



**UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS
PROGRAMACION PARA DISPOSITIVOS MOVILES
PDM115**

GUIA DE LABORATORIO N° 1a

**PROGRAMACION BASICA SOBRE SMART PHONES
ANDROID(versión 22 Lollipop)**



Objetivos:

- *Conocer el entorno de programación del IDE Android Studio versión 3.3.2 para programación bajo el sistema Android.
- *Hacer una nueva aplicación en Android que utilice los controles: TextView, EditText y Button. Interactuando con la interfaz gráfica mediante métodos y eventos. Además de ejecutar un intent de forma implícita.
- *Ejecutar la aplicación mediante un Dispositivo Virtualizado (AVD) o con un dispositivo activo (Smartphone que utilice android)

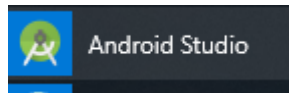
Contenido

Crear una aplicación Básica	1
Reconocimiento del Entorno	7
Creacion de Interfaz Grafica	8
Configuración de los controles.....	9
Configuración de Valores de los controles button y de los mensajes de los Text	10
Agregar recursos(variables string)	11
Asociar controles con los recursos	14
Modificación de Aplicación	16
Anexo 1 Creación de Dispositivo Virtualizado.....	20

Entrada al Android Studio

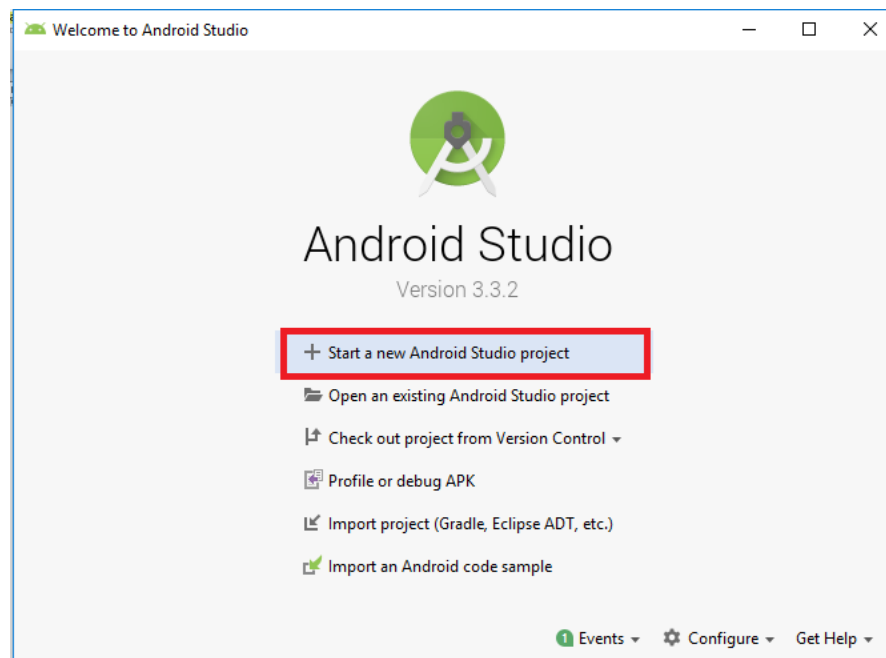
.Inicialice su pc, al ver el menú de clover, seleccione la primera opción correspondiente al sistema Windows.

Entre a la Sesión de PDM115 con clave pdm
En el icono Android Studio o entre a inicio/aplicaciones/Android Studio



Crear una aplicación Básica

Presione clic en “Start a new Android Studio Project”



Seleccione el tipo de Proyecto que desea crear(Basic Activity)

Choose your project

Phone and Tablet Wear OS TV Android Auto Android Things Glass

The grid displays several activity templates. The 'Basic Activity' template is highlighted with a red border. It features a teal header bar with a back arrow and a menu icon, a white main content area, and a yellow circular button with a plus sign at the bottom right. Other templates include 'Empty Activity' (a simple teal header bar), 'Bottom Navigation Activity' (a teal header bar with a bottom navigation bar), and several other activity types with various UI elements like lists, maps, and navigation bars.

Basic Activity

Creates a new basic activity with an app bar.

Previous

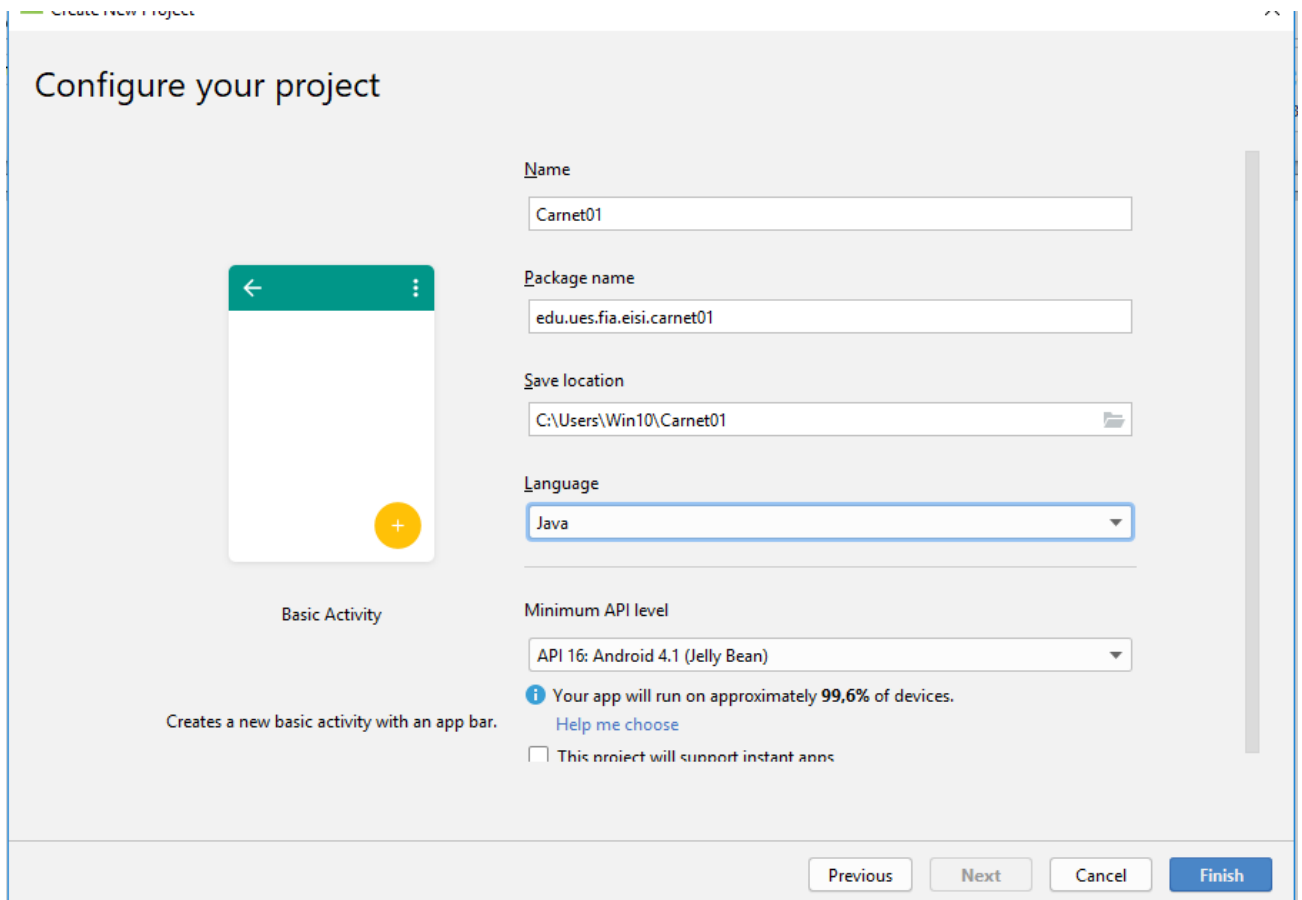
Next

Cancel

Finish

Definir parámetros

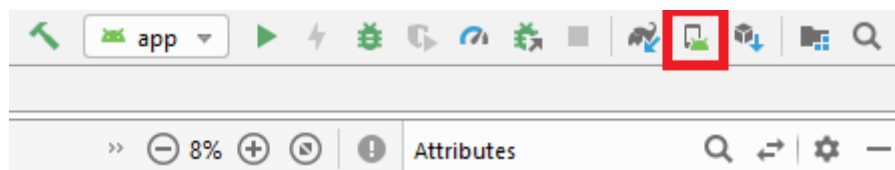
Seleccione la versión mínima en la que correrá su app(API 16) y otros parametros



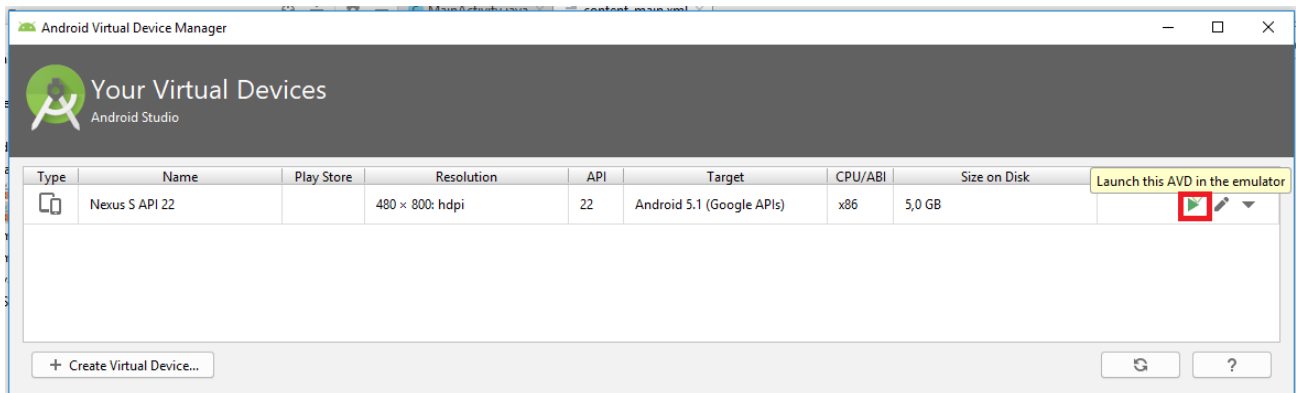
Como nombre de su aplicación ponga su carnet, defina nombre del dominio como se muestra y presione Siguiente(Finish)

Poner en Ejecución el Dispositivo Virtual Existente

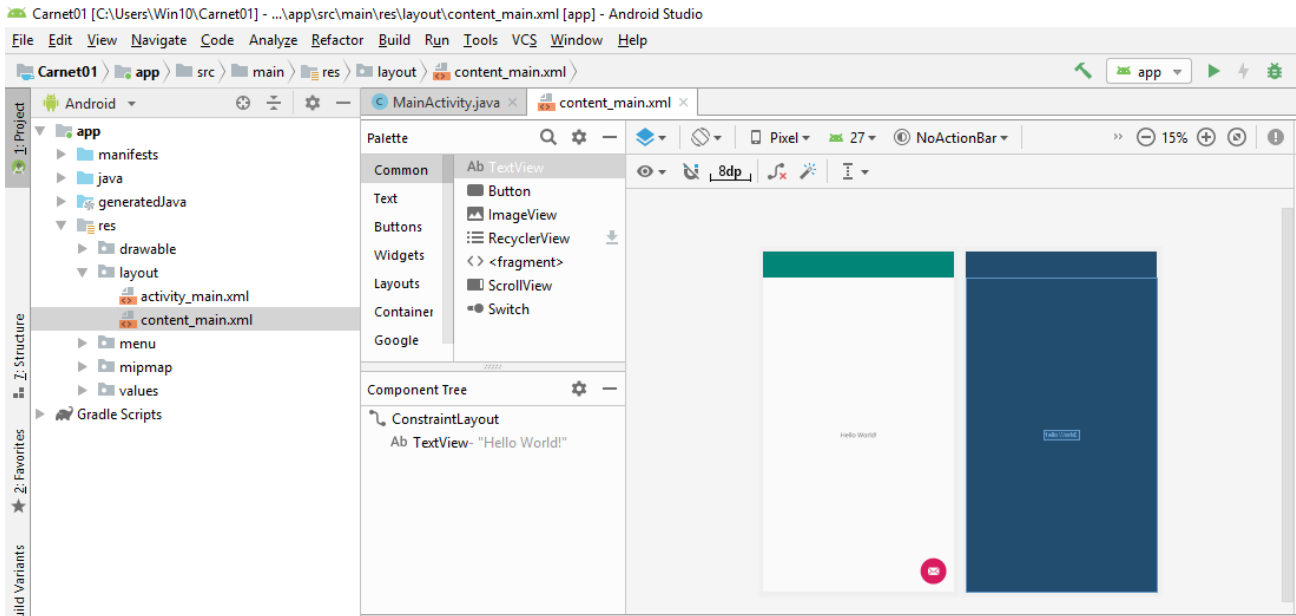
Se carga el Dispositivo virtual para ejecutar nuestro programa más adelante.



Se buscara un Nexus S previamente instalado, si ud desea uno diferente deberá crearlo (Create Virtual Device)

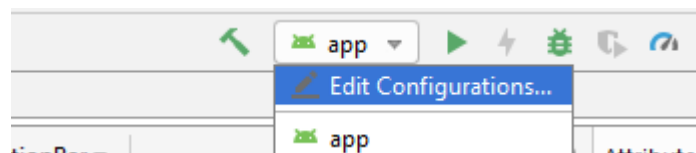


Con lo anterior, deberá aparecer la imagen como se muestra

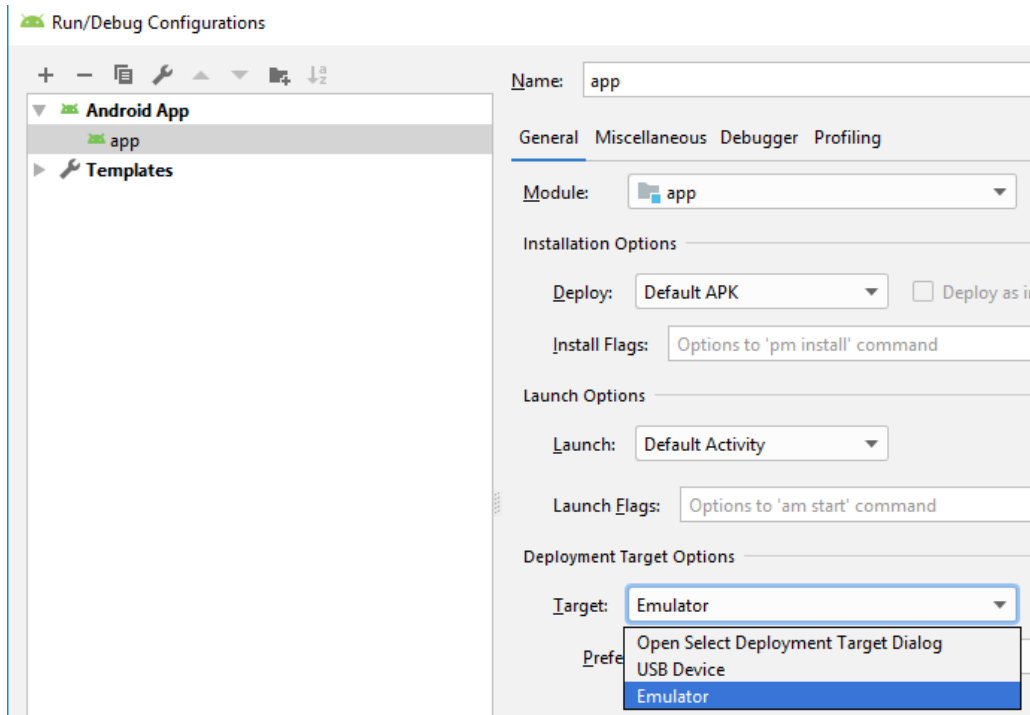


Ejecute la aplicación Holamundo que ya tiene elaborada anteriormente.

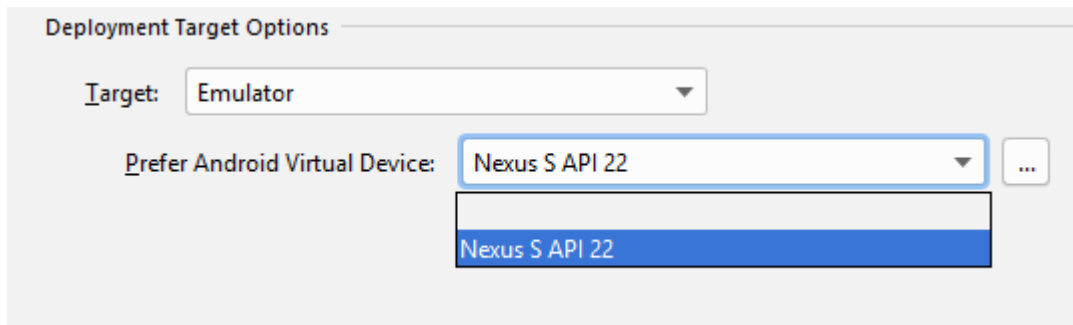
Primero definir el dispositivo emulado que se creó como el que se utilizara por default en el proyecto, presione clic en app, Edit configurations..



Presione clic en el combo de selección target(destino) y seleccione **Emulator**

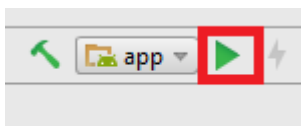


Seleccione su dispositivo..

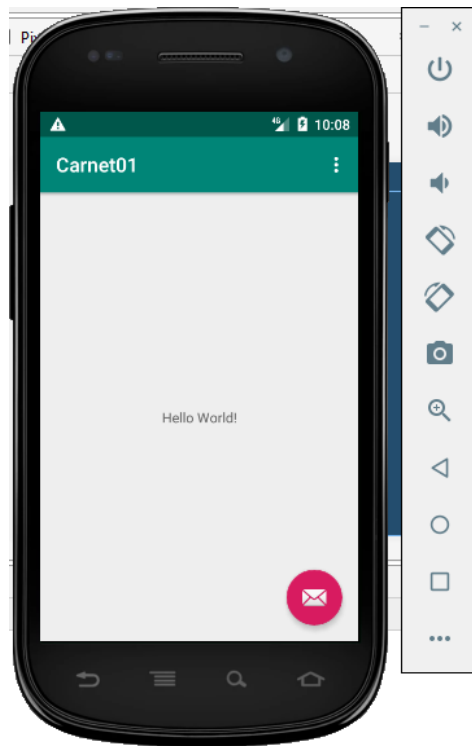


Luego presione OK

Presione clic en run app o shift+F10

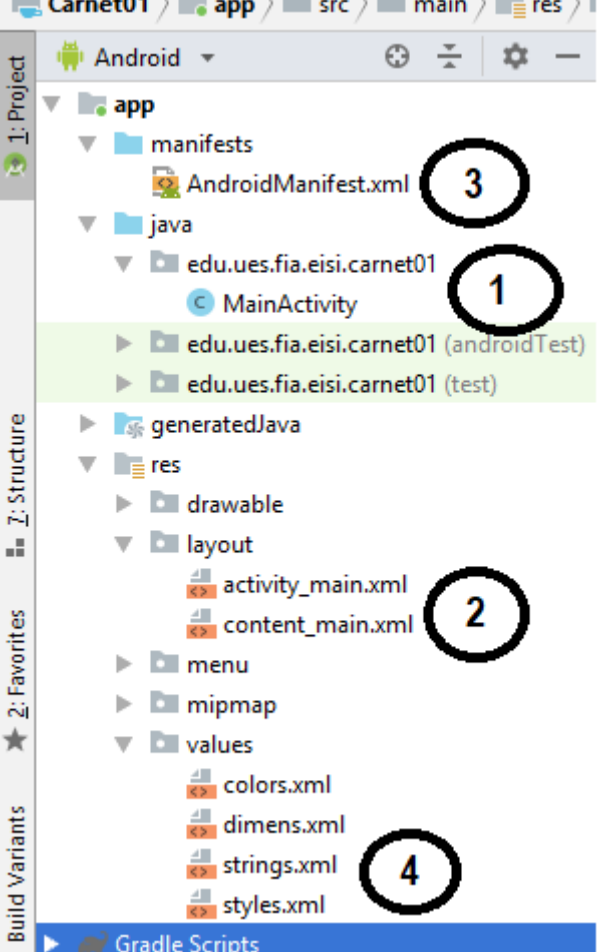


Con lo anterior se cargara la app en el emulador como se muestra.



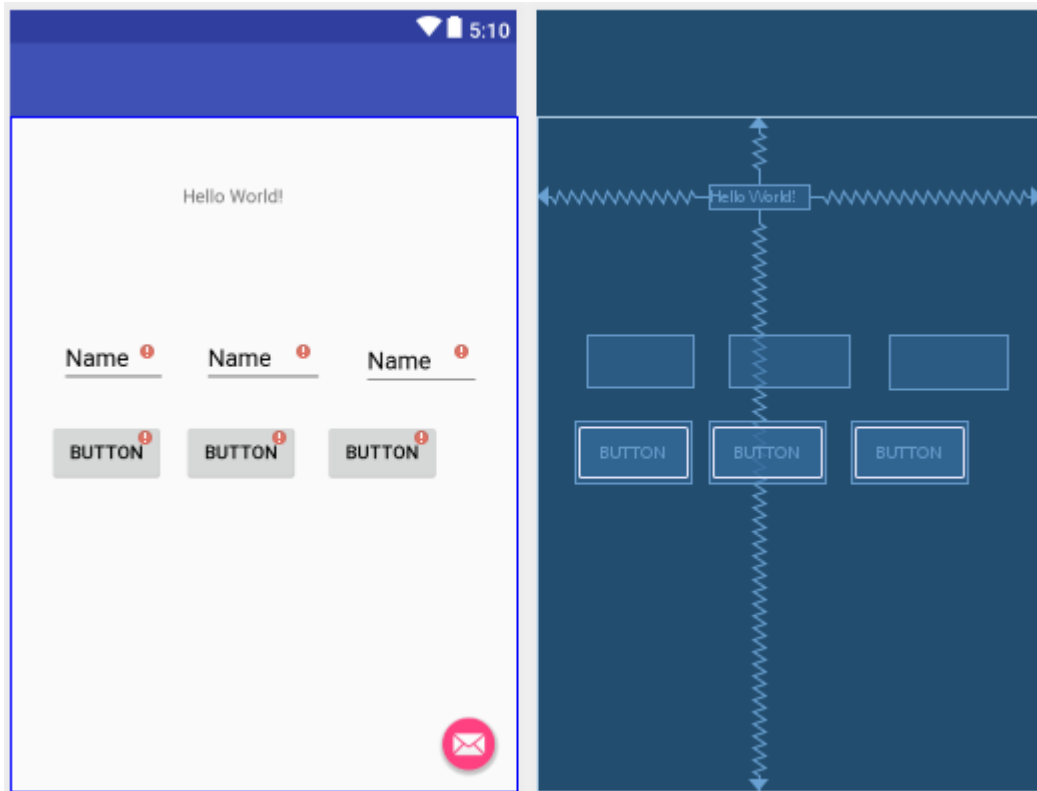
Nota: si no le corre en este momento, infórmelo al instructor para revisar su problema.

Reconocimiento del Entorno

Arbol de objetos de aplicación en Android	Descripcion
	<p>Se trabajara en esta guia tres elementos, Programa(archivo con extension java) en MainActivity.java(1), Interfaz grafica activity_main.xml(2) y un archivo de configuracion de algunos permisos(de ser necesario) llamado AndroidManifest.xml(3) Se utilizara ademas un archivo de recursos llamado string.xml(4) para definir variables utiles para su interfaz grafica.</p>

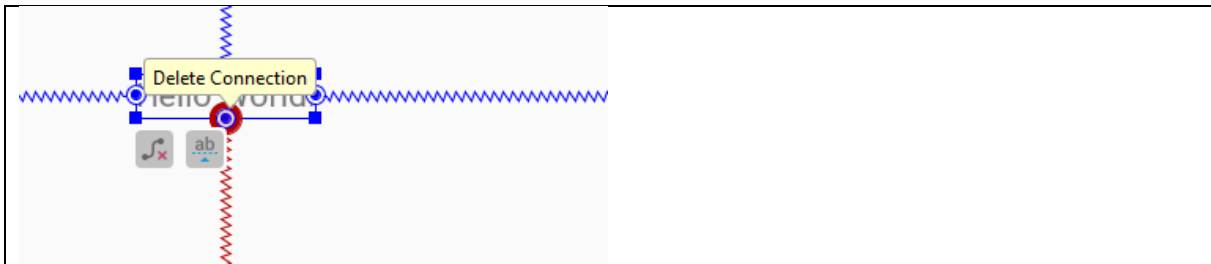
Creacion de Interfaz Grafica

Indicaciones: Hacer una interfaz que contenga ademas del textview de hello world(etiqueta), cuatro botones(button). Arrastrando el objeto desde la paleta hasta el editor grafico de la interfaz(graphical layout) como se muestra. Luego arrastraremos tres objetos Plain text tipo number(3)

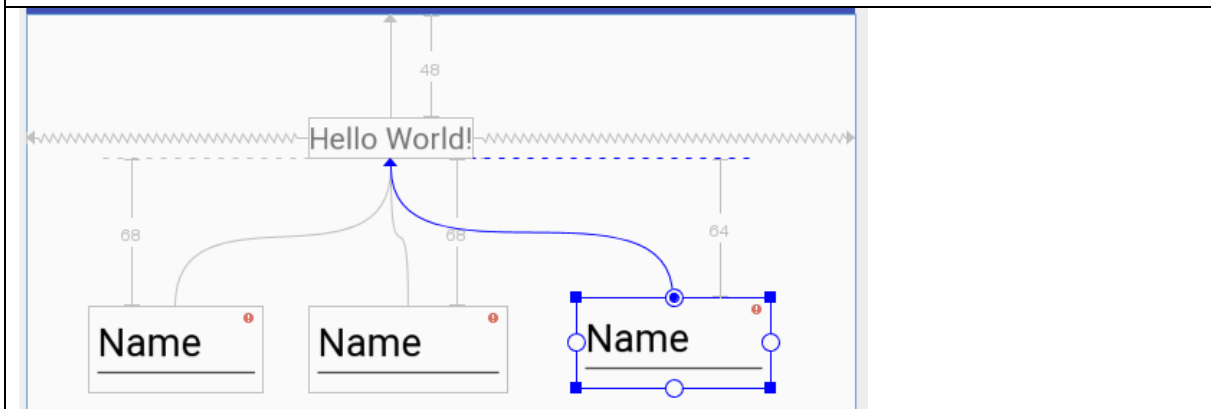


Configuración de los controles

Con lo anterior quedaran así los controles

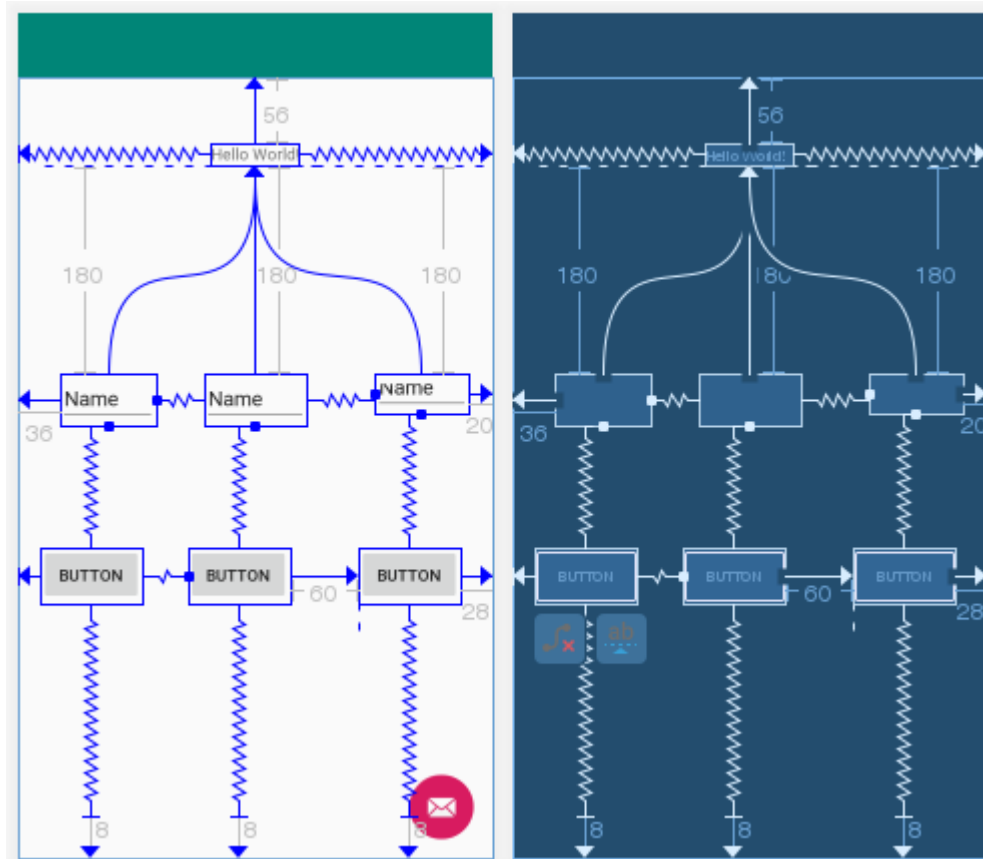


Eliminar conexión de parte inferior



Enlazar parte superior de Plain Texts hacia parte inferior de la EditView(Hello World)

Haga las demás conexiones como se muestra



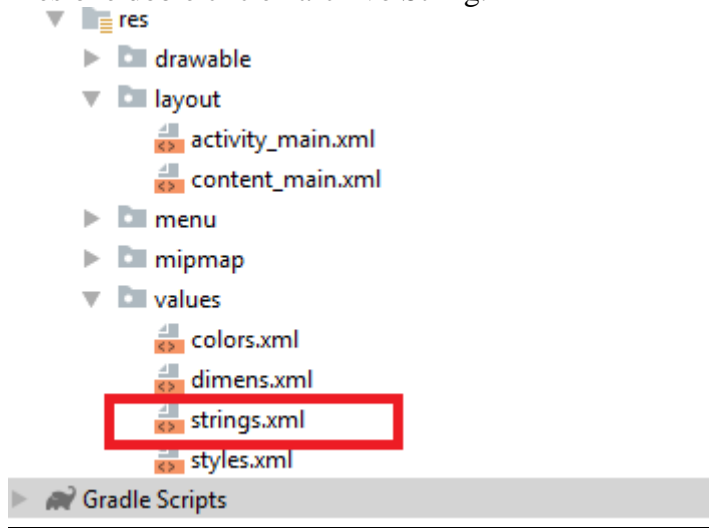
Configuración de Valores de los controles button y de los mensajes de los Text

Primero tendremos que modificar el archivo de recursos string.xml y luego cambiar la propiedad text o hint en los controles para direccionar el contenido hacia esos recursos.

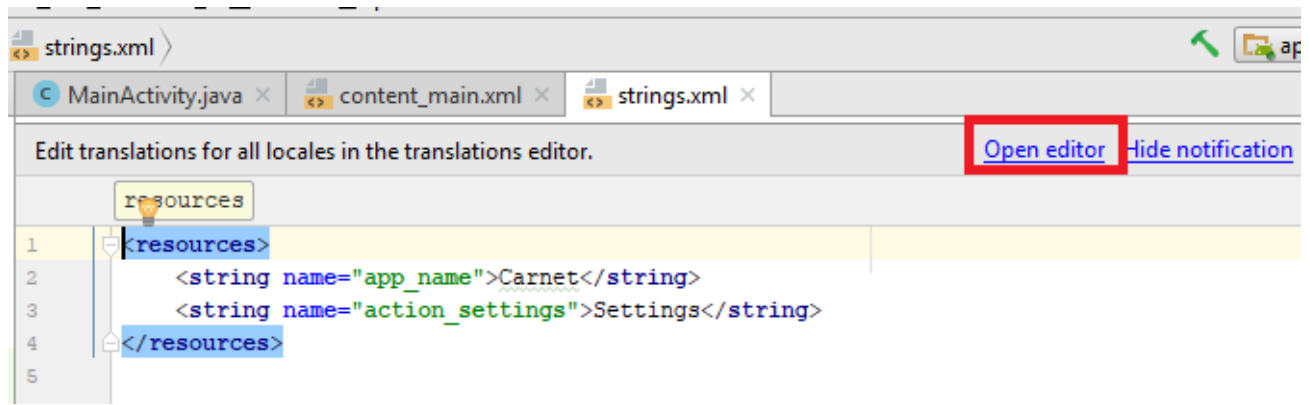
Agregar recursos(variables string)

Primera forma

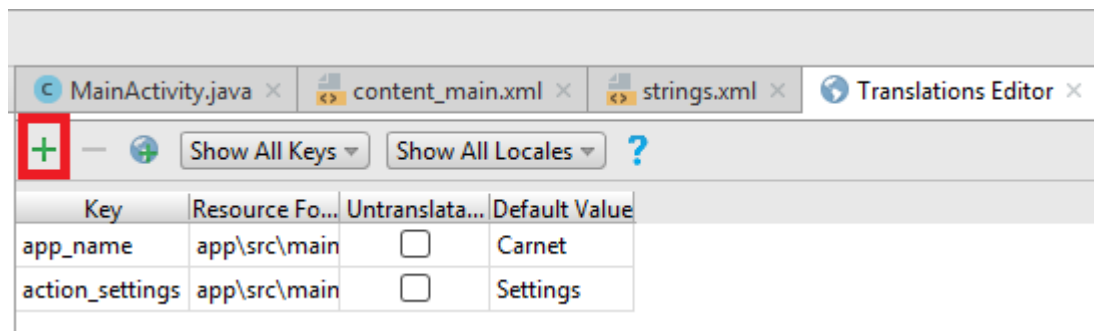
Presione doble clic en archivo String.xml



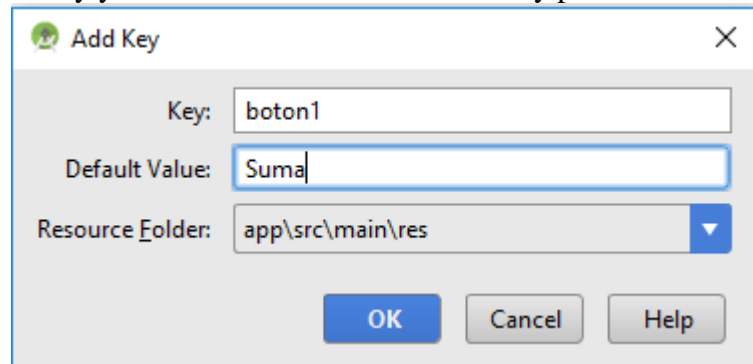
Presione clic en **open_editor**



Presione clic en el signo + para agregar líneas en la tabla de variables string.



Digite los valores de Key y Default Value como se muestra y presione OK



Repita lo anterior para los valores que se muestran

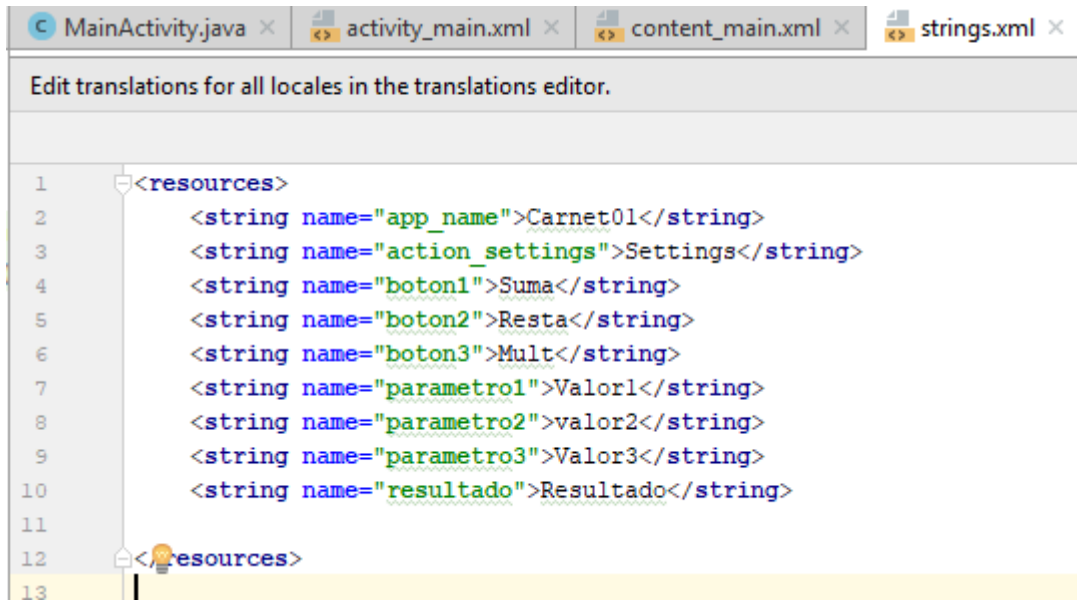
Key	Default Value
boton1	Suma
boton2	Resta
boton3	Mult
parametro1	Valor1
parametro2	Valor2
parametro3	Valor3
resultado	Resultado

En modo visual usted vera que le queda una lista como sigue:

Key	Resource Fo...	Untranslata...	Default Value
app_name	app\src\main	<input type="checkbox"/>	Carnet01
action_settings	app\src\main	<input type="checkbox"/>	Settings
boton1	app\src\main	<input type="checkbox"/>	Suma
boton2	app\src\main	<input type="checkbox"/>	Resta
boton3	app\src\main	<input type="checkbox"/>	Mult
parametro1	app\src\main	<input type="checkbox"/>	Valor1
parametro2	app\src\main	<input type="checkbox"/>	valor2
parametro3	app\src\main	<input type="checkbox"/>	Valor3
resultado	app\src\main	<input type="checkbox"/>	Resultado

Segunda Forma

Presione doble clic en la pestaña interior string.xml y digite lo que está en la imagen:

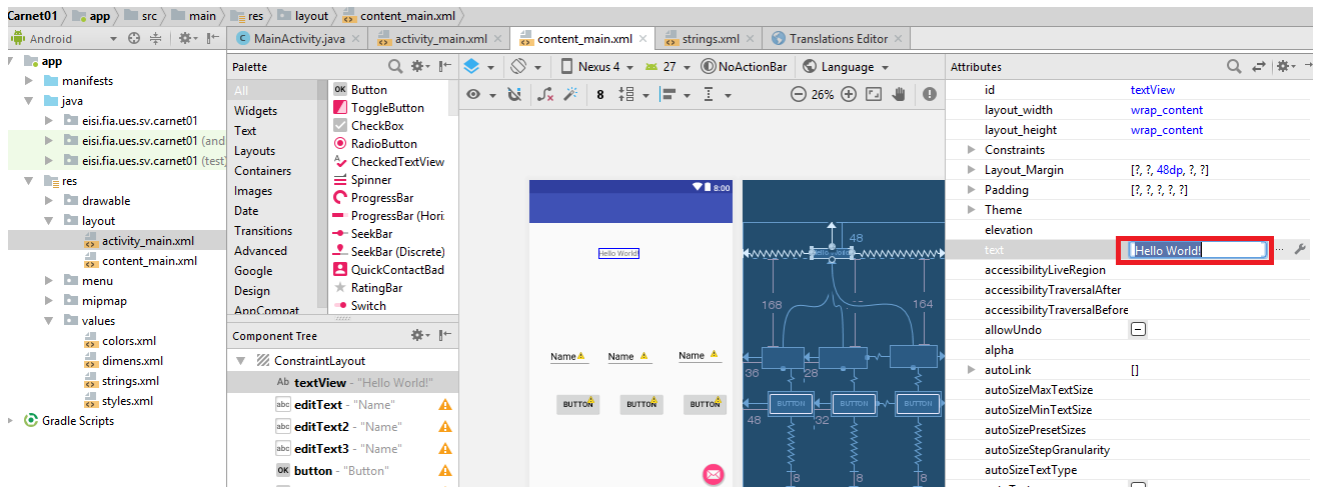


```
1 <resources>
2     <string name="app_name">Carnet01</string>
3     <string name="action_settings">Settings</string>
4     <string name="boton1">Suma</string>
5     <string name="boton2">Resta</string>
6     <string name="boton3">Mult</string>
7     <string name="parametro1">Valor1</string>
8     <string name="parametro2">valor2</string>
9     <string name="parametro3">Valor3</string>
10    <string name="resultado">Resultado</string>
11
12 </resources>
13
```

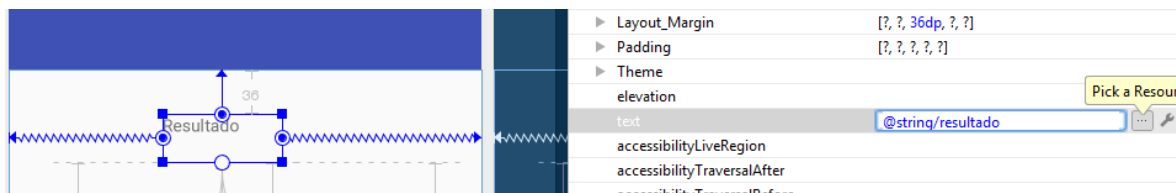
De cualquiera de las formas que la haya hecho, se recomienda que guarde su archivo en este momento.

Asociar controles con los recursos

Seleccione el `activiti_main.xml` y Presione clic en EditView (HelloWorld) luego presione doble clic en los puntos suspensivos de la propiedad `text(2)`



Seleccione el recurso Resultado(resultado) y clic en OK



Vera como cambia la interfaz grafica y la propiedad `text: @string/resultado`

Repita lo anterior para las propiedades `hint` y `text` de los siguientes controles

Objeto(interfaz)	propiedad	Valor
button2	text	@string/boton2
button3	text	@string/boton3
button4	text	@string/boton4
EditText1	hint	@string/parametro1
EditText2	hint	@string/parametro2
EditText3	hint	@string/parametro3

Si no lo quiere hacer de manera visual, hágalo directamente digitando el nombre del recurso en la propiedad requerida.

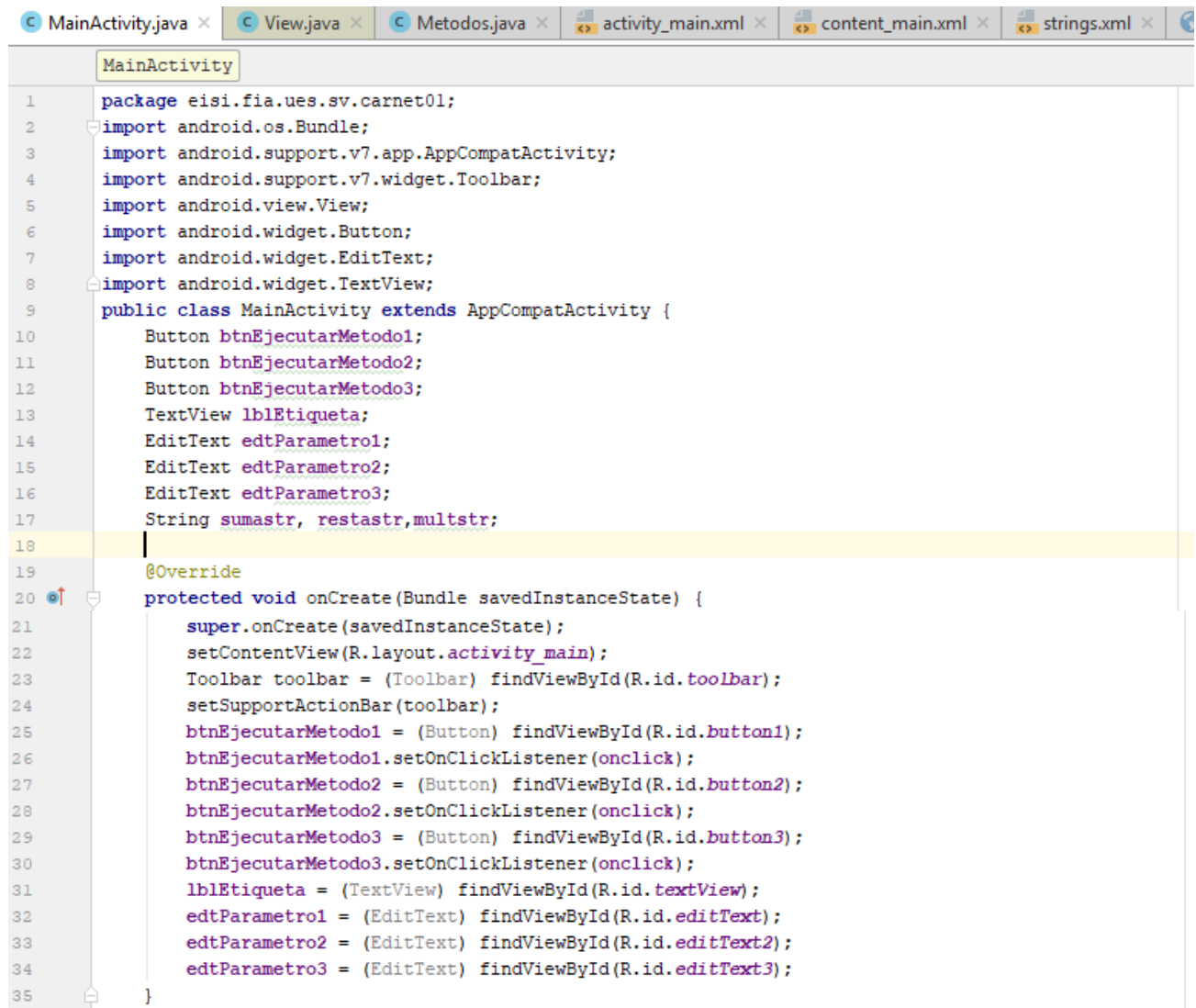
Al finalizar vera su interfaz gráfica similar a esta:



Nota: si no le aparecen los parámetros con el texto valor1, valor2 y valor3, elimine la propiedad text de todos.

Modificación de Aplicación

Entre a la aplicación MainActivity.java en src y vera el siguiente código. Elimine la importación de librerías y código que no se necesite.



```
1 package eisi.fia.ues.sv.carnet01;
2 import android.os.Bundle;
3 import android.support.v7.app.AppCompatActivity;
4 import android.support.v7.widget.Toolbar;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.EditText;
8 import android.widget.TextView;
9 public class MainActivity extends AppCompatActivity {
10     Button btnEjecutarMetodo1;
11     Button btnEjecutarMetodo2;
12     Button btnEjecutarMetodo3;
13     TextView lblEtiqueta;
14     EditText edtParametro1;
15     EditText edtParametro2;
16     EditText edtParametro3;
17     String sumastr, restastr, multstr;
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_main);
23         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
24         setSupportActionBar(toolbar);
25         btnEjecutarMetodo1 = (Button) findViewById(R.id.button1);
26         btnEjecutarMetodo1.setOnClickListener(onclick);
27         btnEjecutarMetodo2 = (Button) findViewById(R.id.button2);
28         btnEjecutarMetodo2.setOnClickListener(onclick);
29         btnEjecutarMetodo3 = (Button) findViewById(R.id.button3);
30         btnEjecutarMetodo3.setOnClickListener(onclick);
31         lblEtiqueta = (TextView) findViewById(R.id.textView);
32         edtParametro1 = (EditText) findViewById(R.id.editText);
33         edtParametro2 = (EditText) findViewById(R.id.editText2);
34         edtParametro3 = (EditText) findViewById(R.id.editText3);
35     }
```

```

35     }
36     View.OnClickListener onclick = new View.OnClickListener() {
37         @Override
38         public void onClick(View v) {
39             // TODO Auto-generated method stub
40             int numint1, numint2, numint3;
41             float numfloat1, numfloat2;
42             switch (v.getId()) {
43                 case R.id.button1:
44                     numfloat1 = (float) Float.valueOf(edtParametro1.getText().toString());
45                     numfloat2 = (float) Float.valueOf(edtParametro2.getText().toString());
46                     sumastr = Metodos.suma(numfloat1, numfloat2);
47                     lblEtiqueta.setText(sumastr);
48                     break;
49                 case R.id.button2:
50                     numfloat1 = (float) Float.valueOf(edtParametro1.getText().toString());
51                     numfloat2 = (float) Float.valueOf(edtParametro2.getText().toString());
52                     restastr = Metodos.resta(numfloat1, numfloat2);
53                     lblEtiqueta.setText(restastr);
54                     break;
55                 case R.id.button3:
56                     numfloat1 = (int) Integer.valueOf(edtParametro1.getText().toString());
57                     numfloat2 = (int) Integer.valueOf(edtParametro2.getText().toString());
58                     multstr = Metodos.mult(numfloat1, numfloat2);
59                     lblEtiqueta.setText(multstr);
60                     break;
61             }
62         }
63     };
64 }

```

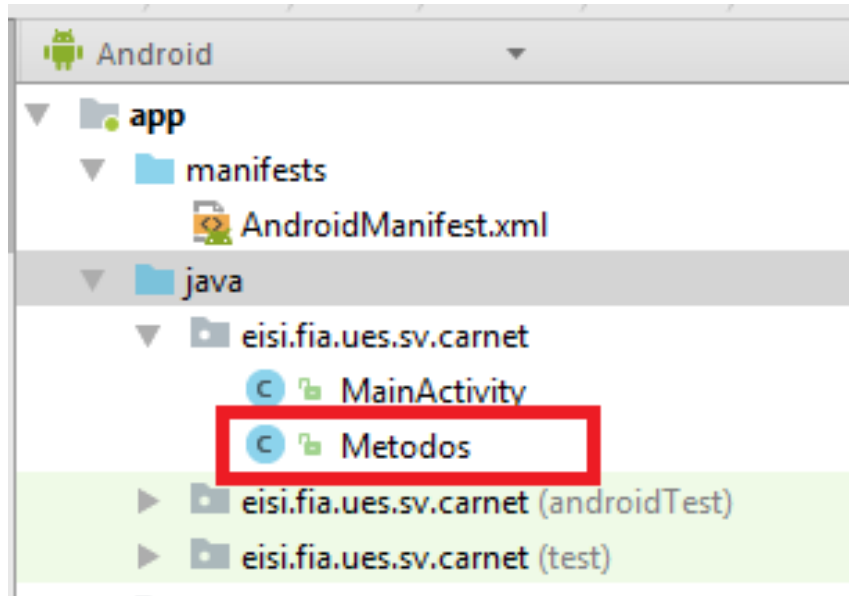
Modifique el código hasta que aparezca como el anterior

Preguntas:

¿Por qué tiene punto y coma la última línea de comando?

¿Es un error?

Bajo el mismo paquete que esta el MainActivity, agregue una nueva clase(clic derecho en el paquete, new , class) que se llame Metodos.



Y dígame el código siguiente:

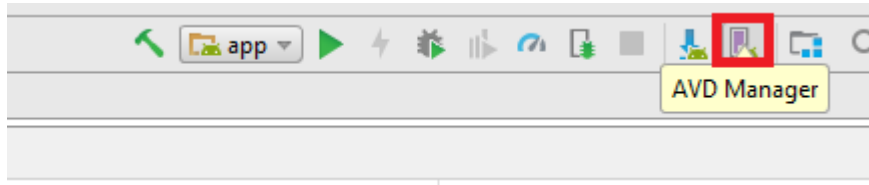
```
1 package eisi.fia.ues.sv.carnet01;
2
3 /**
4  * Created by Win10 on 05/03/2018.
5  */
6 public class Metodos {
7     public static String suma(float valor1, float valor2) {
8         String sumastr;
9         float sumafloat =valor1+valor2;
10        sumastr= String.valueOf(sumafloat);
11        return sumastr;
12    }
13    public static String resta(float valor1, float valor2){
14        String restastr;
15        float sumafloat =valor1-valor2;
16        restastr= String.valueOf(sumafloat);
17        return restastr;
18    }
19    public static String mult(float valor1, float valor2){
20        String multstr;
21        float multfloat =valor1*valor2;
22        multstr= String.valueOf(multfloat);
23        return multstr;
24    }
25 }
```

Pruébelo y al finalizar suba su proyecto comprimido en el link respectivo del aula.

Si existe algún error puede descargar el código fuente en el ftp de la asignatura para verificar.

Anexo 1 Creación de Dispositivo Virtualizado

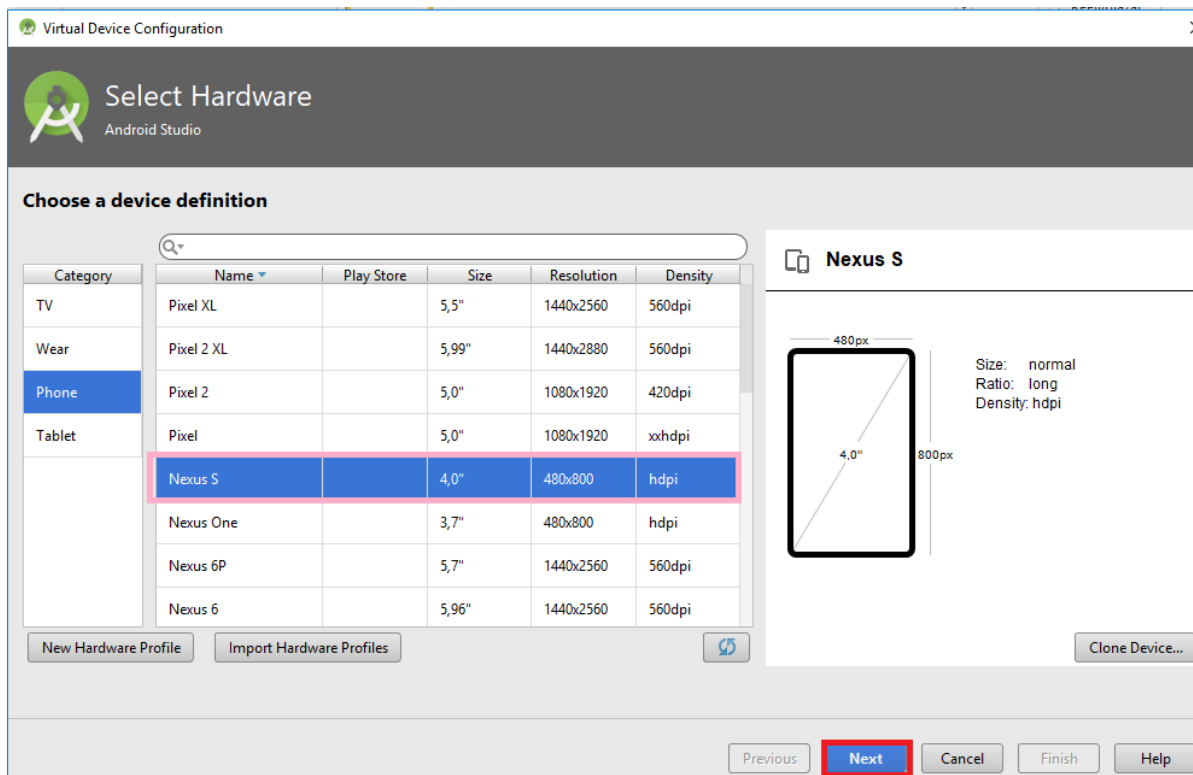
Entre al icono del AVD manager.



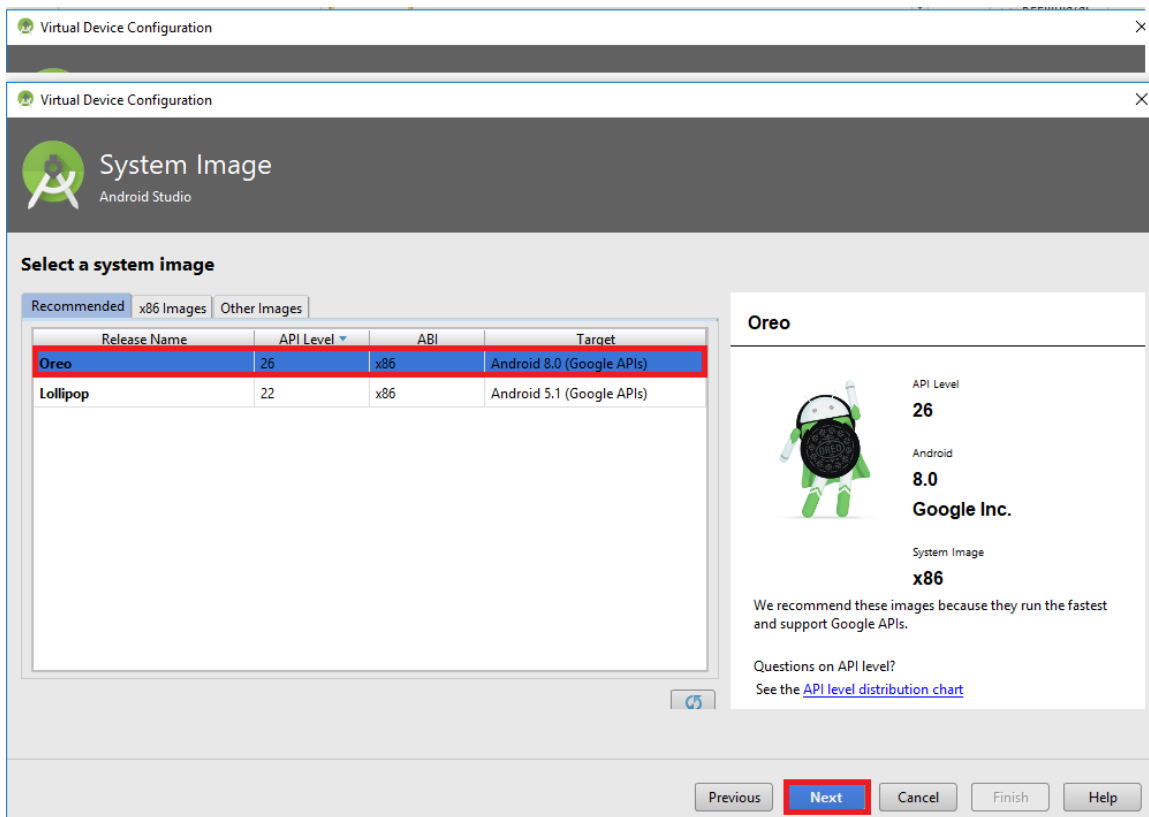
Presione click en Crear dispositivo virtual.



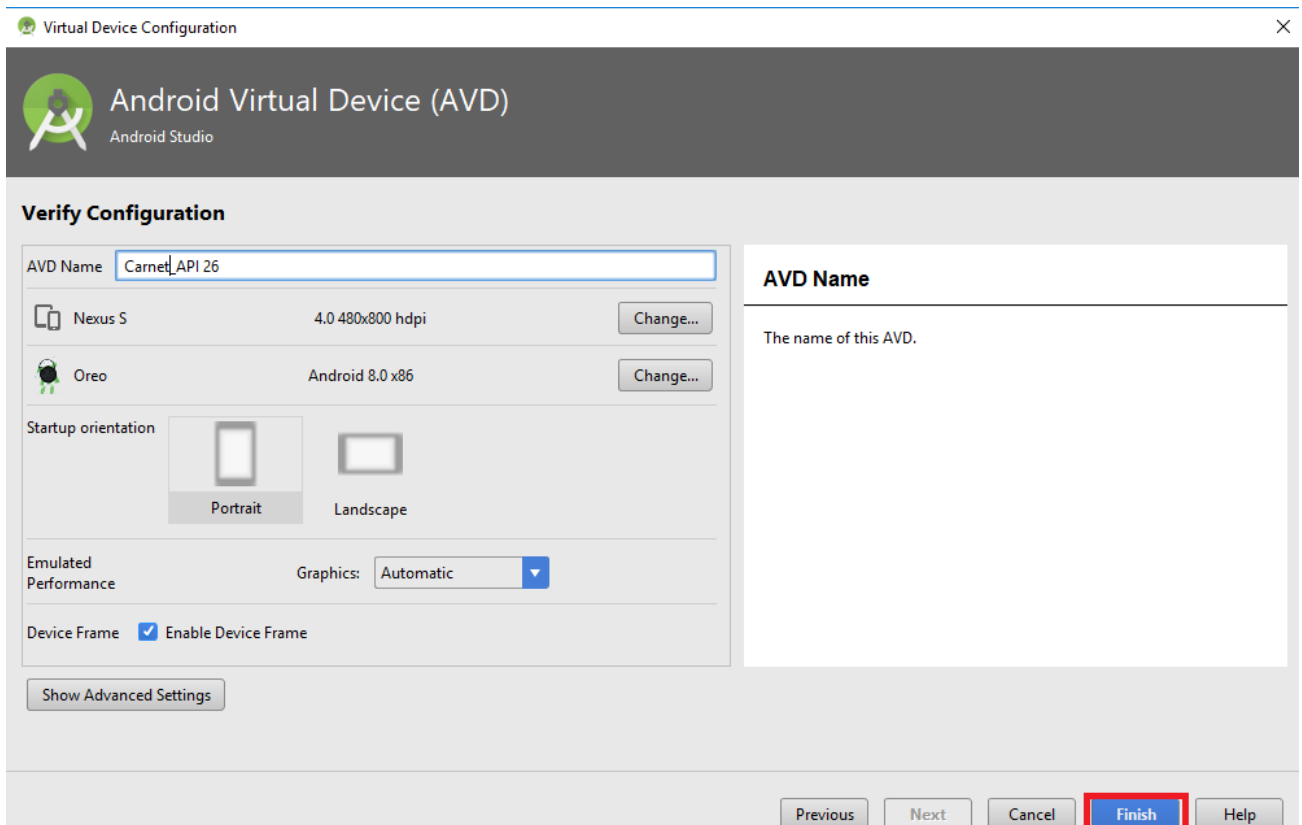
Luego seleccione el teléfono (phone) Nexus S de 4" y presione clic en siguiente.



Luego seleccione Oreo(api 26) o Lollipop(api 22), y presione siguiente.

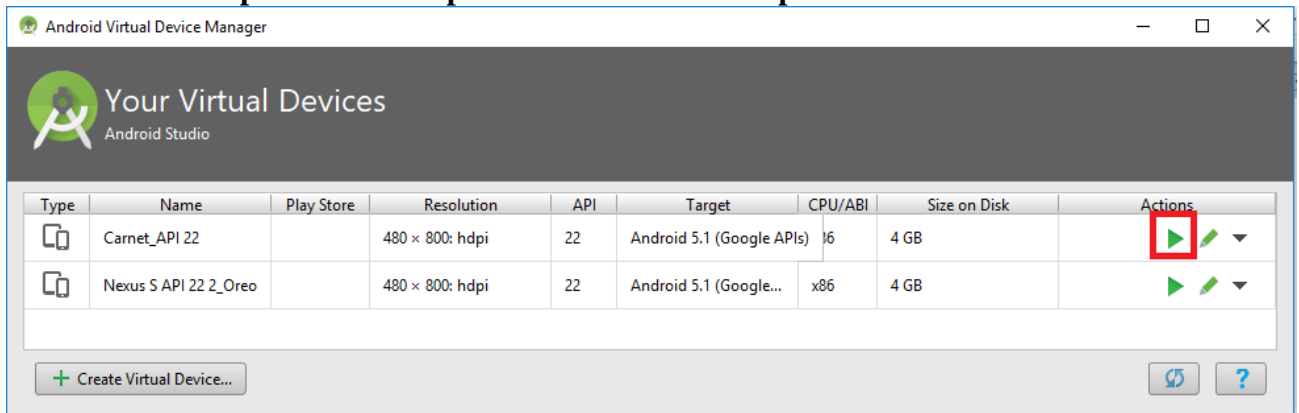


Luego escriba el nombre del dispositivo, se sugiere su **carnet_API26** O 22 según decida.



Luego presione clic en Finish

Con lo anterior aparecerá su dispositivo en la lista de disponibles



Presione clic en el icono de play y en unos minutos su dispositivo estará cargado y listo para ejecutar las aplicaciones que desarrolle.

Si tiene muchos inconvenientes en crear su dispositivo, utilice otro de los que están en la lista.