



**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**



Objetivos:

*Conocer el entorno de programación del IDE Android Studio Bumblebee 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)

Índice

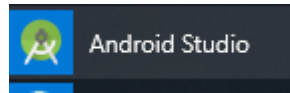
Crear una aplicación Básica	3
Reconocimiento del Entorno	8
Creacion de Interfaz Grafica	9
<i>Crear de Interfaz Grafica</i>	9
<i>Configuración de los controles</i>	1
Configuración de los controles	2
Configuración de Valores de los controles button y de los mensajes de los editText	5
Agregar recursos (variables string)	5
Primera forma	5
Segunda Forma	7
Asociar controles con los recursos	8
Modificación de MainActivity.java	12
Creación de Metodos.java	15
Anexos	17
Anexo 1 Código de Metodos.java	17

Entrada al Android Studio en Lcomp4(EISI-FIA-UES)

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

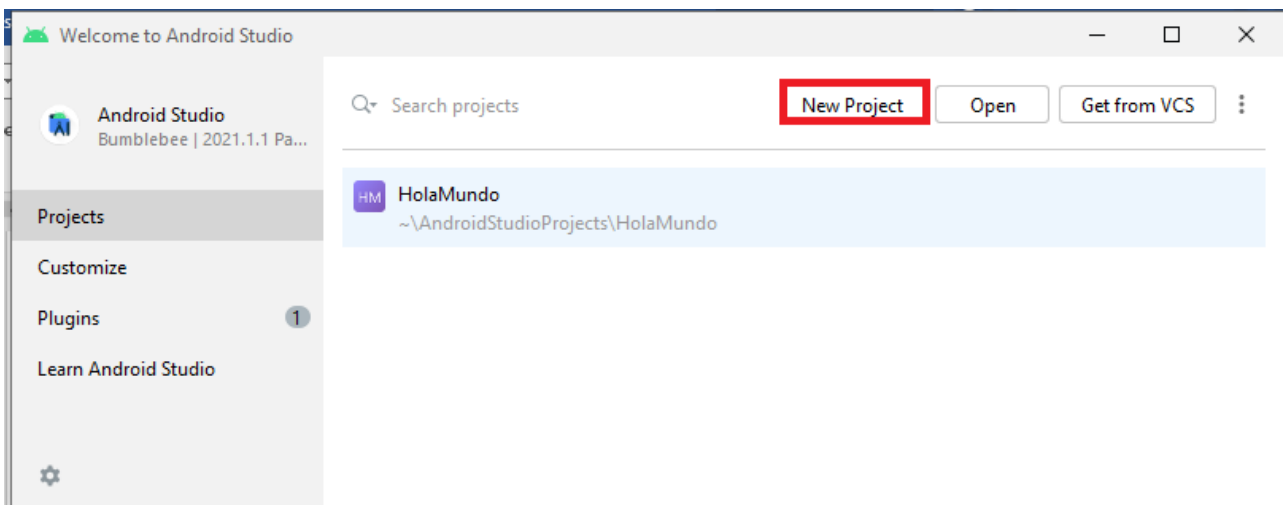
Entre a la Sesión de PDM115 con clave pdm (o pdm2020)

En el icono Android Studio o entre a inicio/aplicaciones/Android Studio

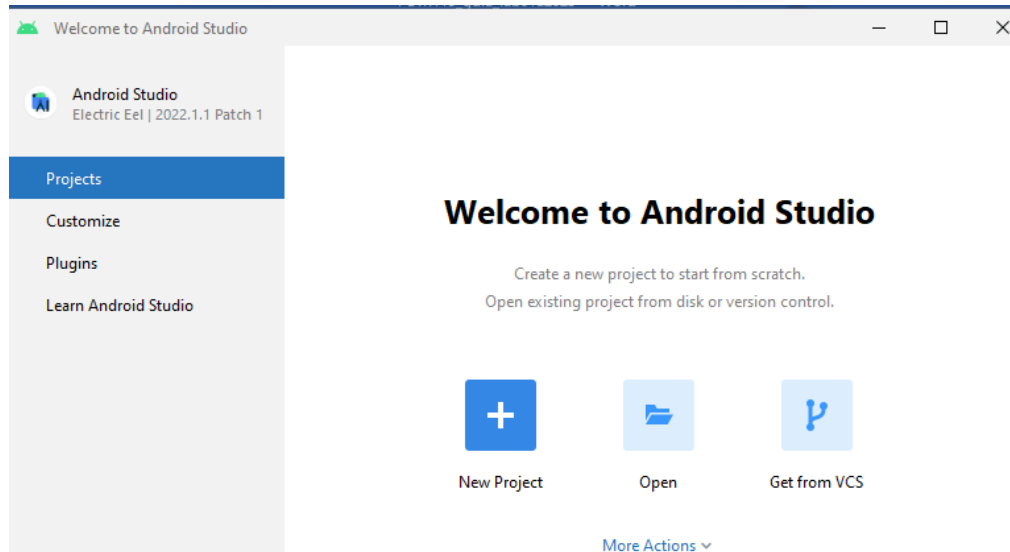


Crear una aplicación Básica

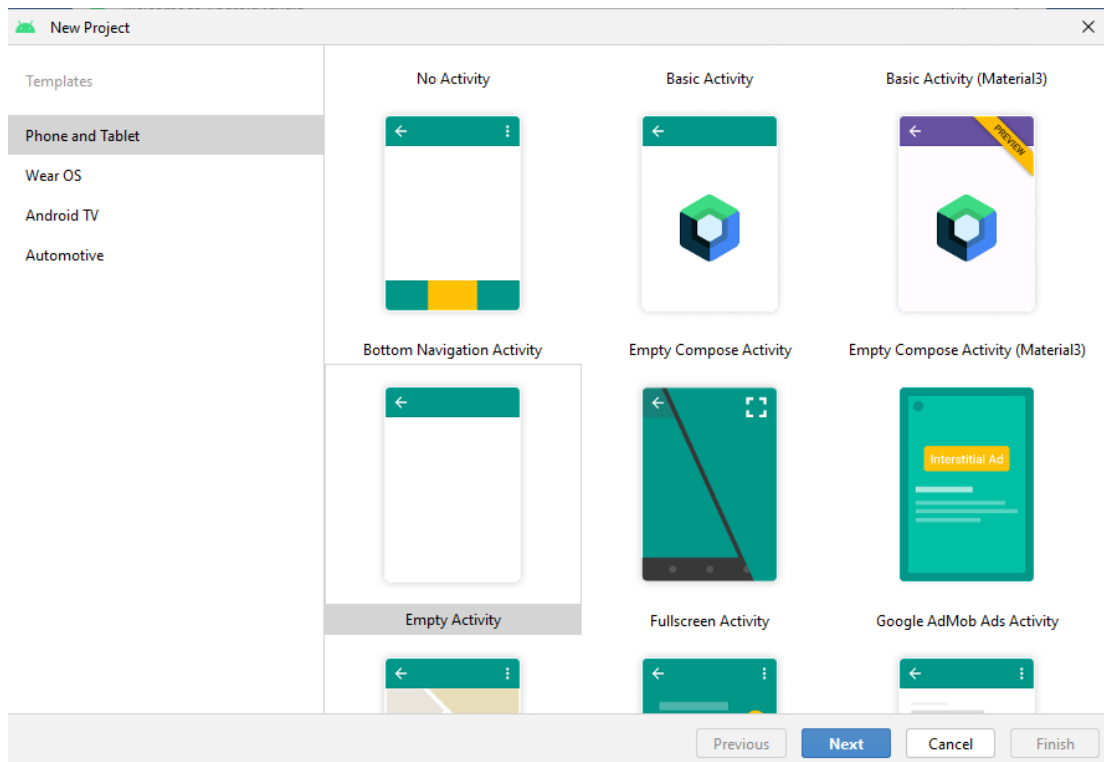
Presione clic en “New Project” Si hiciste la aplicación de Holamundo se vera de esta forma



Si no la hiciste se vera de esta otra forma.

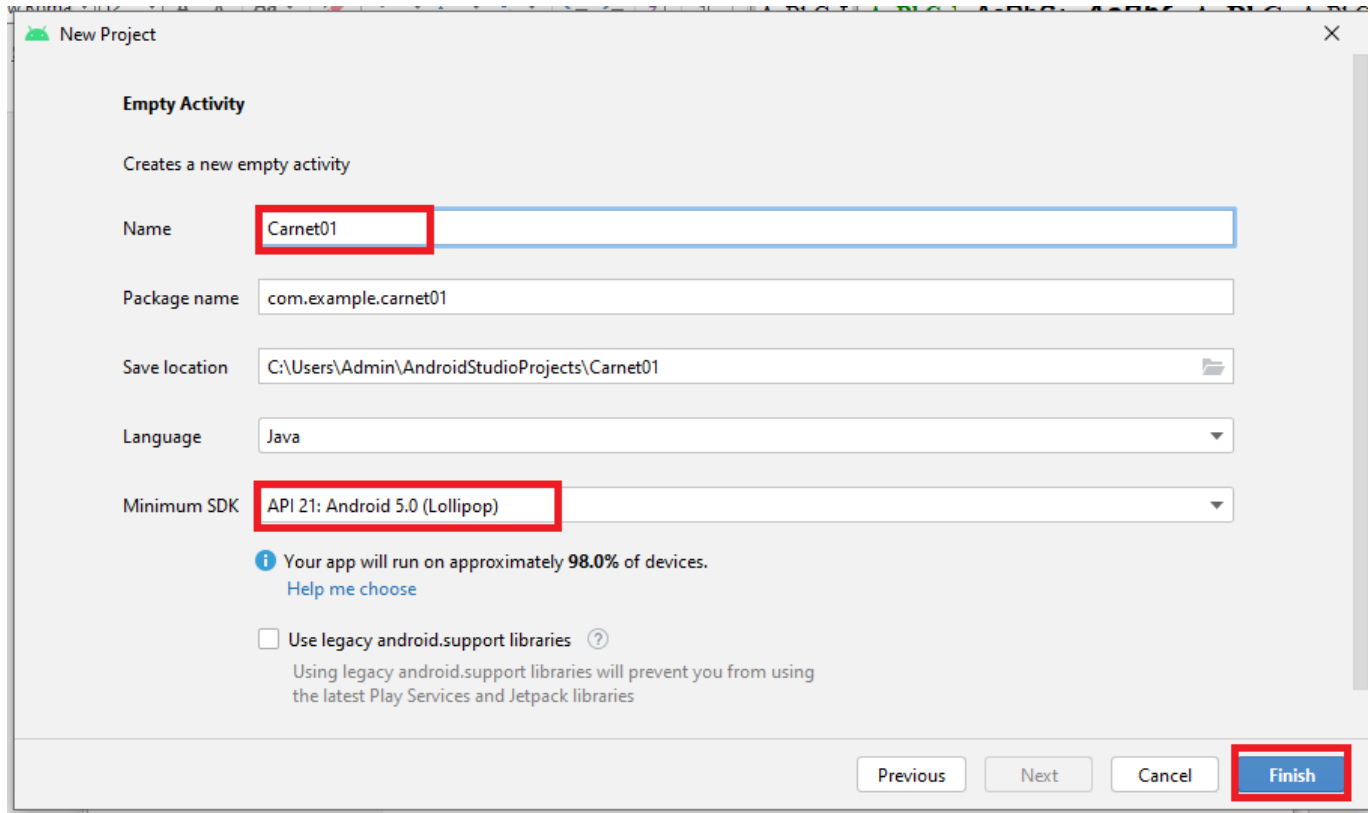


Seleccione el tipo de Proyecto que desea crear(Empty Activity)



Definir parámetros

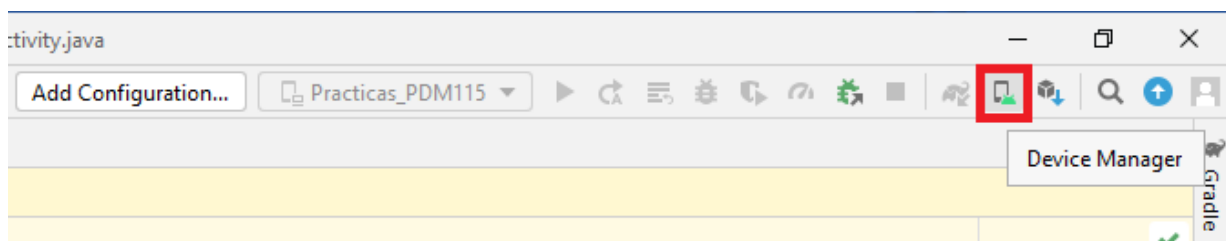
Seleccione la versión mínima en la que correrá su app(API 21) y el nombre de su aplicación(su carnet con correlativo 01)



Como nombre de su aplicación ponga su carnet, defina nombre del dominio como se muestra, el parámetro del Lenguaje debe ser **java**. y presione Siguiente(Finish)

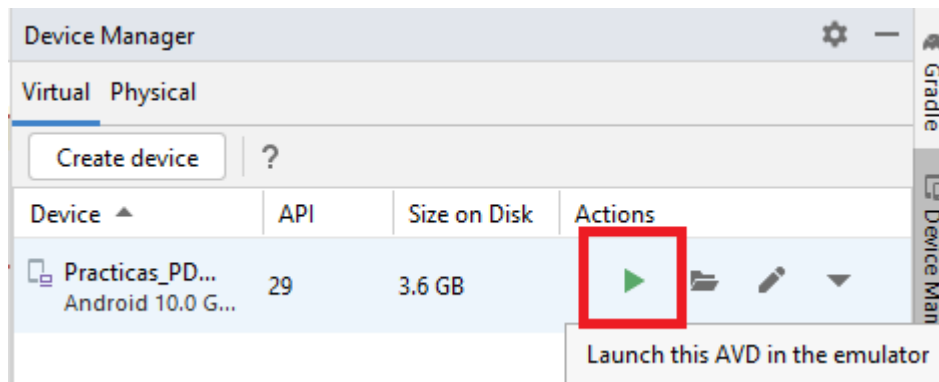
Ponga en Ejecución el Dispositivo Virtual Existente (hecho en la guía de instalación)

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

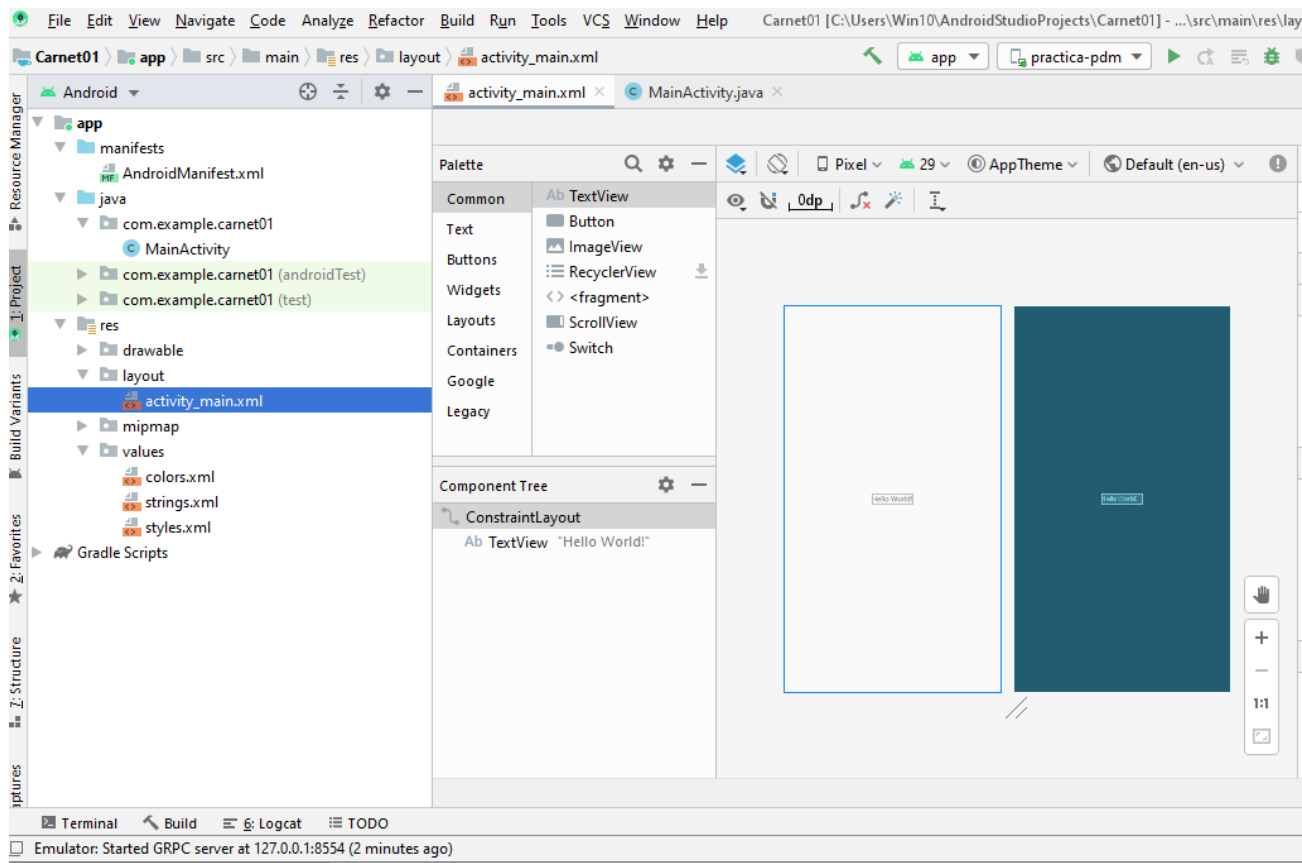


Nota: si aparece una ventana de Android Virtual Device Manager entonces créelo como se indicó en el Manual de instalación

Se buscará un Nexus S(PracticasPDM115) previamente instalado, si ud desea uno diferente deberá crearlo (Create Virtual Device...)

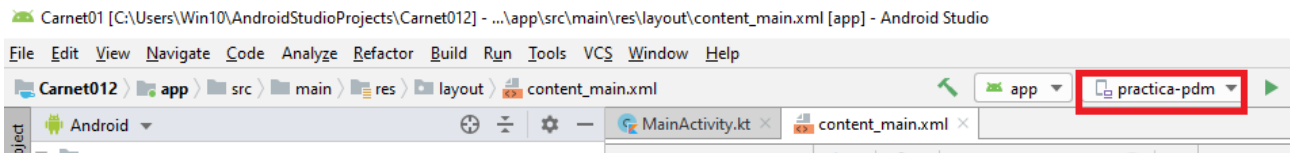


Con lo anterior, deberá aparecer la imagen como se muestra

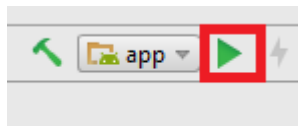


Ejecute la aplicación que imprime Holamundo(HelloWorld) que ya trae elaborada el IDE.

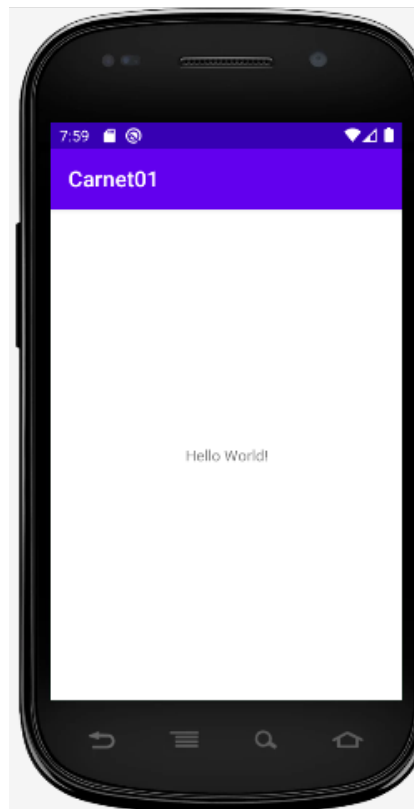
Verifique que su dispositivo sea el que está definido para correr la aplicación



Presione clic en run app o shift+F10

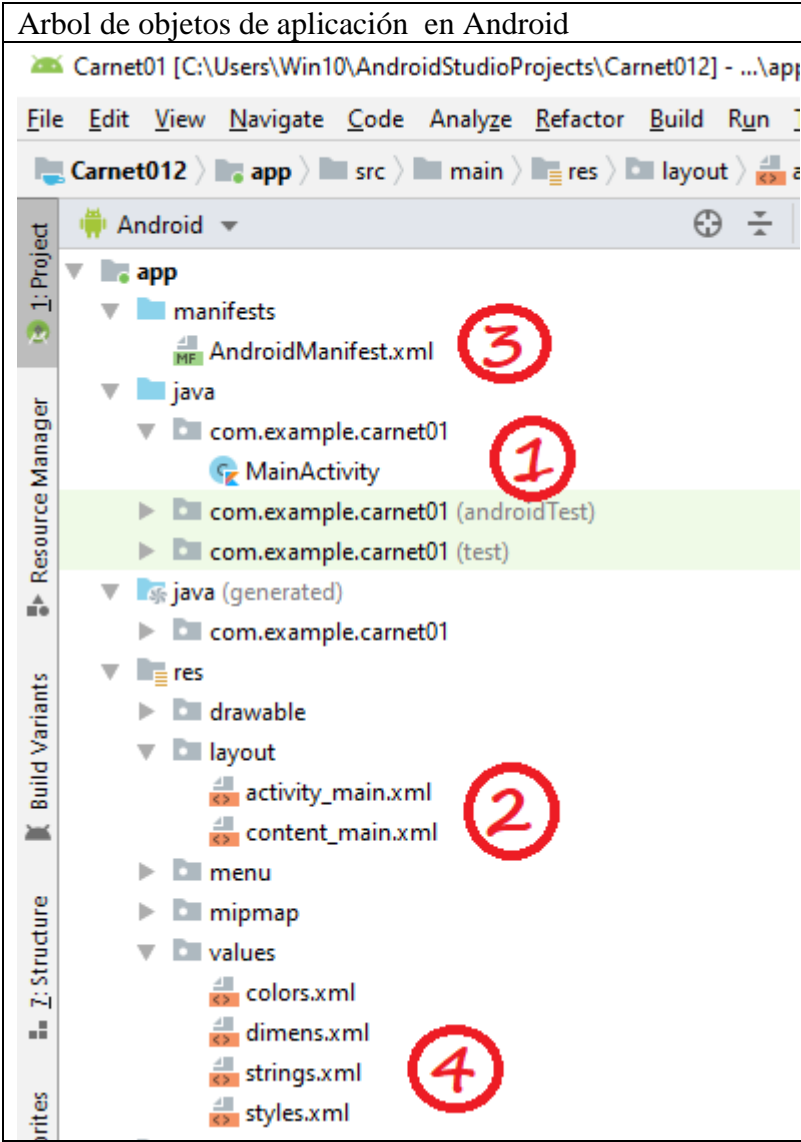


Con lo anterior se cargará la app en el emulador como se muestra.



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

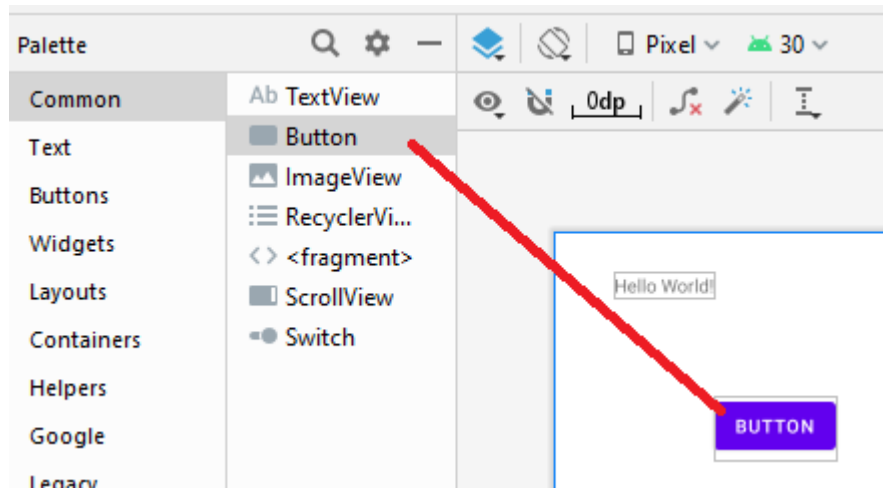
Reconocimiento del Entorno

Arbol de objetos de aplicación en Android	Descripcion
	<p>Se trabajara en esta guia cuatro 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)</p> <p>Se utilizara ademas un archivo de recursos llamado string.xml(4) para definir variables utiles para su interfaz grafica.</p>

Creacion de Interfaz Grafica

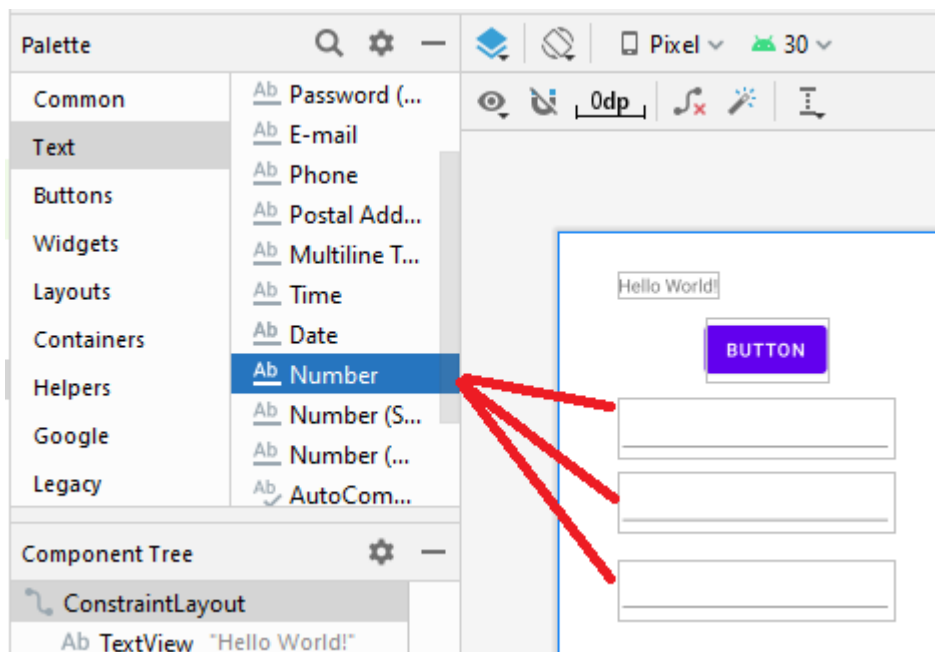
Crear de Interfaz Grafica

Indicaciones: Hacer una interfaz que contenga ademas del textview de hello world(etiqueta), un boton. Arrastrando el objeto desde la paleta hasta el editor grafico de la interfaz(res, layout) como se muestra. Luego arrastraremos los objetos textView(3)



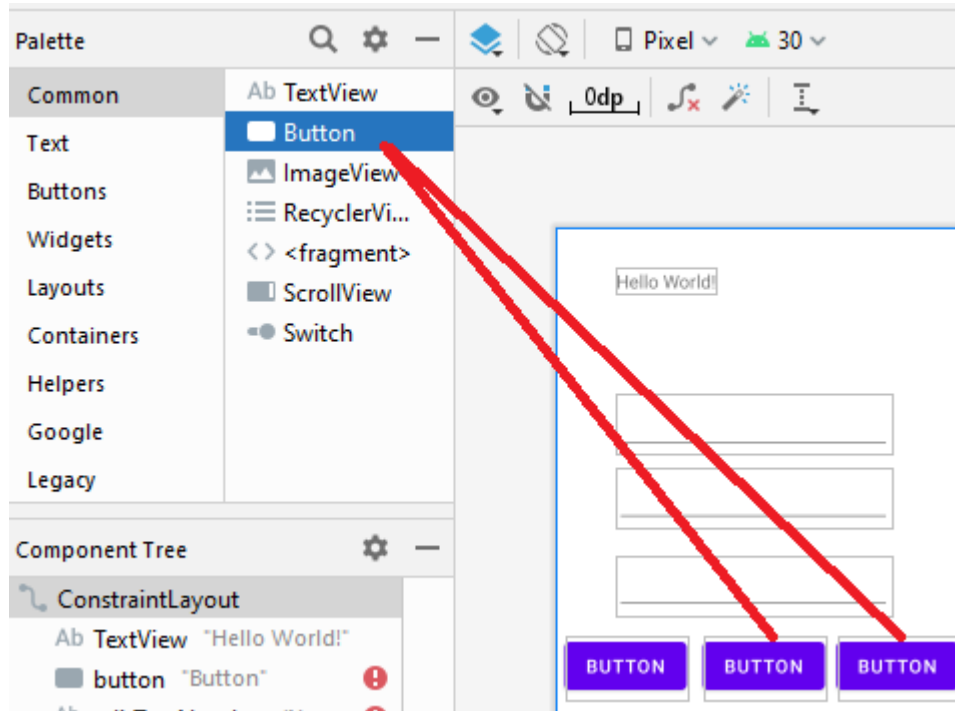
C

Arrastre tres controles Text numéricos (Number)



Configuración de los controles

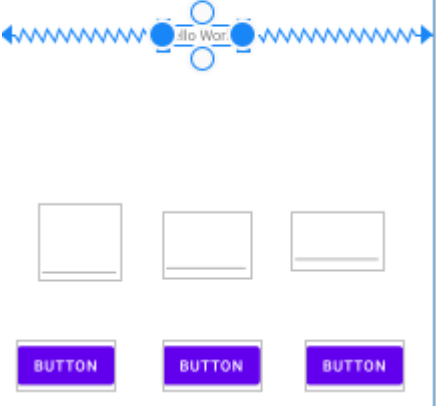
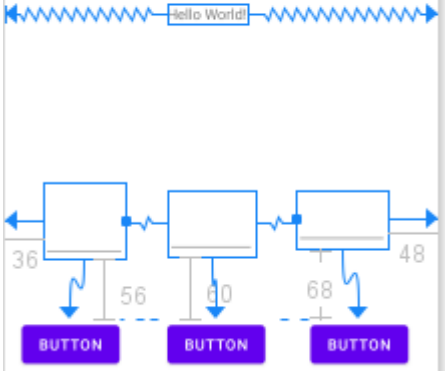
Mueva el primer botón hacia debajo de los Textos, luego entre a pestaña Buttons o Common y Agregue otros 2 botones.

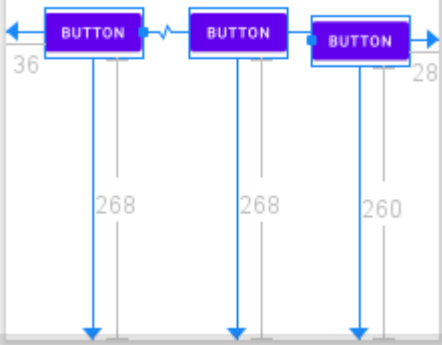
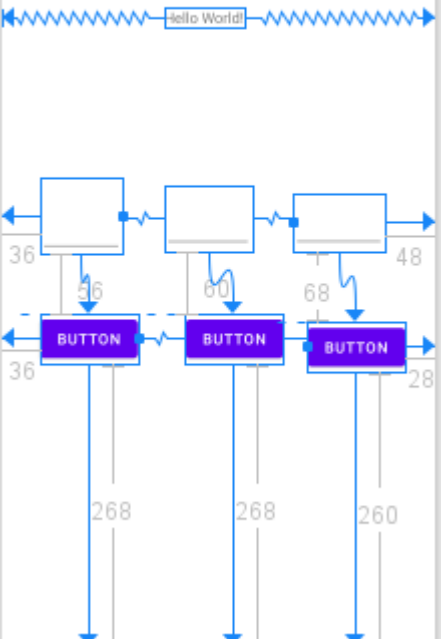


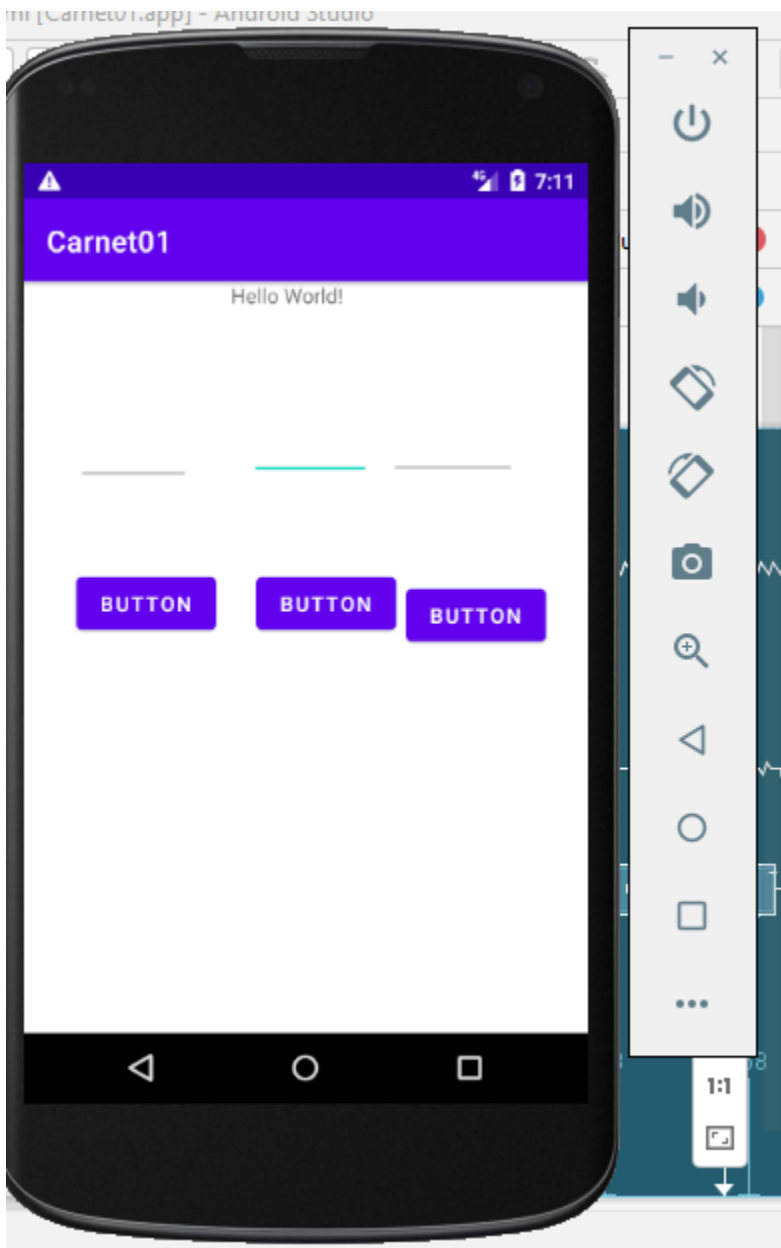
En este momento vamos a configurar la posición relativa de los Controles(objetos) y agregar variables de recursos asociada a la propiedad text de los objetos button. Se desea que el primer botón diga intento, los tres editText digan sumando1,sumando2 y sumando3 . Tendremos que modificar el archivo de recursos string.xml y luego cambiar la propiedad text en los botones y hint en los textos para direccionar el contenido hacia esos recursos.

Configuración de los controles

Constraints de interfaz grafica

	<p>Arrastre hacia el centro(arriba) el Textview(Hello World) hasta el EditText del centro.Elimine los constraints de arriba y abajo. Con Ello habrá configurado en la interfaz grafica una restricción(constraint) que en tiempo de ejecución le indicara al Android que el TextView esta alineado con el EditText Central....continue con la configuraciones que faltan siguiendo la secuencia que se le indica..</p>
	<p>Enlazar el editText1 como se muestra</p>
	<p>Enlazar el Button1 con el editText1 , el Button2 con el editText2 y el Button3 con el editText3 ,</p>

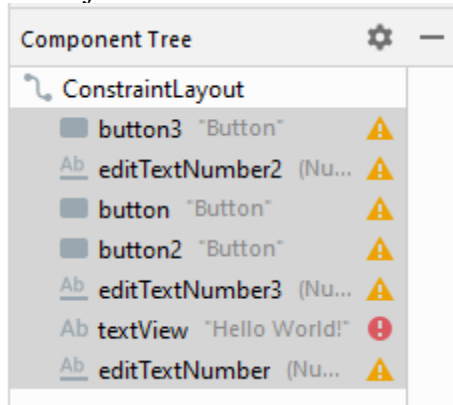
		<p>Enlazar los Buttons como se muestra</p>
		<p>Al finalizar ajuste los controles para que queden como se muestra... ejecute la aplicación.</p>



Nota: si su interfaz no se muestra como esta. Ajuste las propiedades Text(Buttons) y las de Layout, constraints de cada control (Start, End, top, bottom).
A modo de que se puedan ver los botones y los EditText.

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

En la ejecución anterior se vio este árbol de componentes(con warnings o advertencias)

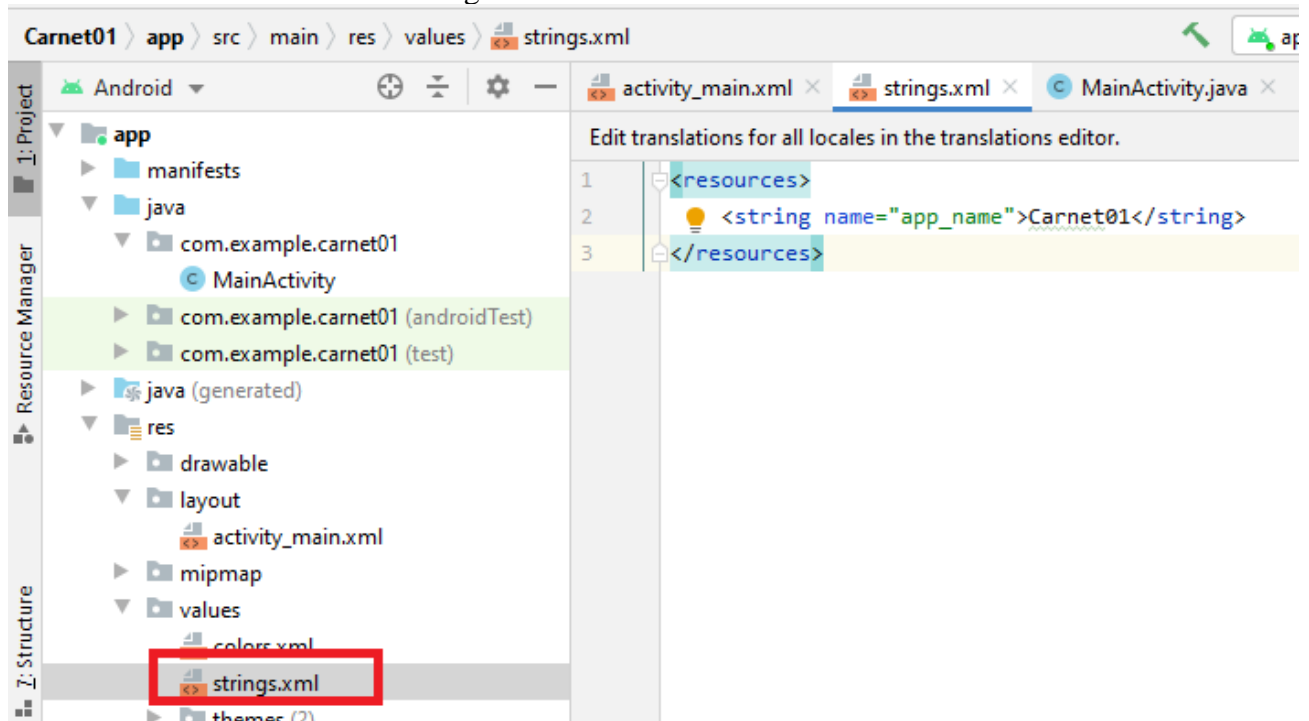


Para quitarlas, primero tendremos que modificar el archivo de recursos string.xml y luego cambiar la propiedad text o hint en los controles para direccionar el contenido hacia esos recursos.

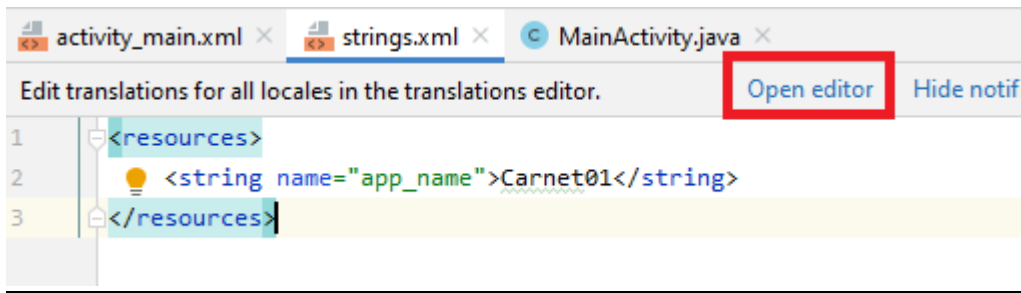
Agregar recursos (variables string)

Primera forma

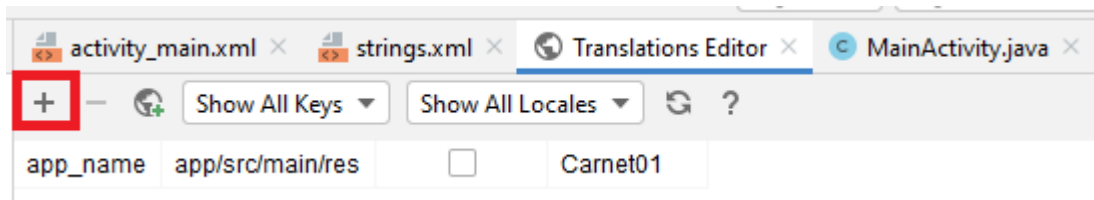
Presione doble clic en archivo String.xml en values



Presione clic en **open_editor**



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



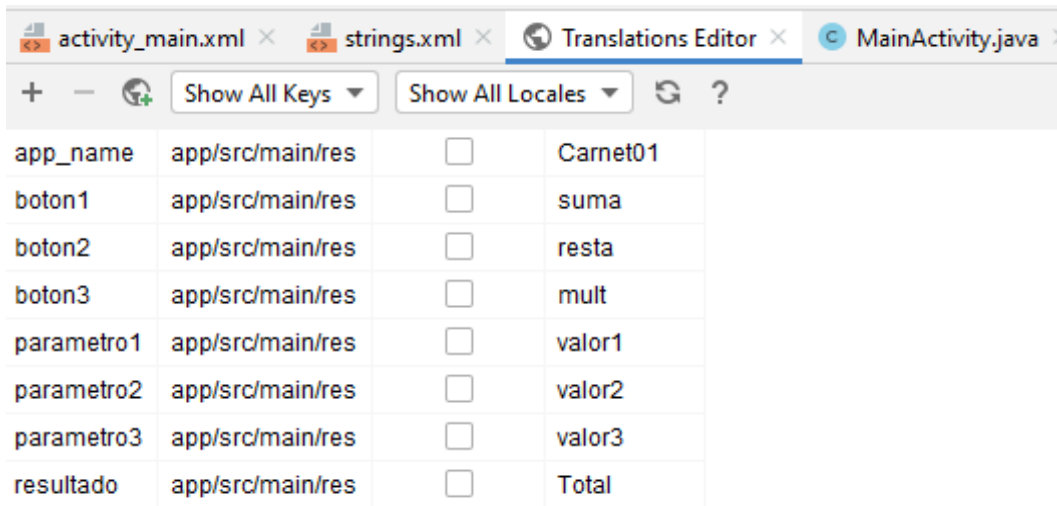
Digite los valores de Key y Default Value como se muestra y presione OK

The screenshot shows the 'Add Key' dialog box. The 'Key' field contains 'boton1' and the 'Default Value' field contains 'suma'. The 'Resource Folder' is set to 'app/src/main/res'. The 'OK' button is highlighted with a blue box.

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	Total

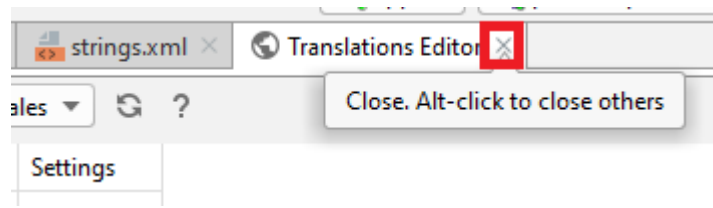
En modo visual usted vera que le queda una lista como sigue:



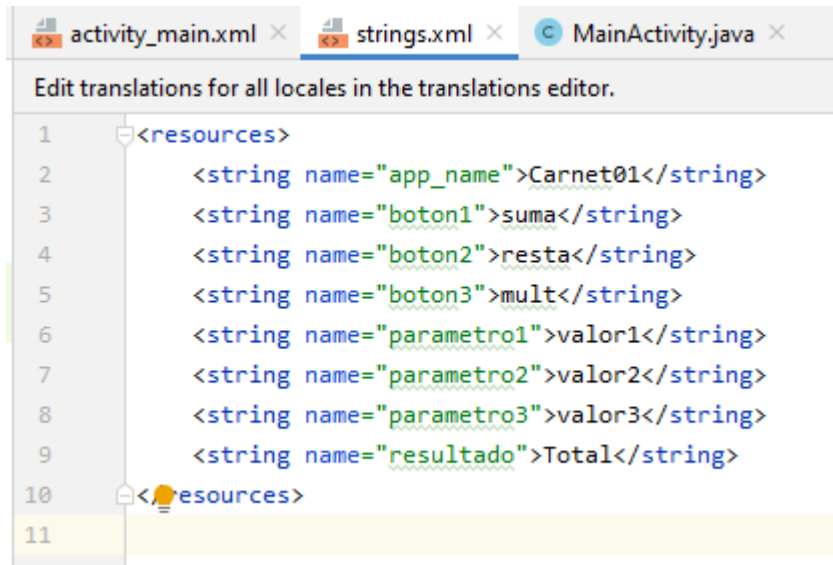
activity_main.xml	strings.xml	Translations Editor	MainActivity.java
+	-	Show All Keys	Show All Locales
app_name	app/src/main/res	<input type="checkbox"/>	Carnet01
boton1	app/src/main/res	<input type="checkbox"/>	suma
boton2	app/src/main/res	<input type="checkbox"/>	resta
boton3	app/src/main/res	<input type="checkbox"/>	mult
parametro1	app/src/main/res	<input type="checkbox"/>	valor1
parametro2	app/src/main/res	<input type="checkbox"/>	valor2
parametro3	app/src/main/res	<input type="checkbox"/>	valor3
resultado	app/src/main/res	<input type="checkbox"/>	Total

Segunda Forma

Cierre el Translations Editor(si lo tiene abierto)



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

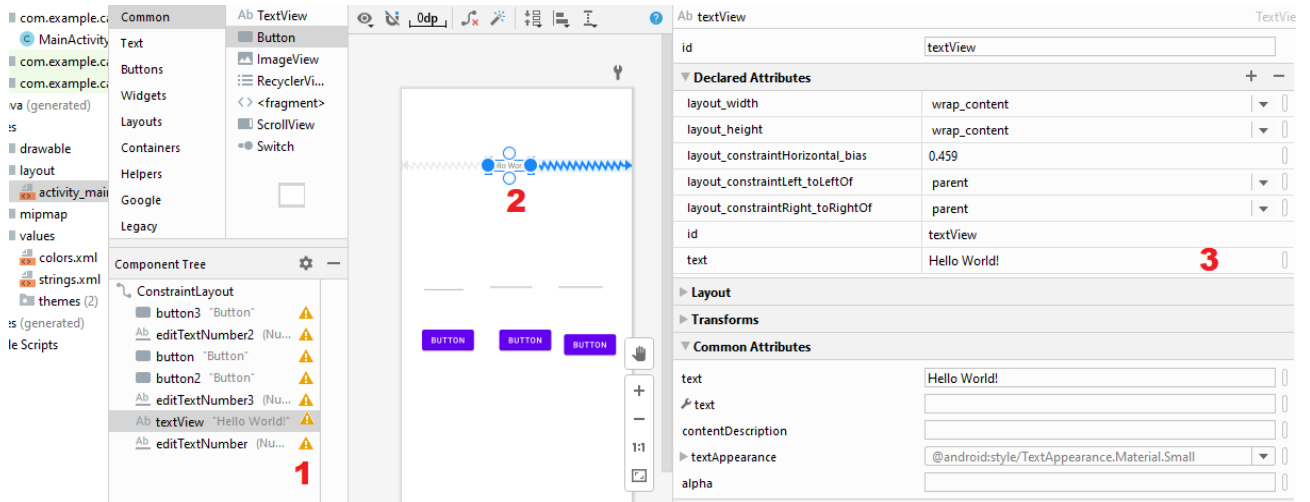


```
activity_main.xml x strings.xml x MainActivity.java x
Edit translations for all locales in the translations editor.
1 <resources>
2     <string name="app_name">Carnet01</string>
3     <string name="boton1">suma</string>
4     <string name="boton2">resta</string>
5     <string name="boton3">mult</string>
6     <string name="parametro1">valor1</string>
7     <string name="parametro2">valor2</string>
8     <string name="parametro3">valor3</string>
9     <string name="resultado">Total</string>
10 </resources>
11
```

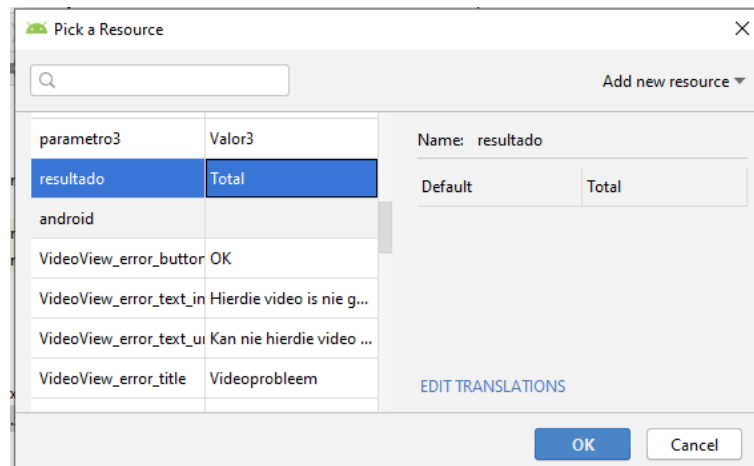
De cualquiera de las formas que la haya hecho, se recomienda que guarde su archivo en este momento.

Asociar controles con los recursos

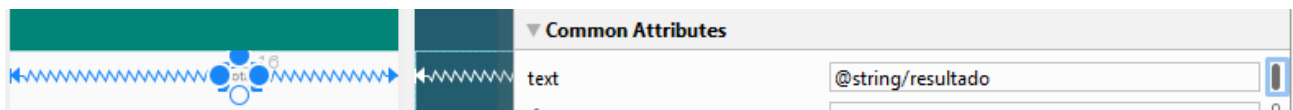
Seleccione el Activity_main.xml, note que hay componentes de su aplicación que tienen icono de advertencia (1). Presione clic en editView (HelloWorld) en la ventana de edición, luego vaya a la ventana de propiedades y presione clic en “pick a resource” de la propiedad text(3)



Seleccione el recurso resultado(Total) y clic en OK



Podrá ver como cambia la interfaz gráfica y la propiedad text: @string/resultado

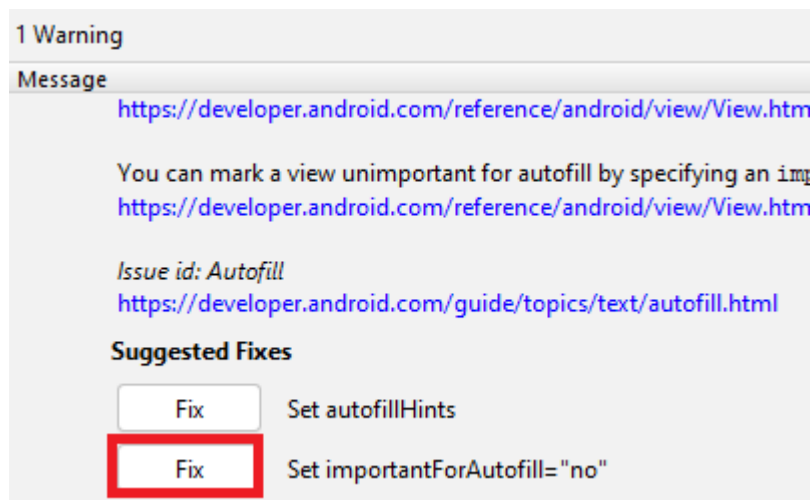


Repita lo anterior para las propiedades hint y text de los siguientes controles

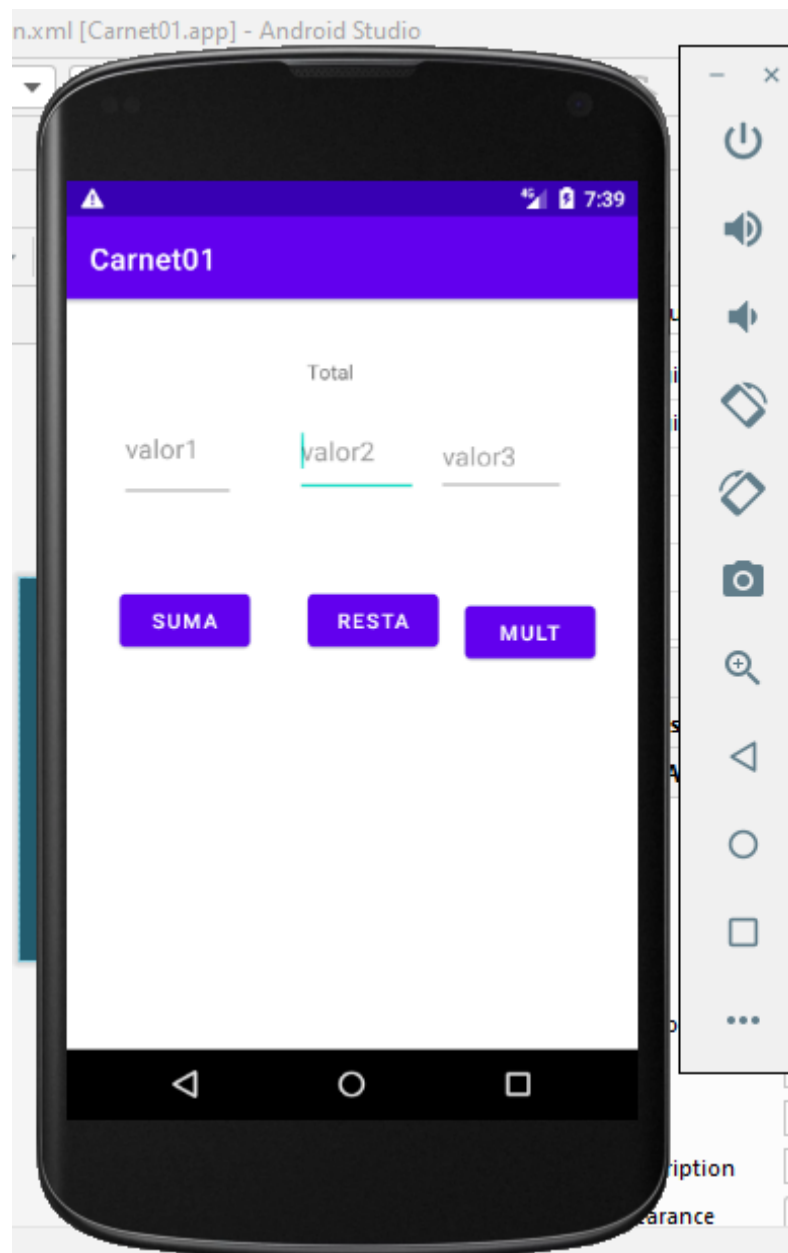
Objeto(interfaz)	propiedad	Valor
button2	text	@string/boton2
button3	text	@string/boton3
button4	text	@string/boton4
EditText1	hint	@string/parametro1
EditText2	hint	@string/parametro2
EditText3	hint	@string/parametro3

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

Si aparecen los warnings como se muestra(Use Autofill...) usar las sugerencias de la ventana warnings como se muestra



Al finalizar vera su interfaz gráfica similar a esta:

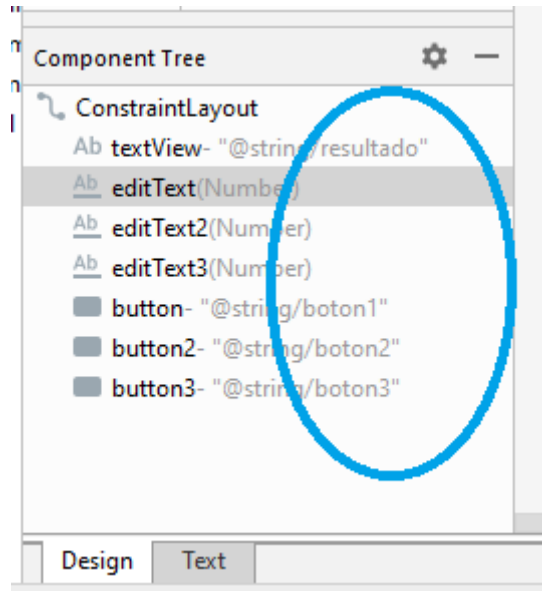


Notas:

- Si no le aparecen los parámetros con el texto valor1, valor2 y valor3, elimine el contenido de la propiedad text de todos los editTexts.

Regresamos a la ventana de Diseño

Verificamos que desaparecieron las advertencias(Warnings)



Modificación de MainActivity.java

Entre a la aplicación MainActivity.java en src y vera el siguiente código. Elimine los elementos de código que no utilizaremos (color rojo).

```
package com.example.carnet01;

import android.os.Bundle;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, text: "Replace with your own action", Snackbar.LENGTH_LONG
                    .setAction(text: "Action", listener: null).show());
            }
        });
    }
}
```

```
30
31
32 *
33
34
35
36
37
38 @
39
40
41
42
43
44
45
46
47
48
49
50
51

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

Suprimir las importaciones de librerías que ya no se utilizaran

Agregar las declaraciones de objetos a utilizar

```
1 package com.example.carnet01;
2 import android.widget.Button;
3 import android.widget.EditText;
4 import android.widget.TextView;
5 import androidx.appcompat.app.AppCompatActivity;
6 import android.os.Bundle;
7 public class MainActivity extends AppCompatActivity {
8     Button btnEjecutarMetodo1;
9     Button btnEjecutarMetodo2;
10    Button btnEjecutarMetodo3;
11    TextView lblEtiqueta;
12    EditText edtParametro1;
13    EditText edtParametro2;
14    EditText edtParametro3;
15    String sumastr, restastr, multstr;
16    @Override
17    protected void onCreate(Bundle savedInstanceState) {
18        super.onCreate(savedInstanceState);
19        setContentView(R.layout.activity_main);
20    }
21 }
```

Con Alt+Enter se puede agregar las importaciones a librerías (import Class)

Agregar el código para enlazar los objetos del formulario(.xml) con los del programa(.java)

```

13      EditText edtParametro2;
14      EditText edtParametro3;
15      String sumastr, restastr, multistr;
16      @Override
17      protected void onCreate(Bundle savedInstanceState) {
18          super.onCreate(savedInstanceState);
19          setContentView(R.layout.activity_main);
20          btnEjecutarMetodo1 = (Button) findViewById(R.id.button1);
21          btnEjecutarMetodo1.setOnClickListener(onclick);
22          btnEjecutarMetodo2 = (Button) findViewById(R.id.button2);
23          btnEjecutarMetodo2.setOnClickListener(onclick);
24          btnEjecutarMetodo3 = (Button) findViewById(R.id.button3);
25          btnEjecutarMetodo3.setOnClickListener(onclick);
26          lblEtiqueta = (TextView) findViewById(R.id.textView);
27          edtParametro1 = (EditText) findViewById(R.id.editText);
28          edtParametro2 = (EditText) findViewById(R.id.editText2);
29          edtParametro3 = (EditText) findViewById(R.id.editText3);
30      }
31  }
32
33

```

Debajo de este apartado debe digitar esta línea incompleta

```

33      edtParametro3 = (EditText) findViewById(R.id.editText3);
34
35
36      View.OnClickListener onClickListener = new View.OnClickListener{
37
38      }

```

Ubicarse en View(rojo) e importar la librería con Control + Enter.

Luego aparecerá en color negro... mover el cursor hasta onClickListener(rojo) y presionar control+shift+TAB

```

    edtParametro3 = (EditText) findViewById(R.id.editText3);
}

View.OnClickListener onClickListener = new View.OnClickListener{
}

```

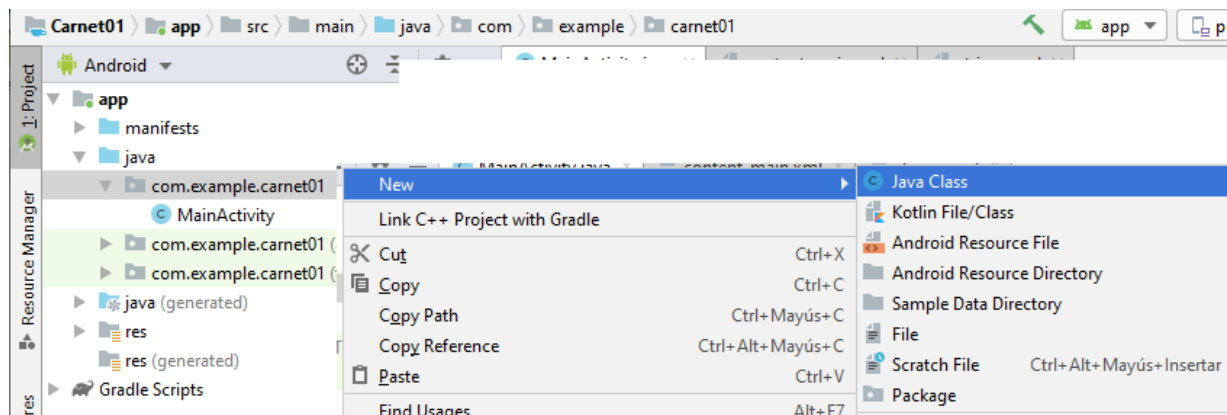
Con lo anterior se habrá implementado un método onClick de la interfaz onClickListener, cuyo nombre es igual, cámbielo por onclick que es el nombre que se definió en los botones,

Antes	Despues
<pre>View.OnClickListener onClickListener = new View.OnClickListener() { @Override public void onClick(View v) { } };</pre>	<pre>View.OnClickListener onclick = new View.OnClickListener() { @Override public void onclick(View v) { } };</pre>

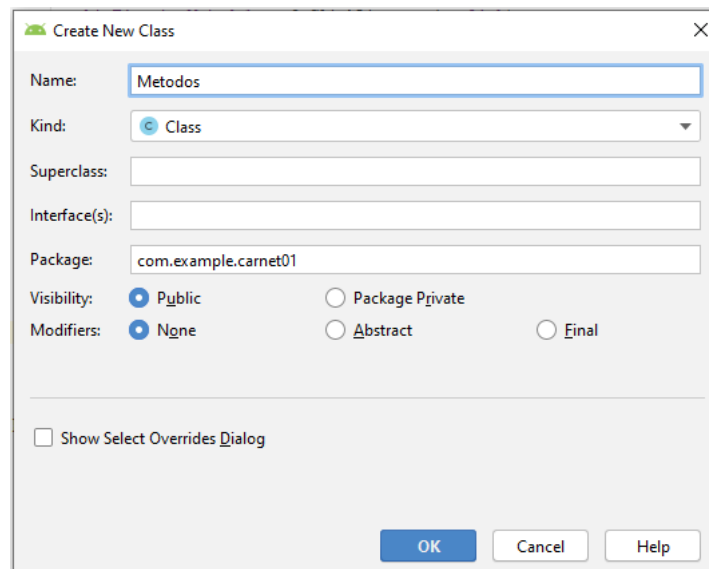
En Este momento introduciremos una clase externa(Metodos.java) para que realice las operaciones de Suma resta y multiplicación en diferentes métodos.

Creación de Metodos.java

Presionar clic derecho sobre el paquete, luego new, java class



Definir el nombre: **Metodos**, Luego presionar ok



Luego debe crear 3 métodos (Suma, Resta y Multiplicacion) con 2 parámetros tipo Float , que retornen un valor String.

Si no se recuerda de la sintaxis puede consultar el anexo 1

Luego regrese al Activity y digite el método click

```
View.OnClickListener onclick = new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        float numfloat1, numfloat2;  
        switch (v.getId()) {  
            case R.id.button:  
                numfloat1 = (float) Float.valueOf(edtParametro1.getText().toString());  
                numfloat2 = (float) Float.valueOf(edtParametro2.getText().toString());  
                sumastr = Metodos.suma(numfloat1, numfloat2);  
                lblEtiqueta.setText(sumastr);  
                break;  
            case R.id.button2:  
                numfloat1 = (float) Float.valueOf(edtParametro1.getText().toString());  
                numfloat2 = (float) Float.valueOf(edtParametro2.getText().toString());  
                restastr = Metodos.resta(numfloat1, numfloat2);  
                lblEtiqueta.setText(restastr);  
                break;  
            case R.id.button3:  
                numfloat1 = (int) Integer.valueOf(edtParametro1.getText().toString());  
                numfloat2 = (int) Integer.valueOf(edtParametro2.getText().toString());  
                multstr = Metodos.mult(numfloat1, numfloat2);  
                lblEtiqueta.setText(multstr);  
                break;  
        }  
    }  
};
```

Preguntas:

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

¿Es un error? _____

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

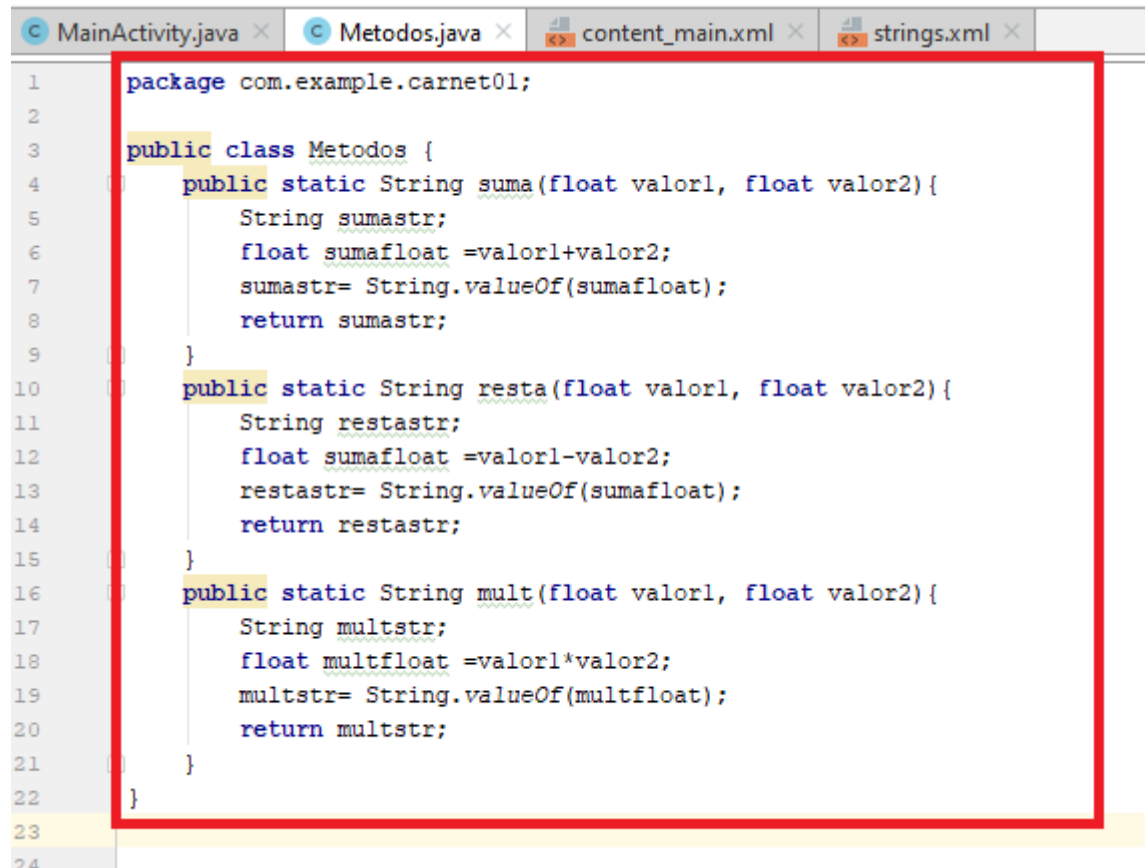
Tareas opcionales :

Sobrecargue el método de multiplicación para tres parámetros y haga que se ejecute solo cuando el parámetro 3 tenga un valor >0.

*Agréguele 2 idiomas adicionales al default(inglés y portugués)

Anexos

Anexo 1 Código de Metodos.java



```
1 package com.example.carnet01;
2
3 public class Metodos {
4     public static String suma(float valor1, float valor2){
5         String sumastr;
6         float sumafloat =valor1+valor2;
7         sumastr= String.valueOf(sumafloat);
8         return sumastr;
9     }
10    public static String resta(float valor1, float valor2){
11        String restastr;
12        float sumafloat =valor1-valor2;
13        restastr= String.valueOf(sumafloat);
14        return restastr;
15    }
16    public static String mult(float valor1, float valor2){
17        String multstr;
18        float multfloat =valor1*valor2;
19        multstr= String.valueOf(multfloat);
20        return multstr;
21    }
22 }
23
24
```