setText(**"No se encontro valores con ese código"**);

}

}

}*//fin del metodo Eliminar*

}//fin de la clase

Paso 9: probamos te sugiero que si algo sale mal solo checa esta sección del android studio:

