



UNIVERSIDAD PRIVADA DE TACNA



SOLUCIONES MÓVILES

Tema I

MSc. ING. WILSON CÉSAR CALLISAYA CHOQUECOTA

TEMARIO

- Ejecutar Hola mundo en Android
- Estructura de un proyecto en Android
- Ejemplo de Vistas básicas
- Ciclo de Vida de Android
- Manipulación de Elementos por Código
- Capturando valor del Spinner
- Github Android - Apidemos

Ejecutar Hola Mundo en Android



Plataforma

Android Studio

Google Play

Android Jetpack

Docs

News



Buscar

Android Studio

DOWNLOADWHAT'S NEWUSER GUIDEPREVIEW

android studio

Android Studio proporciona las herramientas más rápidas para crear apps en todas las clases de dispositivos Android.

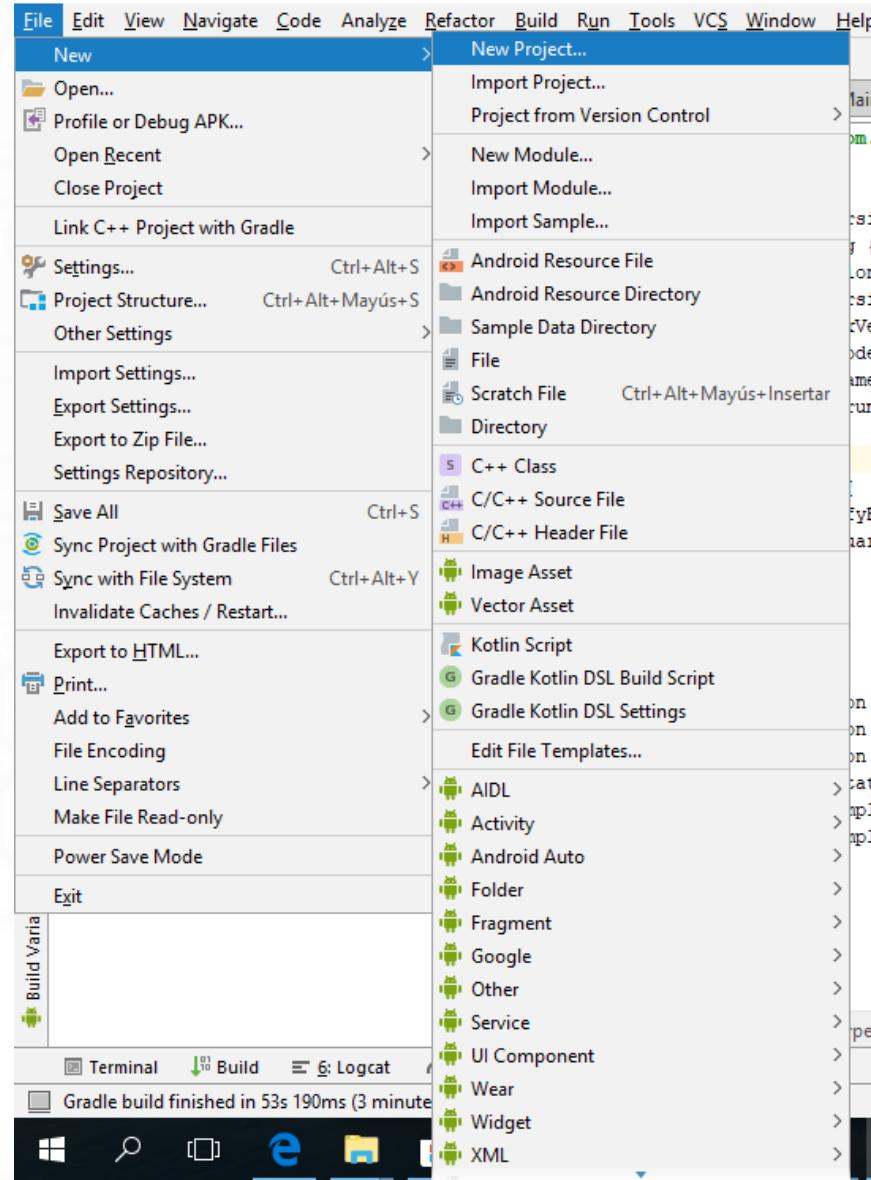
DESCARGA NO DISPONIBLE

Tu dispositivo actual no es compatible.
Consulta los requisitos del sistema.

OPCIONES DE DESCARGANOTAS DE LA VERSIÓN

<https://developer.android.com/studio/?hl=es-419>

MSc. Ing. Wilson César Callisaya Choquecota



```
mainActivity.java x miprimeraapp x app x
com.android.application'

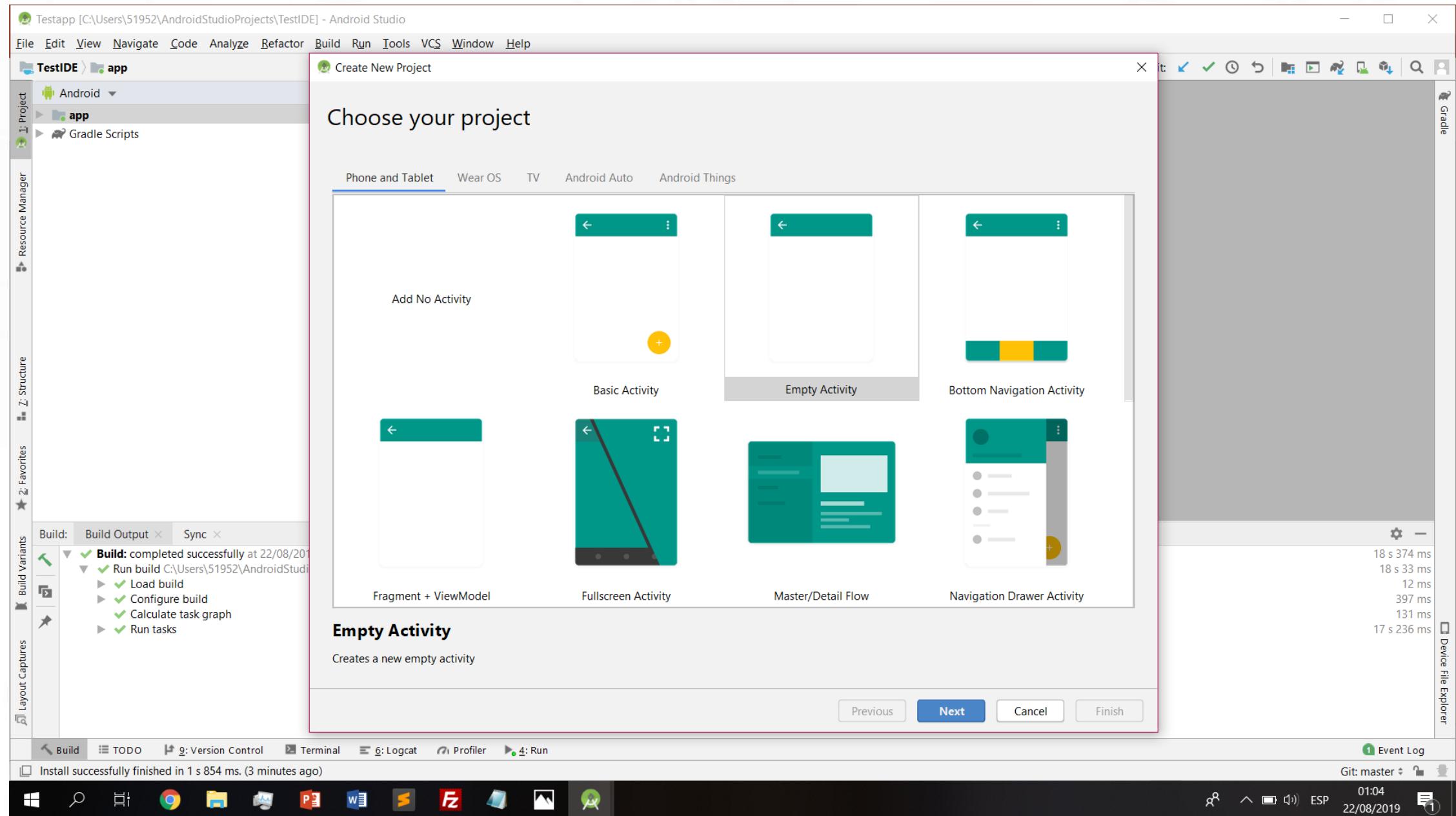
version 28
j {
ionId "com.example.toshiba.miprimeraapp"
ersion 15
Version 28
ode 1
ame "1.0"
umentationRunner "android.support.test.runner.AndroidJUnitRunner"

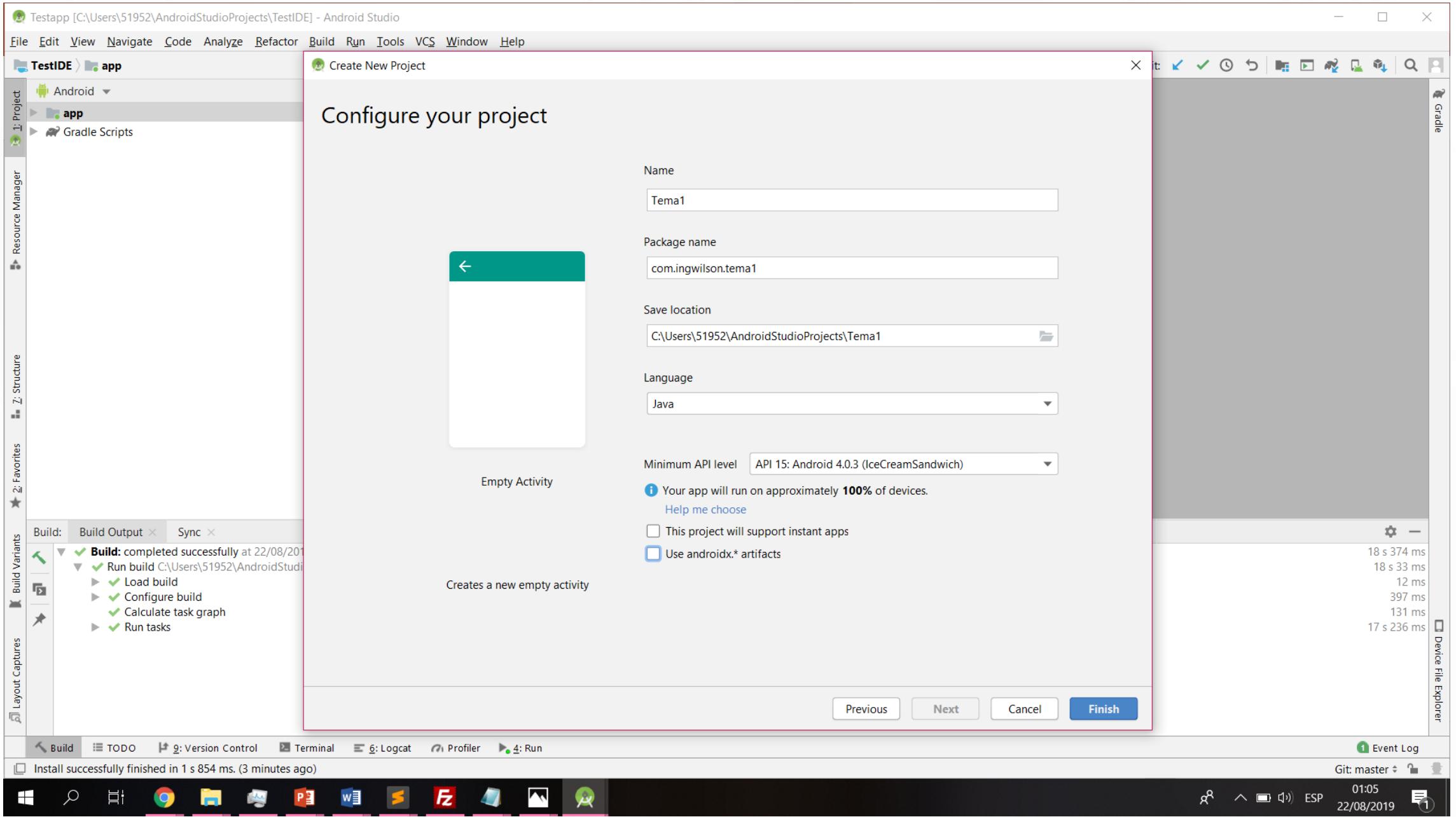
ayEnabled false
ardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

on fileTree(dir: 'libs', include: ['*.jar'])
on 'com.android.support:appcompat-v7:28.0.0'
on 'com.android.support.constraint:constraint-layout:1.1.3'
ation 'junit:junit:4.12'
plementation 'com.android.support.test:runner:1.0.2'
plementation 'com.android.support.test.espresso:espresso-core:3.0.2'

pes{}

13:17 CRLF+ UTF-8+ Context: <no context>
23:54
27/11/2018
```

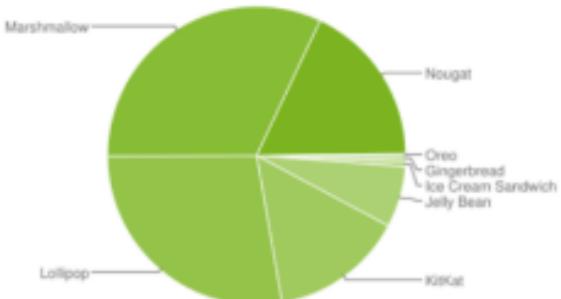






Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.6%
4.1.x	Jelly Bean	16	2.3%
4.2.x		17	3.3%
4.3		18	1.0%
4.4	KitKat	19	14.5%
5.0	Lollipop	21	6.7%
5.1		22	21.0%
6.0	Marshmallow	23	32.0%
7.0	Nougat	24	15.8%
7.1		25	2.0%
8.0	Oreo	26	0.2%

Data collected during a 7-day period ending on October 2, 2017.
Any versions with less than 0.1% distribution are not shown.



<https://developer.android.com/about/dashboards/>

miprimeraapp [C:\Users\TOSHIBA\AndroidStudioProjects\miprimeraapp] - app - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

miprimeraapp > app > build.gradle

Gradle Scripts

build.gradle (Project: miprimeraapp)

build.gradle (Module: app)

gradle-wrapper.properties (Gradle Version)

proguard-rules.pro (ProGuard Rules for app)

gradle.properties (Project Properties)

settings.gradle (Project Settings)

local.properties (SDK Location)

```
1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion 28
5     defaultConfig {
6         applicationId "com.example.toshiba.miprimeraapp"
7         minSdkVersion 15
8         targetSdkVersion 28
9         versionCode 1
10        versionName "1.0"
11        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12    }
13    buildTypes {
14        release {
15            minifyEnabled false
16            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
17        }
18    }
19}
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:28.0.0'
24     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
25     testImplementation 'junit:junit:4.12'
26     androidTestImplementation 'com.android.support.test:runner:1.0.2'
27     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
28}
29
```

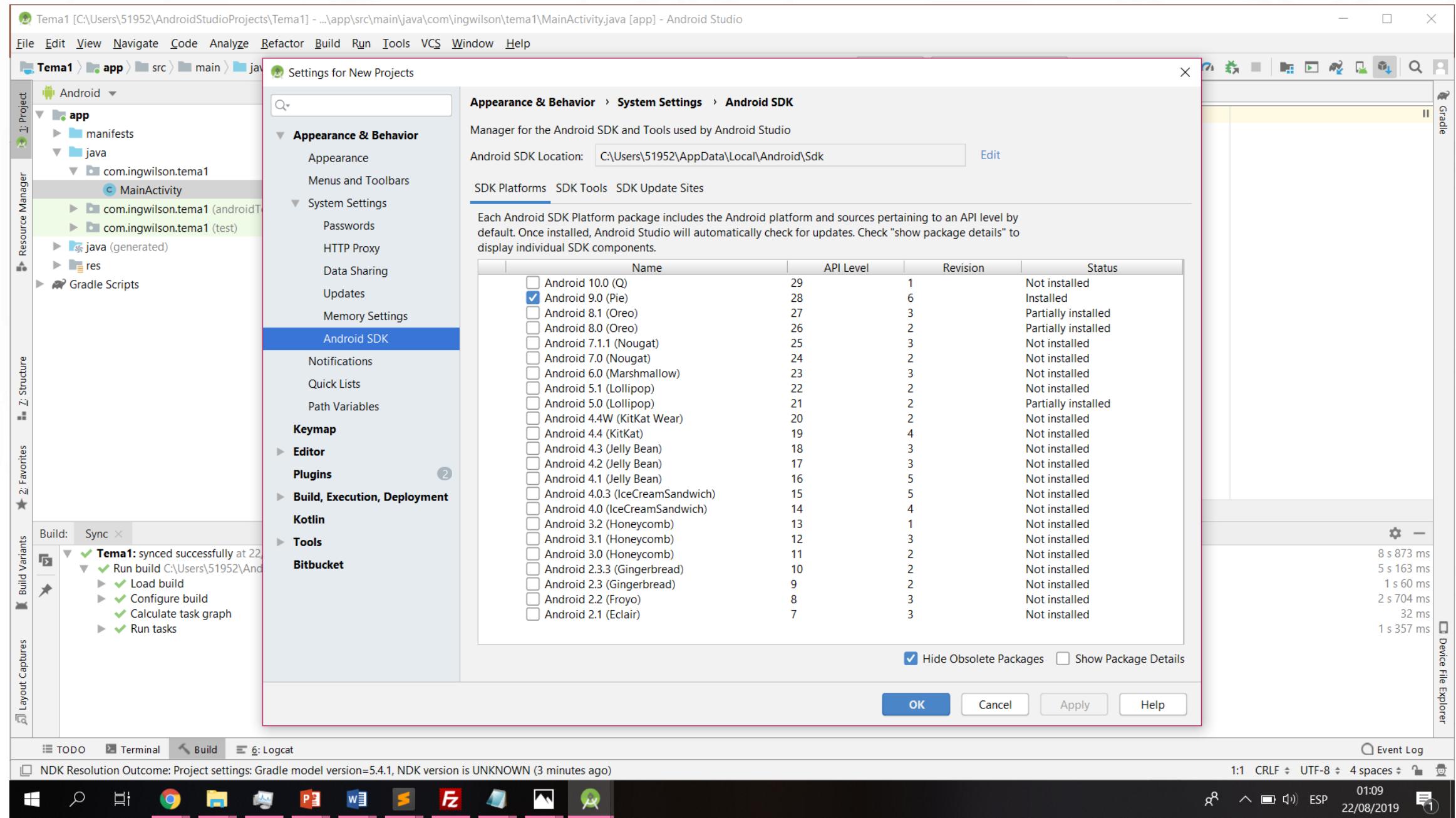
Device File Explorer

Terminal Build Logcat Android Profiler Run TODO Event Log

Gradle build finished in 53s 190ms (24 minutes ago) 13:17 CRLF UTF-8 Context: <no context>

Windows Start File Explorer Microsoft Edge Google Chrome File Microsoft Word Microsoft Excel Microsoft Powerpoint Microsoft Project Microsoft Visio

00:15 28/11/2018



- Ejecución en un terminal

1. Instala el driver y conecta el cable USB
2. En la configuración del terminal indica:



3. A partir de v. 4.2 las opciones para desarrolladores vienen ocultas. Para activarlas:
Ajustes > Información del dispositivo.
Pulsar siete veces sobre el número de compilación.



3% 16:19

3% 16:20

3% 16:20

AJUSTES



ACERCA DEL DISPOSITIVO



INFO. SOFTWARE



Predeterminadas, Permisos de aplicación



Bloqueo y seguridad

Pantalla de bloqueo, Huellas digitales, Loca...



Nube y cuentas

Samsung Cloud, Copiar y restaurar, Smart S...



Google

Ajustes de Google



Accesibilidad

Visión, Audición, Habilidad e interacción



Administración general

Idioma y entrada, Fecha y hora, Restablecer



Actualización de software

Descargar actualizaciones, Actualización pr...



Manual de usuario

Manual de usuario



Acerca del dispositivo

Estado, Información legal, Nombre del disp...

Mi número de teléfono

No disponible.

Estado

Ver el estado de la tarjeta SIM, el IMEI del dispositivo y demás información.

Información legal

Nombre del dispositivo

Samsung Galaxy S7 edge

Número de modelo

SM-G9350

Info. software

Ver la versión actual instalada de Android, la versión de banda base, la versión de kernel, el número de compilación, etc.

Información de la batería

Ver el estado de la batería del dispositivo, la carga restante y demás información.

¿ESTÁ BUSCANDO OTRA COSA?

Versión de Android

7.0

Versión de banda base

G9350ZHU2BQE3

Versión de Kernel

3.18.31-11118867
dpi@SWDG9723 #1
Thu May 25 23:09:25 KST 2017**PULSAR 7 VECES**

Número de compilación

NRD90M.G9350ZHU2BQE3

Estado de SE para Android

Enforcing
SEPF_SECMOBILE_7.0_0005
Thu May 25 23:14:18 2017

Versión de seguridad

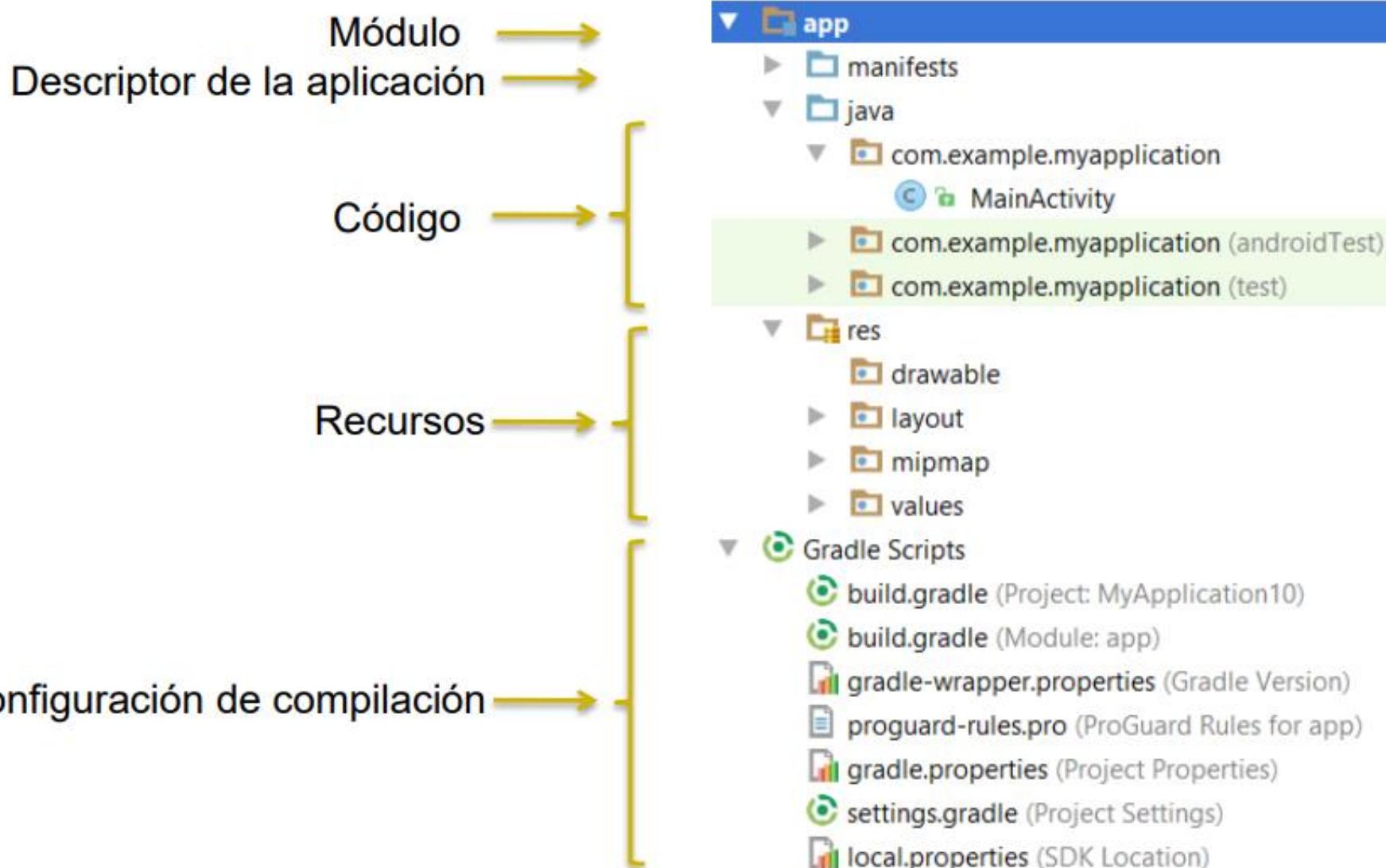
ASKS v1.3_161228

Versión de Knox

Knox 2.7.1
Standard SDK 5.7.1
Premium SDK 2.7.1
Customization SDK 2.7.1

ESTRUCTURA DE UN PROYECTO EN ANDROID

- El SDK ha creado una serie de carpetas y ficheros:



Actividad Principal

PrimeraPractica [C:\Users\51952\AndroidStudioProjects\PrimeraPractica] - ...\\app\\src\\main\\java\\com\\ingwilson\\primerapractica\\MainActivity.java [app] - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

PrimeraPractica > app > src > main > java > com > ingwilson > primerapractica > MainActivity

Android Project Resource Manager Build Variants

activity_main.xml MainActivity.java app

```
1 package com.ingwilson.primerapractica;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14
15 }
```

Project

Resource Manager

Build Variants

Structure

Text



activity_main.xml

MainActivity.java

app

app
 manifests
 java
 com.ingwilson.primerapractica
 MainActivity
 com.ingwilson.primerapractica (androidTest)
 com.ingwilson.primerapractica (test)
 java (generated)
 res
 drawable
 layout
 activity_main.xml
 mipmap
 values

Gradle Scripts
 build.gradle (Project: PrimeraPractica)
 build.gradle (Module: app)
 gradle-wrapper.properties (Gradle Version)
 proguard-rules.pro (ProGuard Rules for app)
 gradle.properties (Project Properties)
 settings.gradle (Project Settings)
 local.properties (SDK Location)

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android=
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="Hello World!"
13         app:layout_constraintBottom_toBottomOf="parent"
14         app:layout_constraintLeft_toLeftOf="parent"
15         app:layout_constraintRight_toRightOf="parent"
16         app:layout_constraintTop_toTopOf="parent" />
17
18  </androidx.constraintlayout.widget.ConstraintLayout>
```

androidx.constraintlayout.widget.ConstraintLayout

Design

Text

miprimeraapp [C:\Users\TOSHIBA\AndroidStudioProjects\miprimeraapp] - ...\\app\\src\\main\\AndroidManifest.xml [app] - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

miprimeraapp app src main AndroidManifest.xml

1: Project 1: miprimeraapp

1.1: miprimeraapp

1.1.1: app

1.1.1.1: manifests

1.1.1.1.1: AndroidManifest.xml

1.1.1.2: java

1.1.1.2.1: com.example.toshiba.miprimeraapp

1.1.1.2.1.1: MainActivity

1.1.1.2.2: com.example.toshiba.miprimeraapp (androidTest)

1.1.1.2.3: com.example.toshiba.miprimeraapp (test)

1.1.1.3: res

1.1.1.3.1: drawable

1.1.1.3.2: layout

1.1.1.3.2.1: activity_main.xml

1.1.1.3.3: mipmap

1.1.1.3.4: values

1.1.1.4: Gradle Scripts

1.1.1.4.1: build.gradle (Project: miprimeraapp)

1.1.1.4.2: build.gradle (Module: app)

1.1.1.4.3: gradle-wrapper.properties (Gradle Version)

1.1.1.4.4: proguard-rules.pro (ProGuard Rules for app)

1.1.1.4.5: gradle.properties (Project Properties)

1.1.1.4.6: settings.gradle (Project Settings)

1.1.1.4.7: local.properties (SDK Location)

1.1.2: Activity

1.1.2.1: activity_main.xml

1.1.2.2: AndroidManifest.xml

1.1.2.3: MainActivity.java

1.1.2.4: miprimeraapp

1.1.2.5: app

1.1.3: Gradle

1.2: Structure

1.3: Captures

1.4: Favorites

1.5: Build Variants

1.6: Device File Explorer

Text Merged Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.toshiba.miprimeraapp">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="miprimeraapp"
        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>
```

Terminal Build Logcat Android Profiler Run TODO Event Log

Gradle build finished in 53s 190ms (38 minutes ago)

1:1 CRLF UTF-8 Context: <no context>

00:29 28/11/2018



1: Project

1: Structure

Captures

2: Favorites

2: Build Variants

resources

```
<resources>
    <string name="app_name">miprimeraapp</string>
</resources>
```

Open editor Hide notification

Gradle

Device File Explorer

build.gradle (Project: miprimeraapp)

build.gradle (Module: app)

gradle-wrapper.properties (Gradle Version)

proguard-rules.pro (ProGuard Rules for app)

gradle.properties (Project Properties)

settings.gradle (Project Settings)

local.properties (SDK Location)



CARPETAS DENTRO DE RES

- **drawable**: ficheros de imágenes y descriptores de imágenes (.png, .gif, .jpg, .xml).
- **mipmap**: igual que drawables pero no pueden ser reescalados por el sistema.
- **layout**: ficheros XML con vistas de la aplicación.
Las vistas nos permitirán configurar las diferentes pantallas que compondrán la interfaz de usuario de la aplicación.
- **menu**: ficheros XML con los menús de la aplicación.
- **values**: contiene los ficheros *string.xml*, *color.xml* y *style.xml*. De esta manera podremos cambiar los valores de cadenas, colores y estilos sin necesidad de ir al código fuente.



- Puede accederse a los recursos por medio de un identificador desde el código Java utilizando la clase *R*.

```
setContentView(R.layout.main_activity);
```

- Para acceder desde XML:

```
android:label="@string/app_name"
```

miprimeraapp [C:\Users\TOSHIBA\AndroidStudioProjects\miprimeraapp] - app - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

miprimeraapp app build.gradle

1: Project 2: Structure Captures 3: Favorites Build Variants

Gradle Scripts

build.gradle (Project: miprimeraapp)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)

activity_main.xml AndroidManifest.xml strings.xml MainActivity.java miprimeraapp app

1 apply plugin: 'com.android.application'
2 android {
3 compileSdkVersion 28
4 defaultConfig {
5 applicationId "com.example.toshiba.miprimeraapp"
6 minSdkVersion 15
7 targetSdkVersion 28
8 versionCode 1
9 versionName "1.0"
10 testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
11 }
12 buildTypes {
13 release {
14 minifyEnabled false
15 proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
16 }
17 }
18}
19
20 dependencies {
21 implementation fileTree(dir: 'libs', include: ['*.jar'])
22 implementation 'com.android.support:appcompat-v7:28.0.0'
23 implementation 'com.android.support.constraint:constraint-layout:1.1.3'
24 testImplementation 'junit:junit:4.12'
25 androidTestImplementation 'com.android.support.test:runner:1.0.2'
26 androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
27 }
28 }
29

Device File Explorer

Terminal Build Logcat Android Profiler Run TODO Event Log

Gradle build finished in 53s 190ms (44 minutes ago) 29:1 CRLF UTF-8 Context: <no context>

Windows Start File Explorer Google Chrome Microsoft Edge Microsoft Word Microsoft Excel Microsoft Powerpoint Microsoft OneDrive

00:36 28/11/2018

Componentes de una aplicación Android

- *Vista (View)*
- *Layout*
- *Actividad (Activity)*
- *Servicio (Service)*
- *Fragment*
- *Intención (Intent)*
- *Receptor de anuncios (Broadcast receiver)*
- *Proveedor de contenido (Content provider)*

Ejemplo de Vistas básicas

Ejemplo 1 – Reemplazar el activity_main.xml

The screenshot shows the Android Studio interface with the project 'Taller1' open. The 'activity_main.xml' file is selected in the Project Structure view. The code editor displays the XML layout for a ConstraintLayout:

```
<?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">

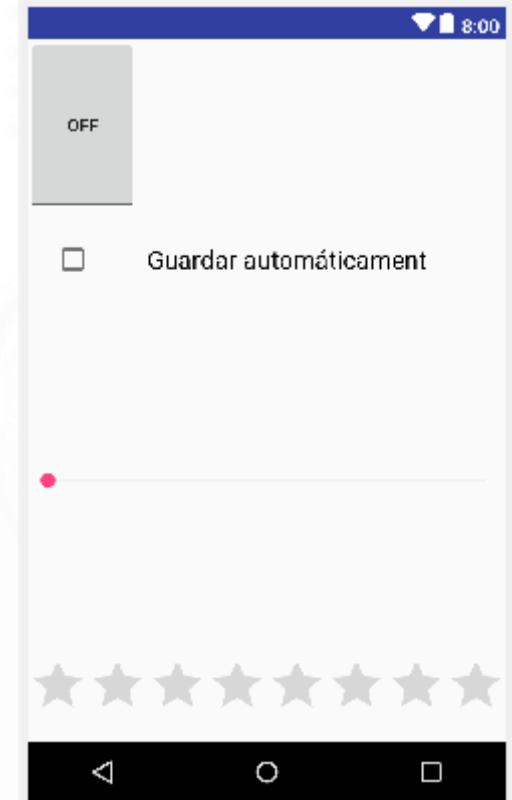
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

The code editor includes tabs for 'Design' and 'Text'. The bottom status bar shows the terminal output: 'Gradle build finished in 18s 896ms (2 minutes ago)'. The system tray at the bottom right shows the date and time as '31/01/2019'.

Ejemplo 1 – Reemplazar el activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
>
    <ToggleButton
        android:id="@+id/toggleButton"
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_weight="2"
        android:text="ToggleButton" />
    <CheckBox
        android:id="@+id/checkBox"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_margin="20dp"
        android:layout_weight="0.5"
        android:gravity="center"
        android:text="Guardar automáticamente"
        android:textSize="9pt" />
    <SeekBar
        android:id="@+id/seekBar"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="4" />
    <RatingBar
        android:id="@+id/ratingBar"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />
</LinearLayout>
```

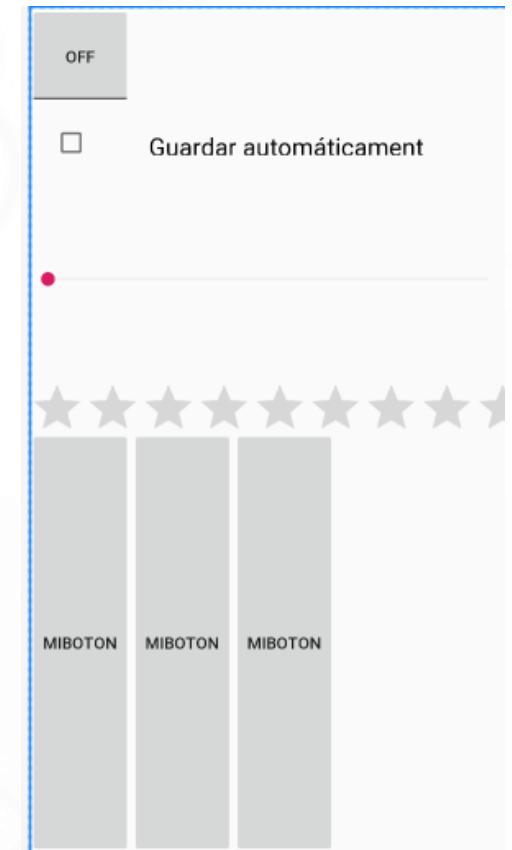


Ejemplo 1 – Insertamos las cuatro Views en un nuevo LinearLayout

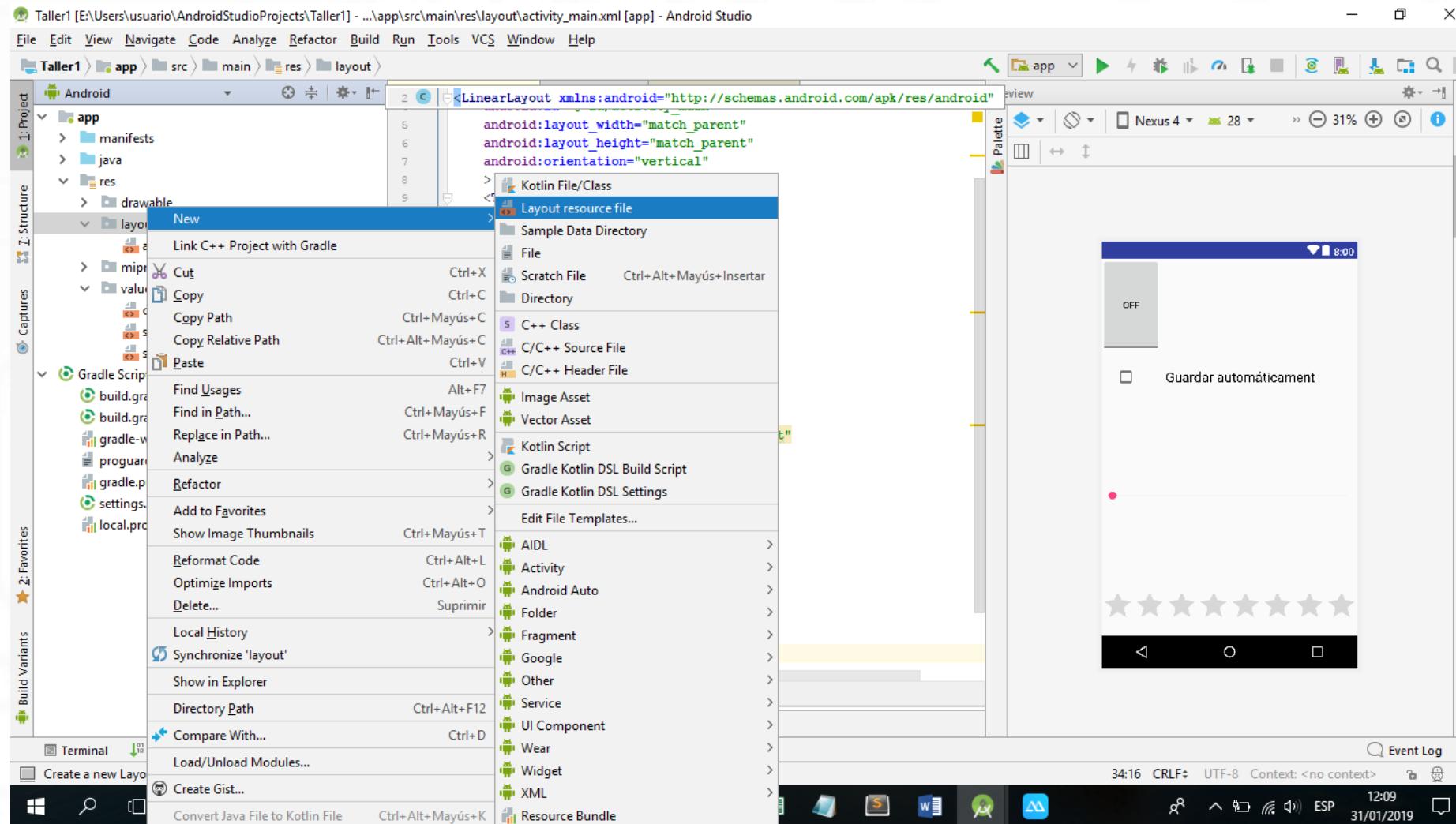
```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1"  
    android:orientation="vertical">  
    <ToggleButton  
        android:id="@+id/toggleButton"  
        android:layout_width="wrap_content"  
        android:layout_height="0dp"  
        android:layout_weight="2"  
        android:text="ToggleButton" />  
    <CheckBox  
        android:id="@+id/checkBox"  
        android:layout_width="match_parent"  
        android:layout_height="0dp"  
        android:layout_margin="20dp"  
        android:layout_weight="0.5"  
        android:gravity="center"  
        android:text="Guardar automáticamente"  
        android:textSize="9pt" />  
    <SeekBar  
        android:id="@+id/seekBar"  
        android:layout_width="match_parent"  
        android:layout_height="0dp"  
        android:layout_weight="4" />  
    <RatingBar  
        android:id="@+id/ratingBar"  
        android:layout_width="match_parent"  
        android:layout_height="0dp"  
        android:layout_weight="1" />  
</LinearLayout>
```

Ejemplo 1 – Reemplazar el activity_main.xml

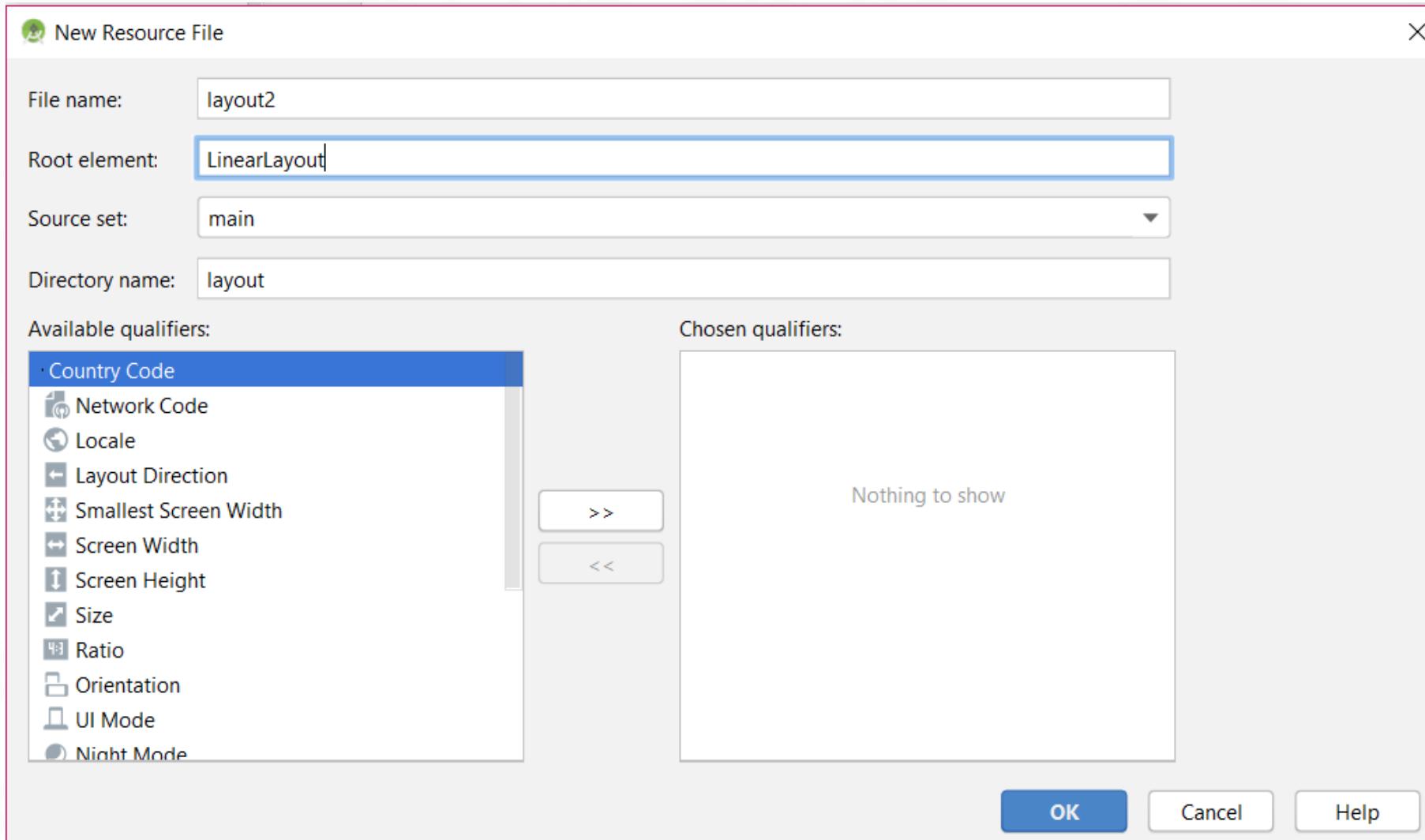
```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1"  
    android:orientation="horizontal">  
    <Button  
        android:layout_width="wrap_content"  
        android:layout_height="match_parent"  
        android:text="miboton">  
    </Button>  
    <Button  
        android:layout_width="wrap_content"  
        android:layout_height="match_parent"  
        android:text="miboton">  
    </Button>  
    <Button  
        android:layout_width="wrap_content"  
        android:layout_height="match_parent"  
        android:text="miboton">  
    </Button>  
</LinearLayout>
```



Ejemplo 2 – Creamos un nuevo layout



Ejemplo 2 – Creamos un nuevo layout



Ejemplo 2 – Reemplazamos layout2.xml

```
<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TableRow>

        <AnalogClock
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <CheckBox
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Un checkBox" />
    </TableRow>

    <TableRow>

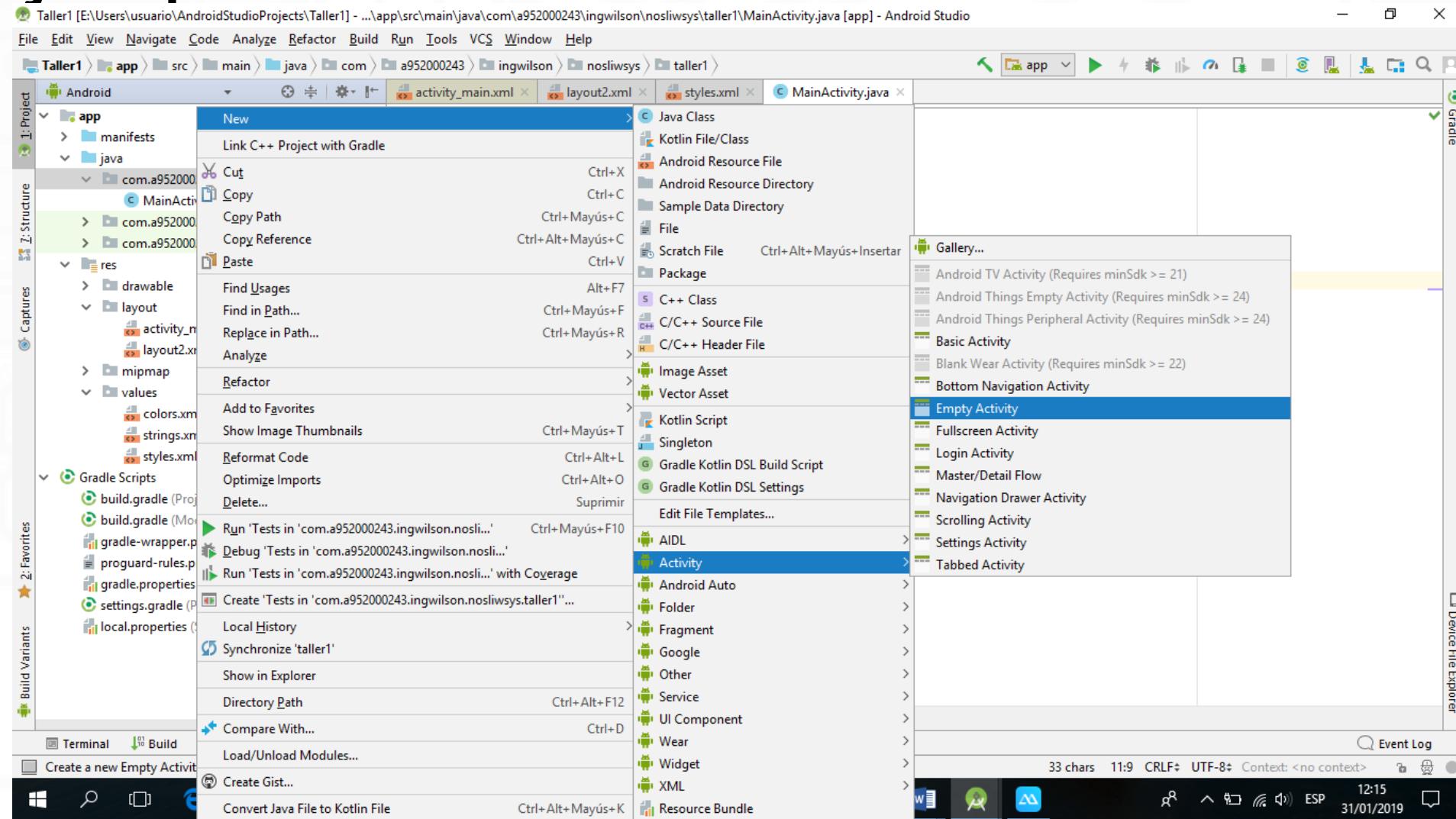
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Un botón" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Un texto cualquiera" />
    </TableRow>
</TableLayout>
```

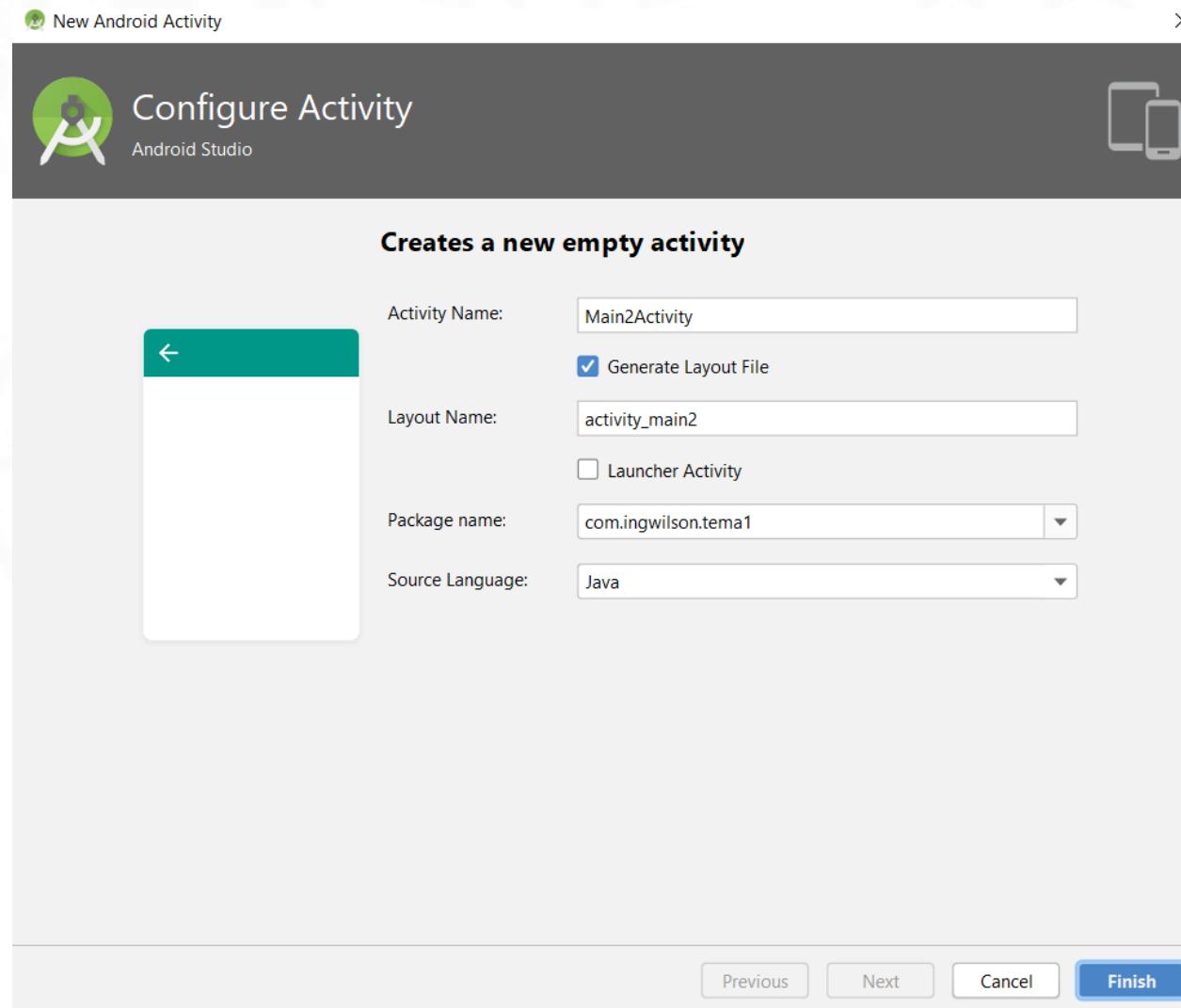
Ejemplo 2 – Seleccionar layout2 en el MainActivity.class

```
setContentView(R.layout.layout2) ;
```

Ejemplo 3 – Creamos una nueva Actividad



Ejemplo 3 – Creamos una nueva Actividad



Ejemplo 3 – Modificamos activity_main2.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <AnalogClock
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Un checkBox" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Un botón"
        android:visibility="invisible" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Un texto cualquiera"
        android:visibility="invisible" />
</FrameLayout>
```

Ejemplo 3 – Modificamos el Manifest

```
<activity android:name=".Main2Activity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".MainActivity"></activity>
```

Ejemplo 4

- CREAMOS UNA NUEVA ACTIVIDAD, EMPTY ACTIVITY, CON EL NOMBRE : Actividad3
- Al final dar la categoría Launcher en el Manifest

Ejemplo 4 – Modificamos activity_actividad3.xml

```
<LinearLayout xmlns:android  
    ="http://schemas.android.com/apk/res/android"  
    xmlns:tools  
    ="http://schemas.android.com/tools"  
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:gravity="center"  
    android:padding="30dp"  
    >  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="@string/tituloAplicacion"  
        android:gravity="center"  
        android:textSize="25sp"  
        android:layout_marginBottom="20dp"/>  
    <Button android:id="@+id/button01"  
        android:layout_height="wrap_content"  
        android:layout_width="match_parent"  
        android:text="@string/opcion1"  
        android:onClick="actividad2"  
        />  
    <Button android:id="@+id/button02"  
        android:layout_height="wrap_content"  
        android:layout_width="match_parent"  
        android:text="@string/opcion2"/>  
    <Button android:id="@+id/button03"  
        android:layout_height="wrap_content"  
        android:layout_width="match_parent"  
        android:text="@string/Salir"/>  
/</LinearLayout>
```

Ejemplo 4 – Modificamos String.xml

```
<resources>
    <string name="opcion1">Actividad 2</string>
    <string name="opcion2">Manipulando elementos por codigo</string>
    <string name="Salir">Salir</string>
    <string name="tituloAplicacion">Mi primera app</string>
    <string name="app_name">Mi app</string>
    <string name="action_settings">Configuración</string>
</resources>
```

Ejemplo 4 –

android:onClick="actividad2"

```
public void actividad2(View view) {
    startActivity(new Intent(this, Main2Activity.class));
}
```

Presionamos alt + intro para realizar corrección rápida de
proyecto (mostrar acciones de intención y correcciones rápidas)

Ejemplo 5

- Generamos una actividad llamada MiSpinner
- Añadimos un botón adicional para realizar el llamado a MiSpinner en la botonera del Ejemplo4 (Actividad3)
- Generamos un nuevo String para dicho boton
- No olvidar generar el Onclick respectivo
- Compile para verificar funcionamiento
- Luego reemplazar por el código siguiente

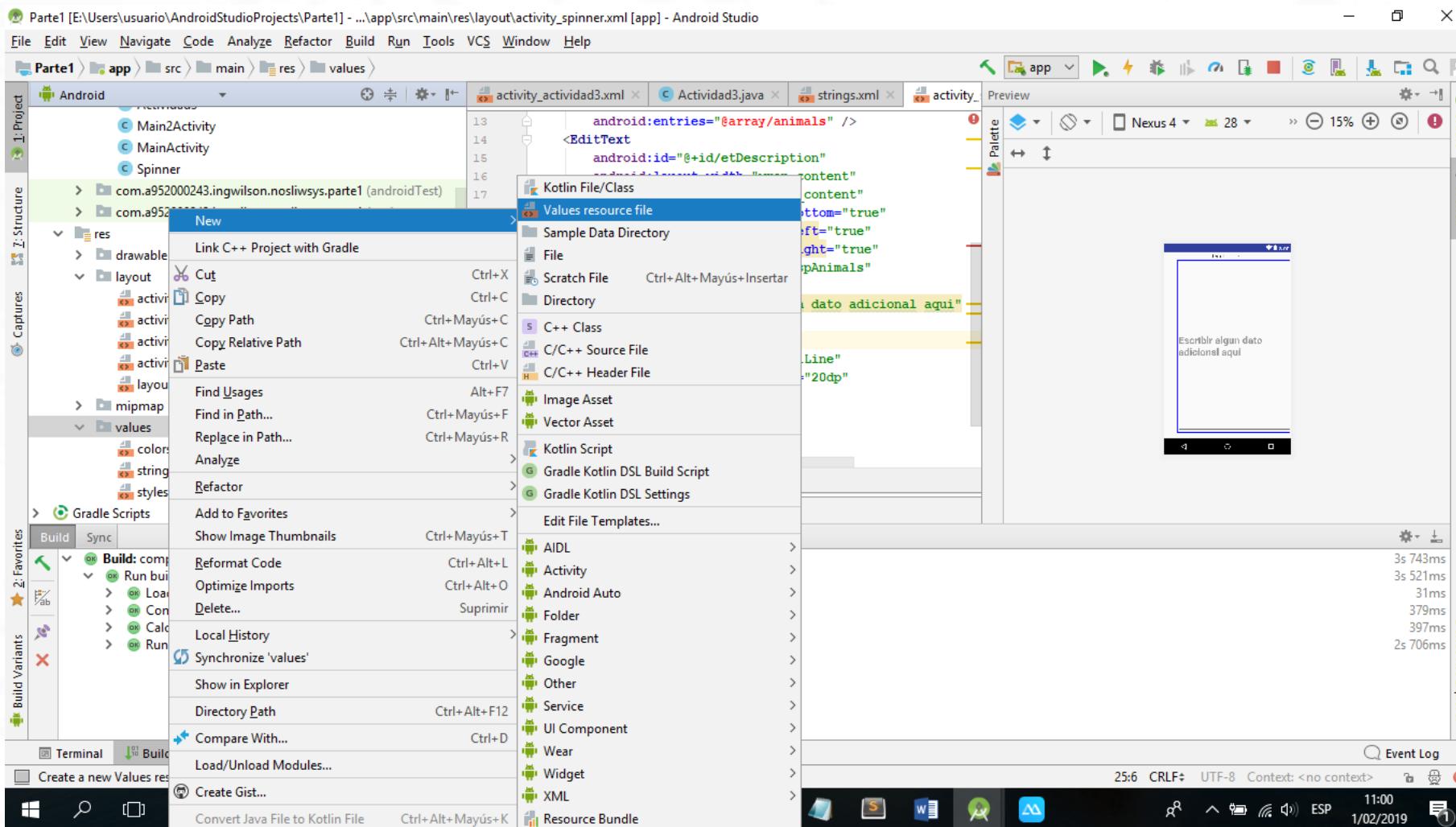
Ejemplo 5 – activity_mi_spinner.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="40dp"
    >
    <Spinner
        android:id="@+id/spAnimals"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:entries="@array/animals" />

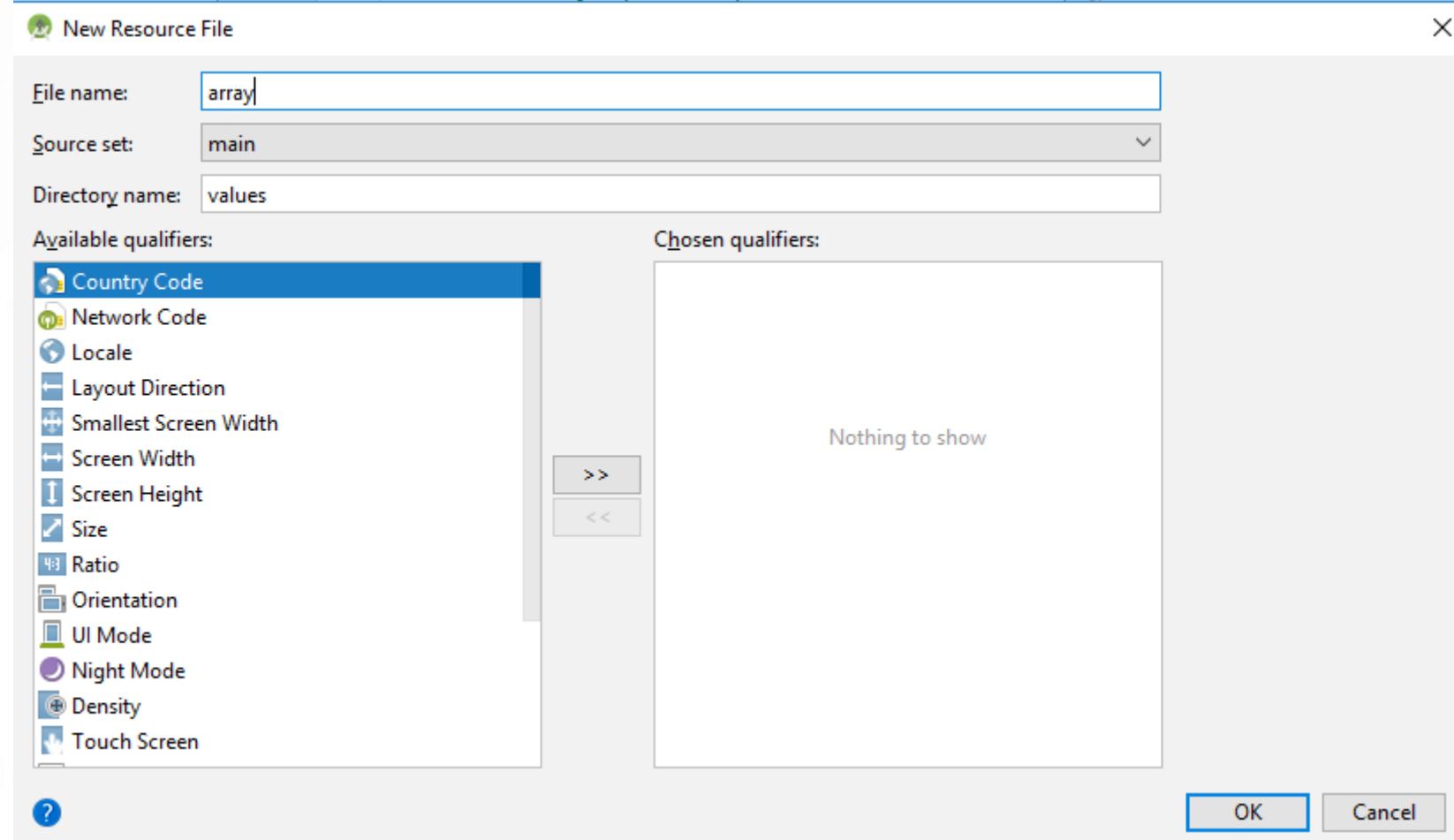
    <EditText
        android:id="@+id/etDescription"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/spAnimals"
        android:textStyle="bold"
        android:hint="Escribir algun dato adicional aqui"
        android:textSize="30sp"
        android:focusable="false"
        android:inputType="textMultiLine"
        android:layout_marginBottom="20dp"
        />

</RelativeLayout>
```

Ejemplo 5 – Crear un nuevo recurso



Ejemplo 5 – Ponemos de nombre array



Ejemplo 5 – values/array.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="animals">
        <item>Perro</item>
        <item>Gato</item>
        <item>Caballo</item>
        <item>Canario</item>
        <item>Vaca</item>
        <item>Cerdo</item>
    </string-array>
</resources>
```

Iniciando Actividades

MSc. Ing. Wilson César Callisaya Choquecota

Iniciar otra Actividad

```
public class MainActivity extends AppCompatActivity {

    Button miboton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        miboton=(Button)findViewById(R.id.button03);
        miboton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getApplicationContext(), RecyclerView_activity.class));
            }
        });
    }
    public void actividad2(View view){
        startActivity(new Intent(this, fragmenttabhost_activity.class));
    }
}
```

Crear una Actividad

- CREAR UNA Java Class
- CREAR UN XML Y VINCULARLO A LA ACTIVIDAD JAVA
- DECLARAR LA CLASE JAVA EN EL MANIFEST

```
<activity android:name=".RecyclerView.RecyclerView_activity"
    android:launchMode="singleTask"
    android:screenOrientation="portrait">
</activity>
```

CICLO DE VIDA DE ANDROID

Ciclo de vida

- Creamos la Actividad AcercaDe

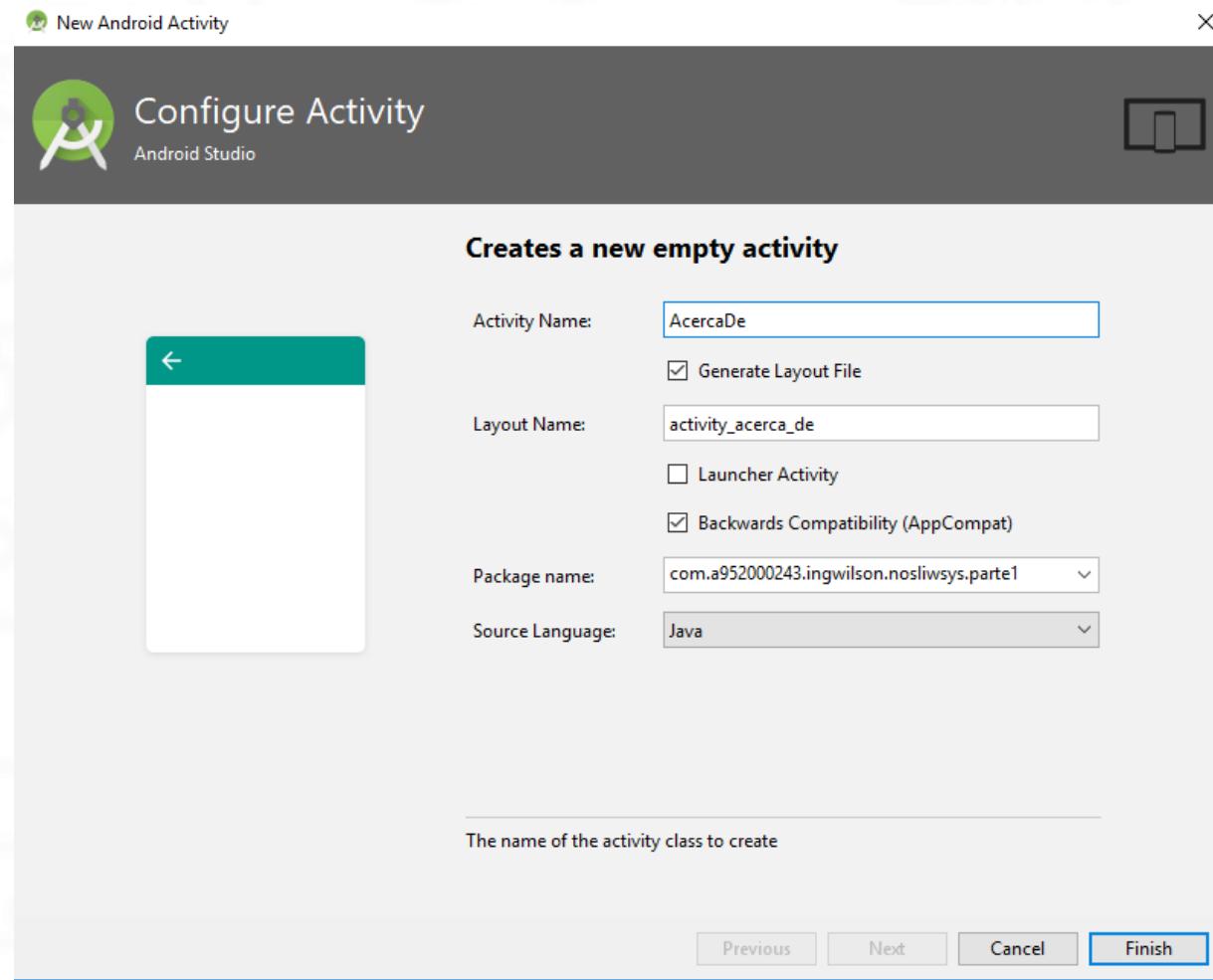
Añadimos los siguientes botones al xml del ejemplo4(Actividad3) y añadimos el string

```
<Button  
    android:layout_height="wrap_content"  
    android:layout_width="match_parent"  
    android:text="@string/dialogo"  
    android:onClick="dialogo"/>  
  
<Button  
    android:layout_height="wrap_content"  
    android:layout_width="match_parent"  
    android:text="@string/Acercade"  
    android:onClick="AcercaDe"/>  
  
<string name="dialogo">Mi dialogo</string>  
<string name="Acercade">Acerca de </string>
```

Añadimos al final de Actividad3.class

```
@Override protected void onStart() {
    super.onStart();
    Toast.makeText(this, "onStart", Toast.LENGTH_SHORT).show();
}
@Override protected void onResume() {
    super.onResume();
    Toast.makeText(this, "onResume", Toast.LENGTH_SHORT).show();
}
@Override protected void onPause() {
    Toast.makeText(this, "onPause", Toast.LENGTH_SHORT).show();
    super.onPause();
}
@Override protected void onStop() {
    super.onStop();
    Toast.makeText(this, "onStop", Toast.LENGTH_SHORT).show();
}
@Override protected void onRestart() {
    super.onRestart();
    Toast.makeText(this, "onRestart", Toast.LENGTH_SHORT).show();
}
@Override protected void onDestroy() {
    Toast.makeText(this, "onDestroy", Toast.LENGTH_SHORT).show();
    super.onDestroy();
}
public void dialogo(View view){
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("Este es mi dialogo")
        .setTitle("Mi primer dialogo...")
        .setCancelable(false)
        .setNeutralButton("Aceptar",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    dialog.cancel();
                }
            });
    AlertDialog alert = builder.create();
    alert.show();
}
```

Creamos la actividad AcercaDe



Modificamos su XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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=".AcercaDe">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Mi primera aplicacion version 1.0"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Modificamos su Java Class

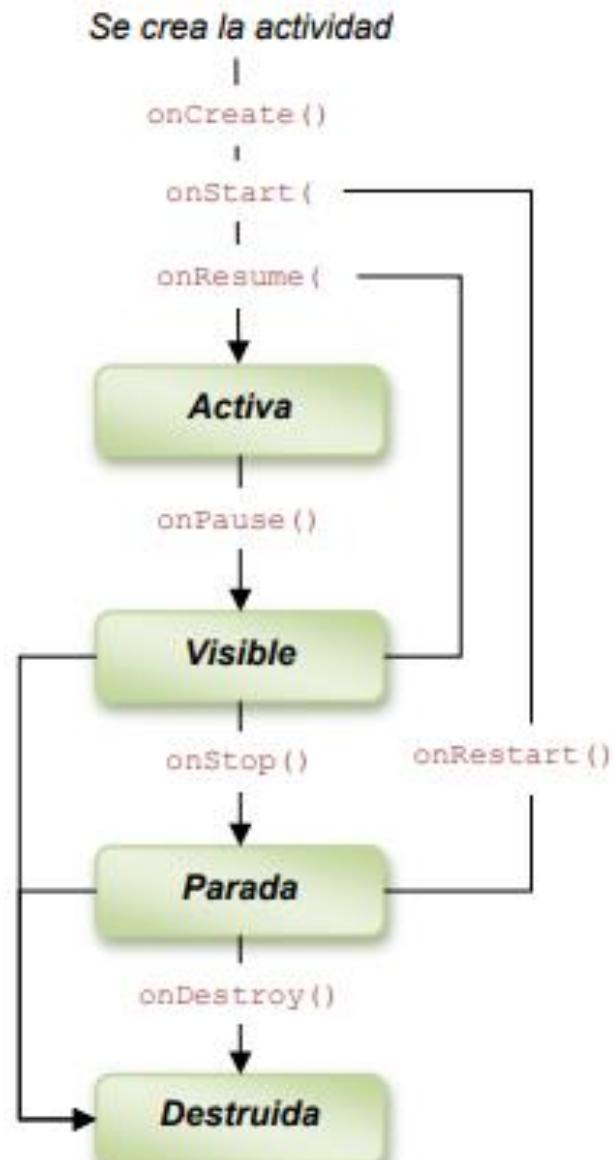
```
public class AcercaDe extends AppCompatActivity {
```

por

```
public class AcercaDe extends Activity {
```

Modificamos el Manifest

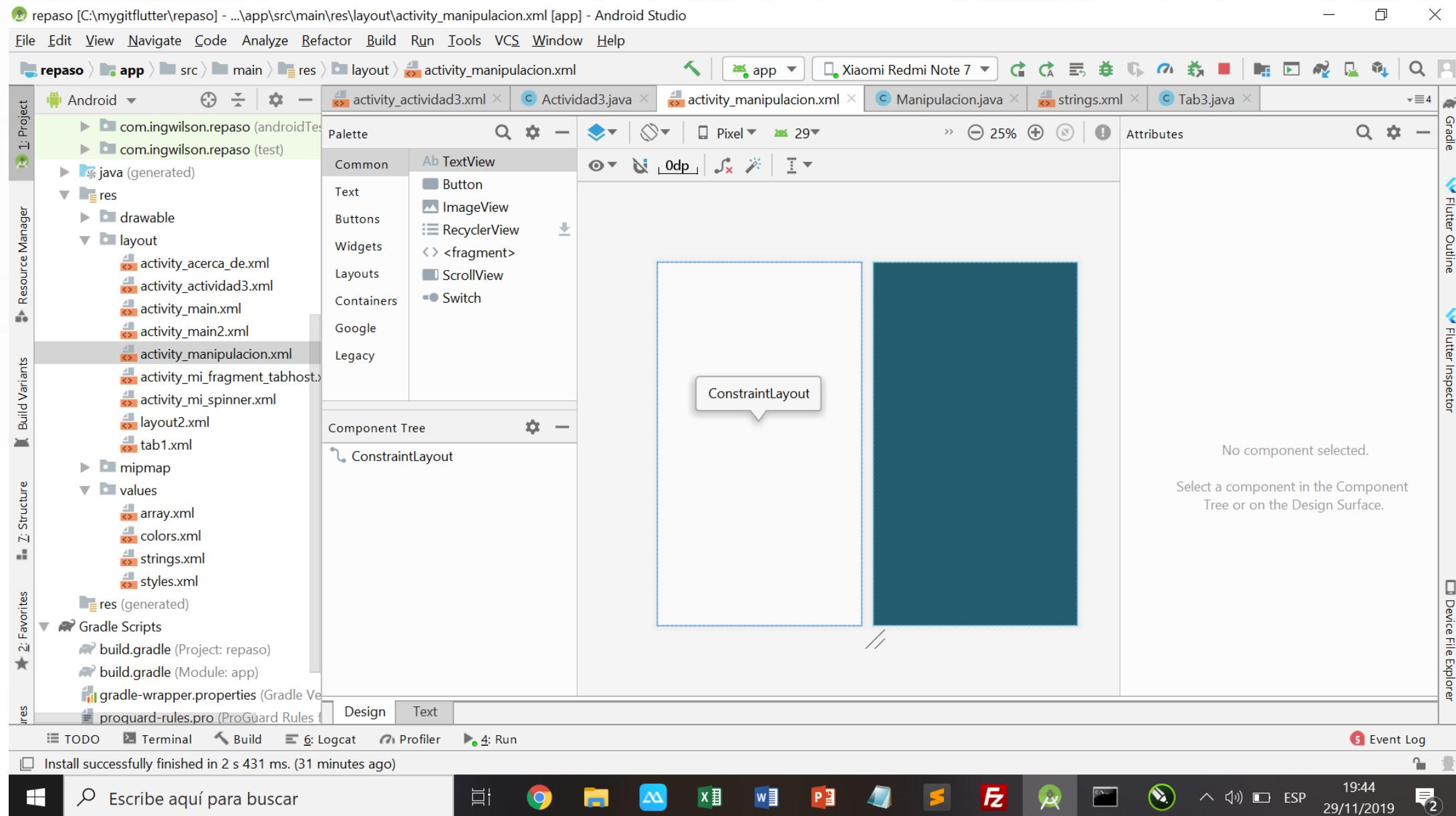
```
<activity android:name=".AcercaDe"
    android:label="Acerca de . . ."
    android:theme="@android:style/Theme.Dialog"></activity>
```

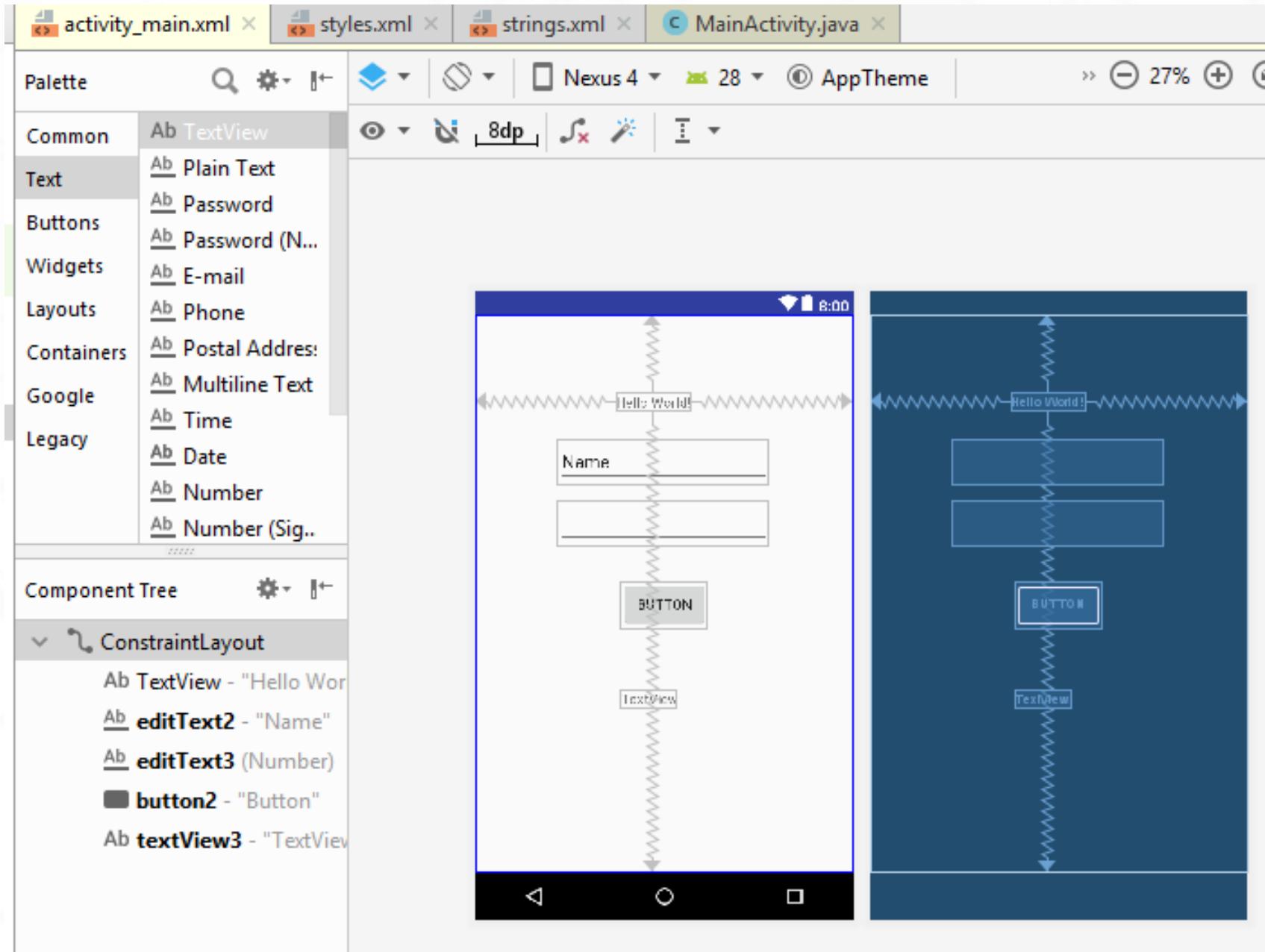


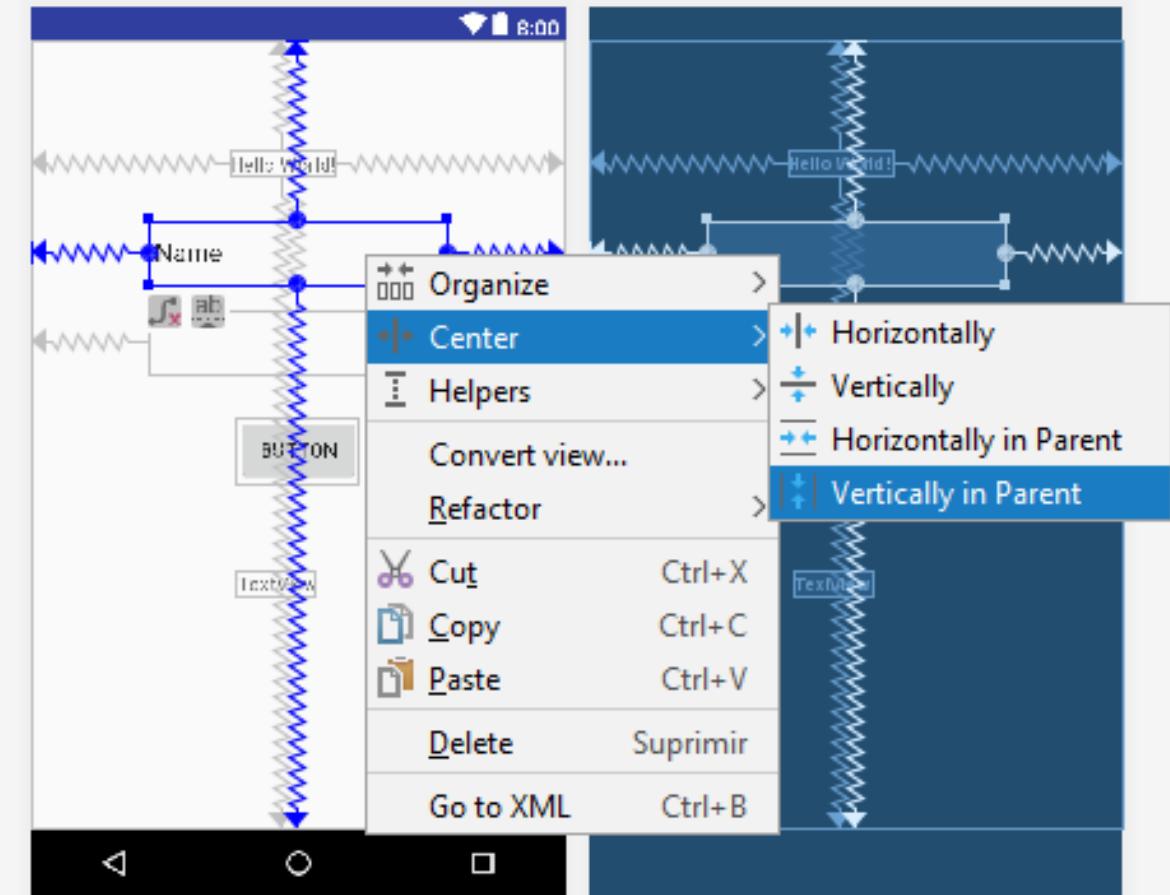
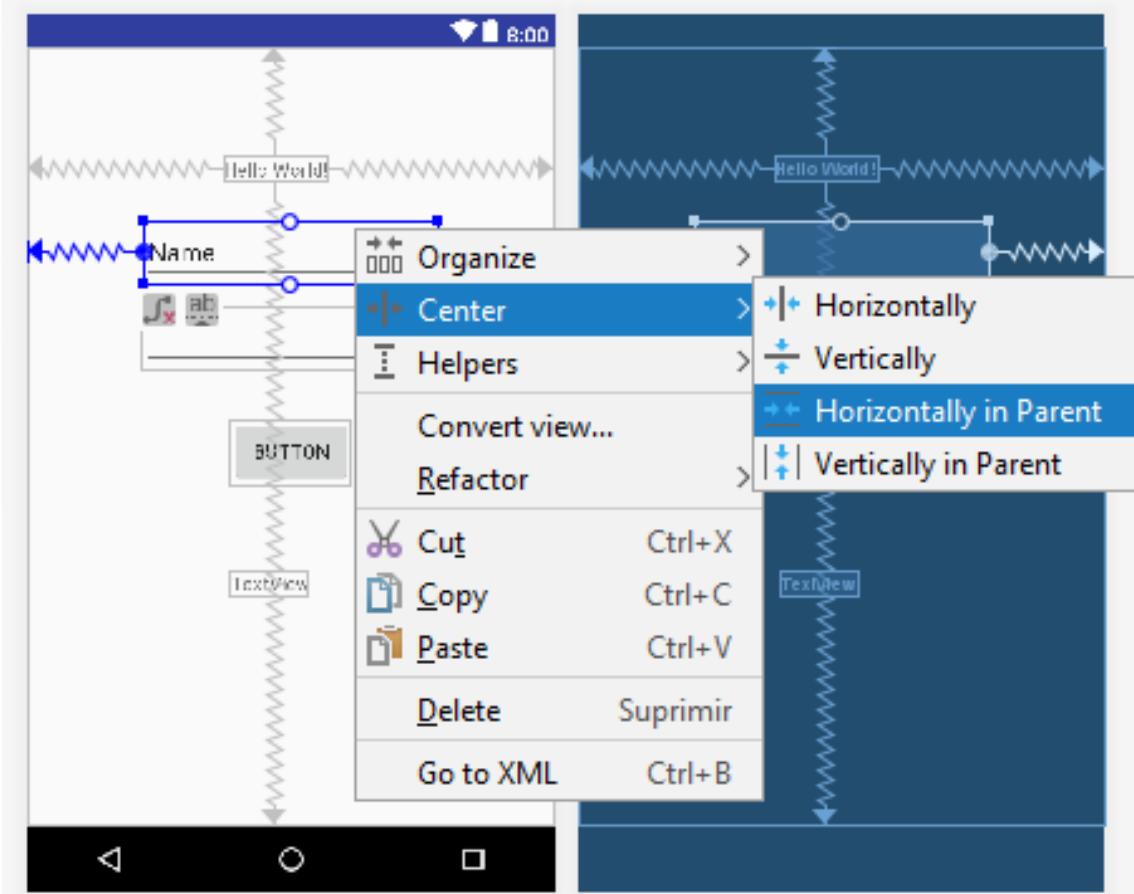
Manipulación de elementos por código

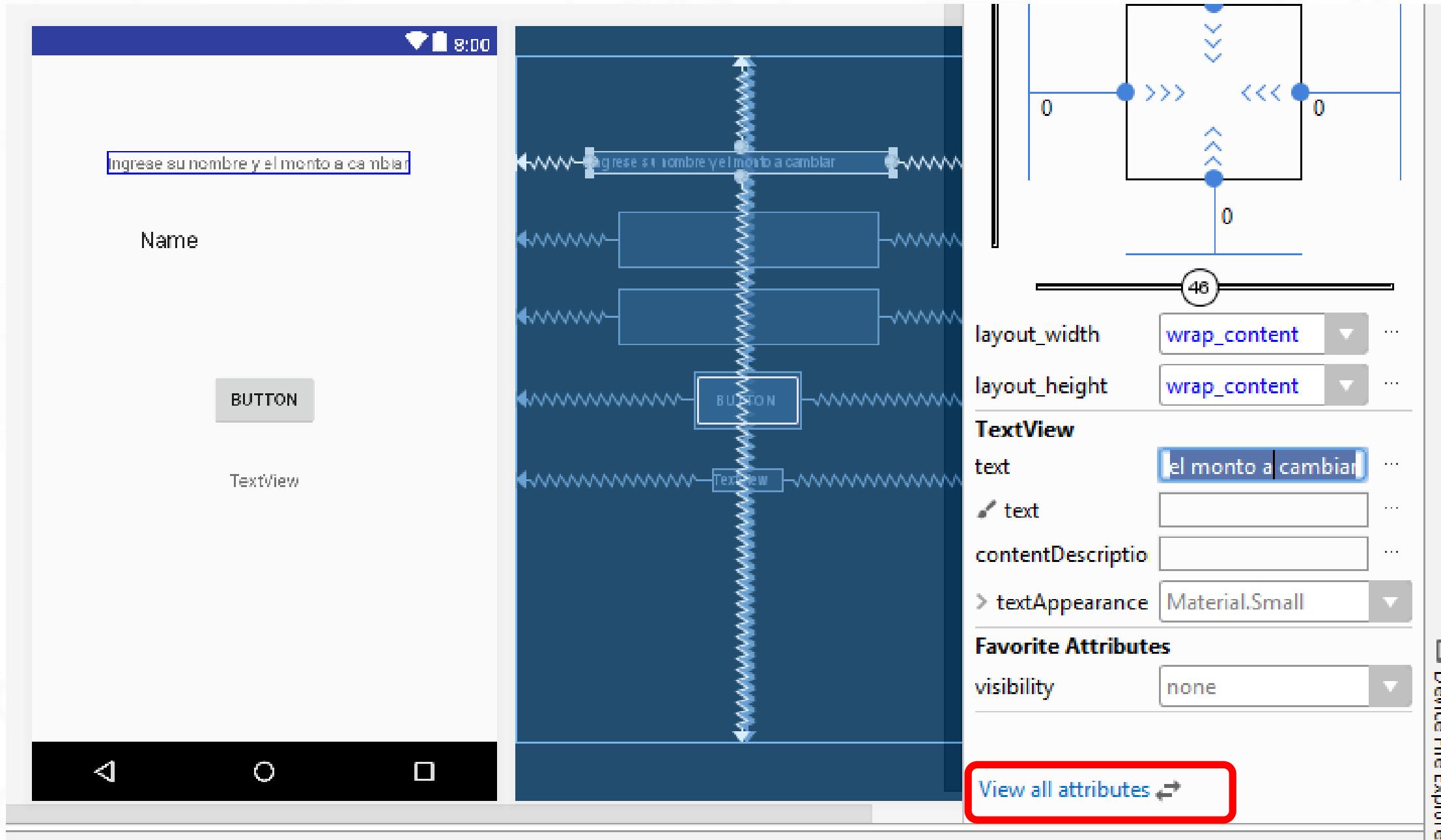
Manipulando Elementos por Código

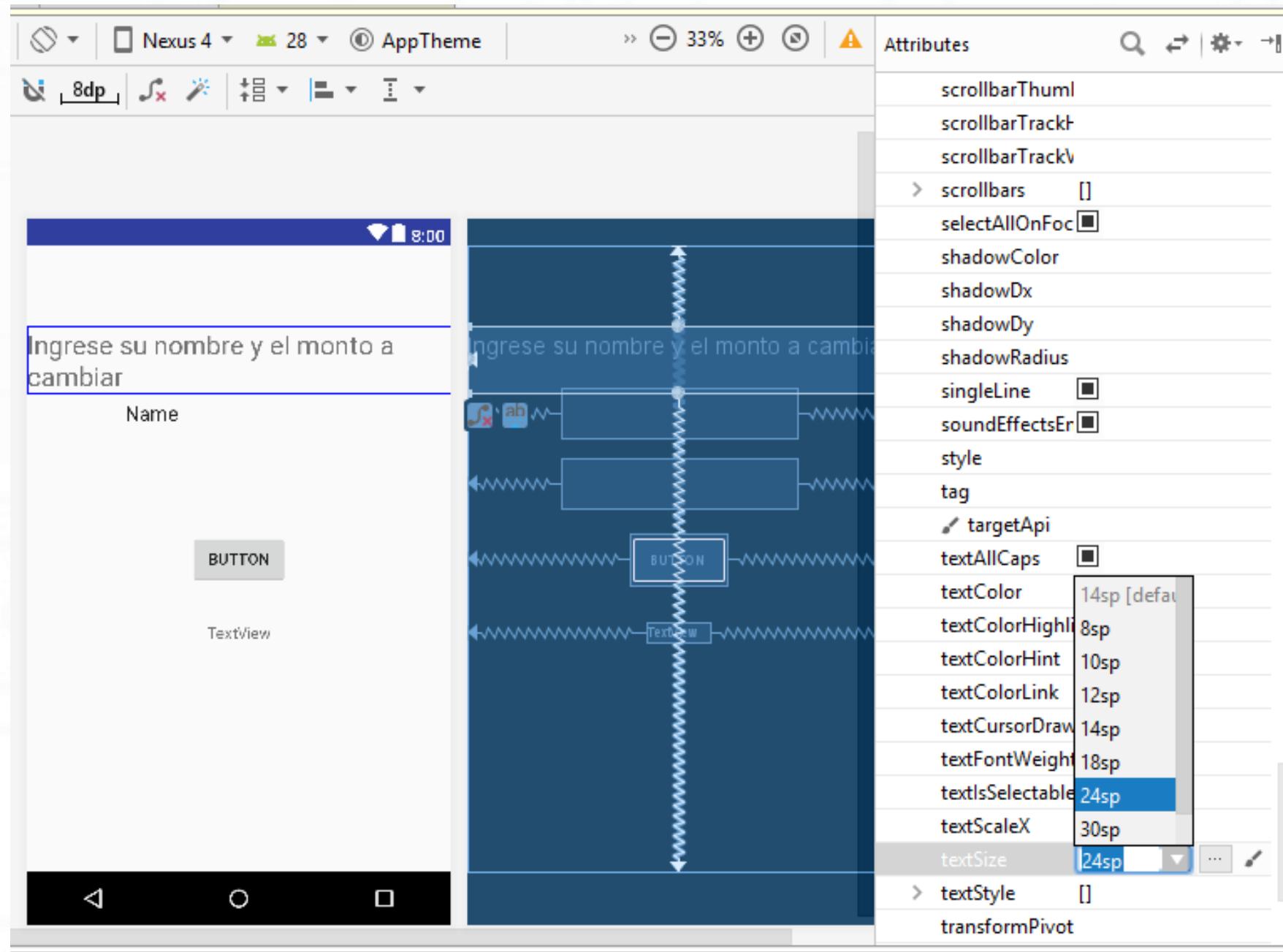
- Creamos la Actividad Manipulacion
- Generamos el Onclick respectivo en la botonera de la actividad principal (Actividad3)
- Ejecutamos para verificar el llamado
- Posteriormente usamos el modo Design para construir su layout según lo que se muestra a continuación.

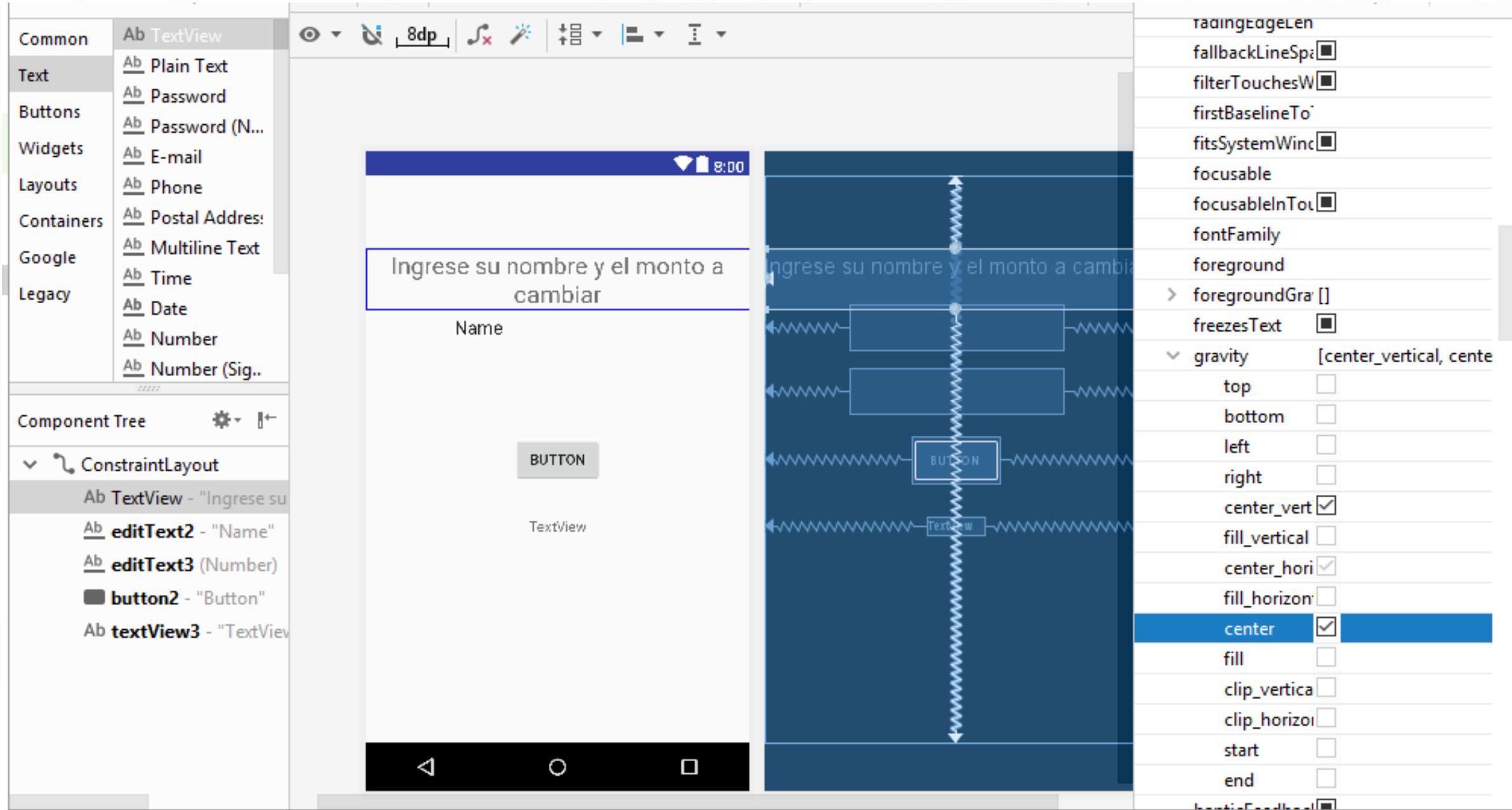


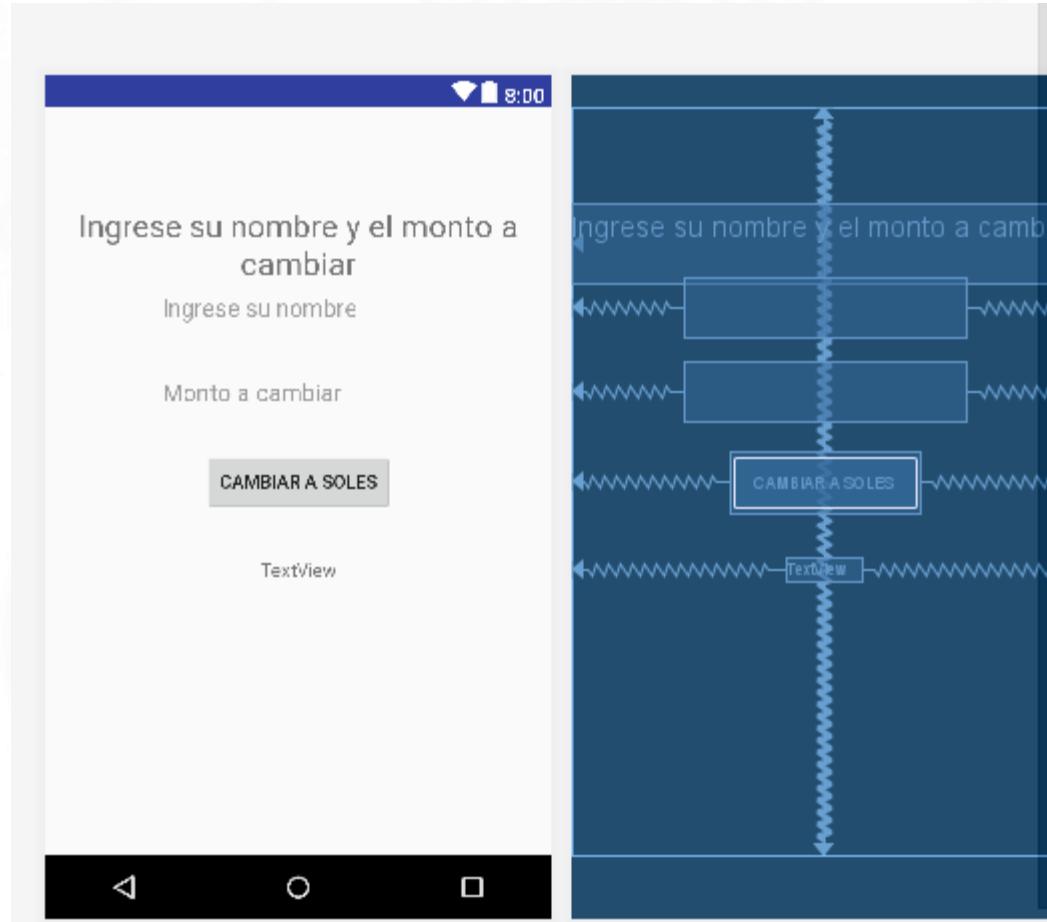












Modificamos los identificadores de sus vistas

```
<EditText  
    android:id="@+id/edtnombre"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="Ingrese su nombre"  
    android:inputType="textPersonName"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.502"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.247" />  
  
<EditText  
    android:id="@+id/edtmonto"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="Monto a cambiar"  
    android:inputType="number"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.502"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.37" />
```

```
<Button  
    android:id="@+id/btncambiar"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Cambiar a soles"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.5"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />  
  
<TextView  
    android:id="@+id/txtResultado"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="TextView"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.622" />
```

```
public class Manipulacion extends AppCompatActivity {
    EditText edtnombre;
    EditText edtmonto;
    TextView txtresultado;
    Button btnejecutar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_manipulacion);
        edtnombre=findViewById(R.id.edtnombre);
        edtmonto=findViewById(R.id.edtmonto);
        txtresultado=findViewById(R.id.txtResultado);
        btnejecutar=findViewById(R.id.btncambiar);
        btnejecutar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Double cambio;
                cambio=Double.parseDouble(edtmonto.getText().toString())*3.3;
                txtresultado.setText(edtnombre.getText() + " el monto de "+edtmonto.getText()+
                        " dolares equivale a " + cambio.toString() + "Soles");
            }
        });
    }
}
```

CODIGO java

```
public void ocultar(View view){  
    btnejecutar.setVisibility(View.GONE);  
}
```

CODIGO xml

```
<Button  
    android:id="@+id/button3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Ocultar"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.763"  
    android:onClick="ocultar"/>
```

Ingrese su nombre y el monto a cambiar

Ingrese su nombre

Monto a cambiar

CAMBIAR A SOLES

TextView

OCULTAR

Salir

Generar un onclick para salir que ejecute la siguiente instrucción

```
public void salir(View view){  
    finish();  
}
```

Capturando valor del spinner del Ejemplo5

Modificamos MiSpinner.class

```
public class MiSpinner extends AppCompatActivity {

    Spinner mispinner;
    String miitem;
    EditText edtDescripcion;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mi_spinner);
        mispinner=findViewById(R.id.spAnimals);
        edtDescripcion=findViewById(R.id.etDescription);
        mispinner
            .setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
                @Override
                public void onItemSelected(AdapterView<?> arg0,
                                           View arg1, int position, long arg3) {
                    // TODO Auto-generated method stub
                    miitem= (String) mispinner.getSelectedItem();
                    if(edtDescripcion.getText().length()>0)
                        edtDescripcion.setText(edtDescripcion.getText()+" , "+miitem);
                    else
                        edtDescripcion.setText(miitem);
                }
                @Override
                public void onNothingSelected(AdapterView<?> arg0) {
                    // TODO Auto-generated method stub
                }
            });
    }
}
```

Comunicación entre Actividades

Comunicación entre actividades

- Generamos dos actividades llamadas Comunicacion1 y Comunicacion2
- Añadimos un botón adicional para realizar el llamado a Comunicacion1 en la botonera del Ejemplo4 (Actividad3)
- Generamos un nuevo String para dicho boton (Comunicación entre actividades)
- No olvidar generar el Onclick respectivo
- Compile para verificar funcionamiento
- Luego reemplazar por el código siguiente

activity_comunicacion1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".Comunicacion1">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="Ingrese su nombre" />
    <EditText
        android:id="@+id/edtnombre"
        android:hint="Ingresa tu nombre"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <EditText
        android:id="@+id/edtedad"
        android:hint="Ingresa tu edad"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="number"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="Verificar"
        android:text="Verificar" />
    <TextView
        android:id="@+id/txtvalor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="" />
</LinearLayout>
```

Comunicacion1.class

```
public class Comunicacion1 extends AppCompatActivity {

    EditText edtnombre;
    EditText edtedad;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_comunicacion1);
        edtnombre=findViewById(R.id.edtnombre);
        edtedad=findViewById(R.id.edtedad);
    }
    public void Verificar(View view){
        Intent intent = new Intent(this, Comunicacion2.class);
        intent.putExtra("nombre", edtnombre.getText().toString());
        intent.putExtra("edad", Integer.parseInt(edtedad.getText().toString()));
        startActivity(intent);
    }
}
```

activity_comunicacion2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".Comunicacion2">
    <TextView
        android:id="@+id/txtmensaje"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="test"
        android:textSize="35sp"/>
</LinearLayout>
```

Comunicacion2.class

```
public class Comunicacion2 extends AppCompatActivity {
    TextView txtmensaje;
    String nombre;
    int edad;
    int evaluacion;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_comunicacion2);
        Bundle extras = getIntent().getExtras();
        nombre= extras.getString("nombre");
        edad = extras.getInt("edad");
        txtmensaje=findViewById(R.id.txtmensaje);
        if (edad > 25)
            evaluacion = 3000;
        else
            evaluacion = 2000;
        txtmensaje.setText("Buen día "+ nombre + " por su edad de " + edad +
            " podemos ofrecerle " + evaluacion + "soles");
    }
}
```

Enviando y recibiendo mensaje

Modificamos activity_comunicacion2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".Comunicacion2">
    <TextView
        android:id="@+id/txtmensaje"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="test"
        android:textSize="35sp"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_horizontal">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Aceptar"
            android:onClick="aceptar"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Rechazar"
            android:onClick="rechazar"/>
    </LinearLayout>
</LinearLayout>
```

Modificamos comunicacion1.class

```
public class Comunicacion1 extends AppCompatActivity {

    EditText edtnombre;
    EditText edtedad;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_comunicacion1);
        edtnombre=findViewById(R.id.edtnombre);
        edtedad=findViewById(R.id.edtedad);
    }
    public void Verificar(View view){
        Intent intent = new Intent(this, Comunicacion2.class);
        intent.putExtra("nombre", edtnombre.getText().toString());
        intent.putExtra("edad", Integer.parseInt(edtedad.getText().toString()));
        //startActivity(intent);
        startActivityForResult(intent, 1234);
    }
    @Override protected void onActivityResult (int requestCode,
                                              int resultCode, Intent data){
        if (requestCode==1234 && resultCode==RESULT_OK) {
            String res = data.getExtras().getString("resultado");
            TextView miresultado=findViewById(R.id.txtvalor);
            if(res.equals("aceptado"))
                miresultado.setText("Esperamos que trabajar con nosotros sea de su agrado");
            else
                miresultado.setText("Esperemos que cambie de opinion para trabajar con nosotros");
        }
    }
}
```

Modificamos comunicacion2.class

```
public class Comunicacion2 extends AppCompatActivity {
    TextView txtmensaje;
    String nombre;
    int edad;
    int evaluacion;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_comunicacion2);
        Bundle extras = getIntent().getExtras();
        nombre= extras.getString("nombre");
        edad = extras.getInt("edad");
        txtmensaje=findViewById(R.id.txtmensaje);
        if (edad > 25)
            evaluacion = 3000;
        else
            evaluacion = 2000;
        txtmensaje.setText("Buen día " + nombre + " por su edad de " + edad +
                " podemos ofrecerle " + evaluacion + "soles");
    }
    public void aceptar(View view){
        Intent intent = new Intent();
        intent.putExtra("resultado","aceptado");
        setResult(RESULT_OK, intent);
        finish();
    }
    public void rechazar(View view){
        Intent intent = new Intent();
        intent.putExtra("resultado","rechazado");
        setResult(RESULT_OK, intent);
        finish();
    }
}
```

ScrollView

MSc. Ing. Wilson César Callisaya Choquecota

Ponemos el contenido de activity_actividad3.xml dentro de un ScrollView

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">
```

GITHUB ANDROID Y APIDEMOS

MSc. Ing. Wilson César Callisaya Choquecota

Comandos Básicos del GIT por consola

git clone <https://github.com/WCNosliwsys/apidemos.git>

clonar un proyecto

git add –A

agregar archivos para subir

git commit –m “comentario del fragmento de codigo”

añadir archivos al historial

git push origin master

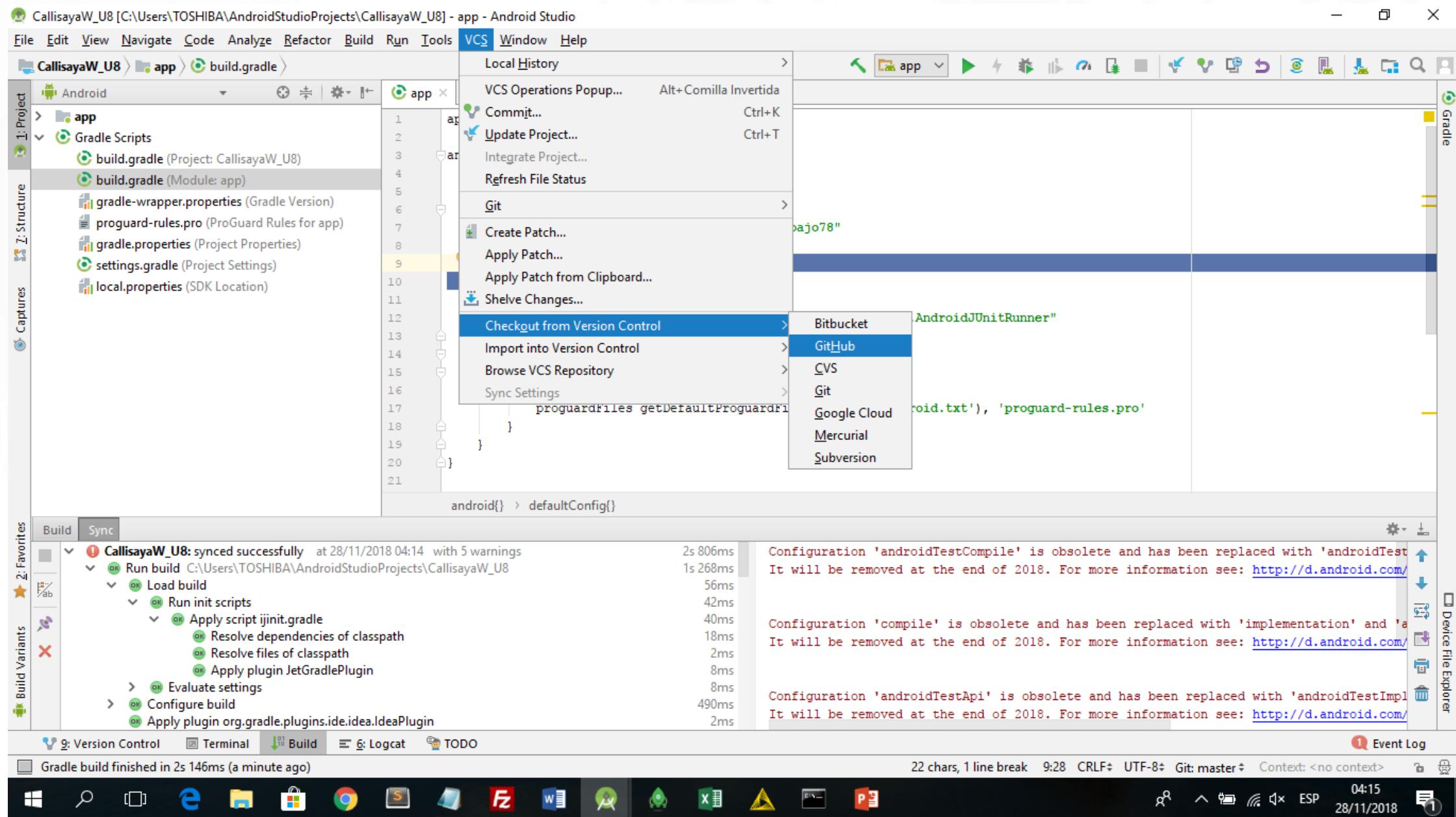
subir archivos al repositorio

git status –s

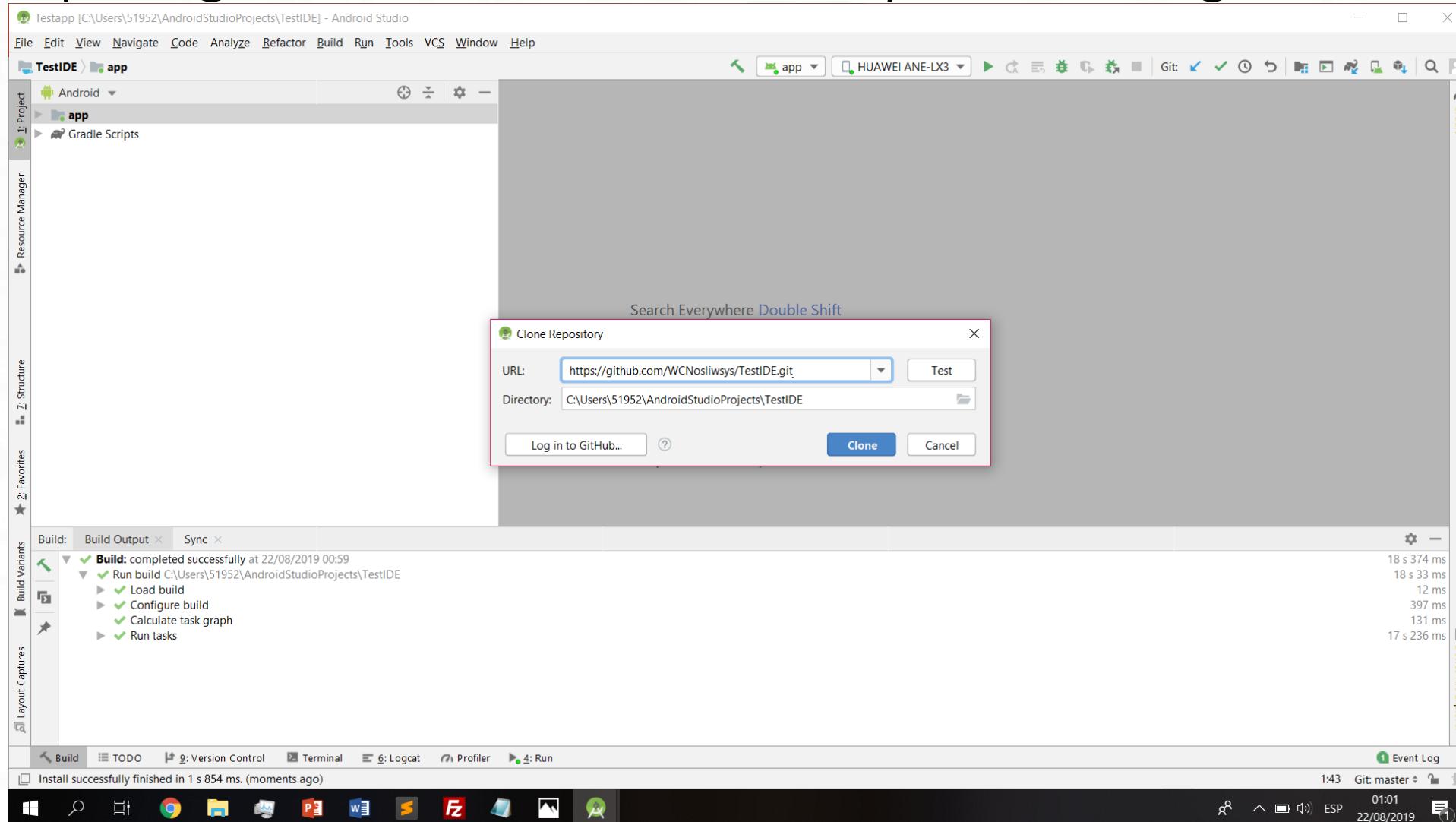
para ver el estado de los archivos

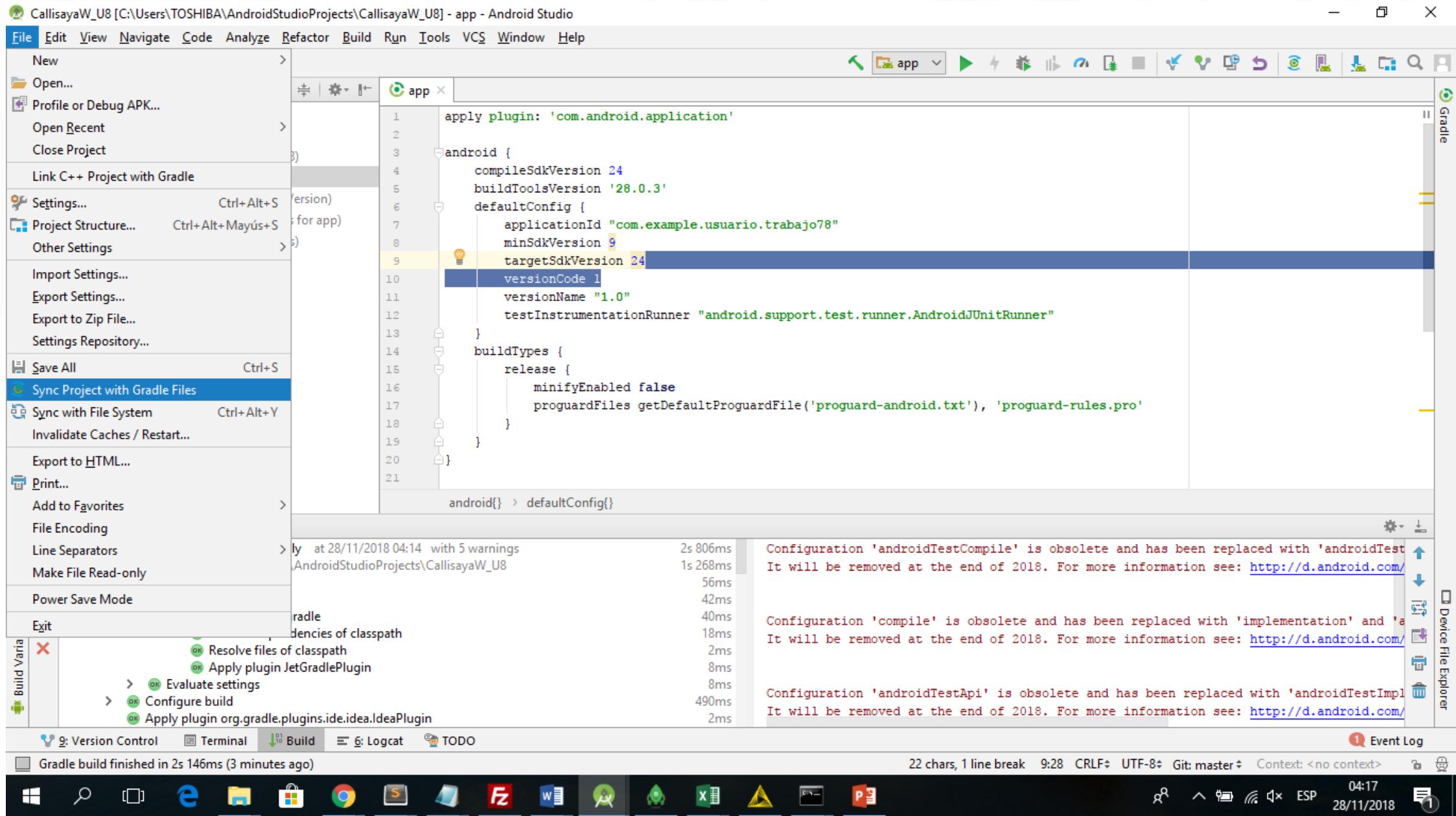
<https://github.com/>

The screenshot shows the GitHub homepage with a dark background featuring a faint circuit board pattern. On the left, the GitHub logo and navigation links like 'Why GitHub?', 'Business', 'Explore', 'Marketplace', and 'Pricing' are visible. A search bar and 'Sign in'/'Sign up' buttons are at the top right. The main headline 'Built for developers' is prominently displayed in large white text. Below it, a paragraph explains GitHub's purpose: 'GitHub is a development platform inspired by the way you work. From [open source](#) to [business](#), you can host and review code, manage projects, and build software alongside 31 million developers.' To the right of this text is a white sign-up form with fields for 'Username' (placeholder 'Pick a username'), 'Email' (placeholder 'you@example.com'), and 'Password' (placeholder 'Create a password'). A note below the password field states: 'Make sure it's more than 15 characters, or at least 8 characters, including a number, and a lowercase letter.' A large green 'Sign up for GitHub' button is at the bottom of the form. At the very bottom of the screen, a Windows taskbar shows various open applications and the user's name 'Wilson César Callisaya Choquecota'.



<https://github.com/WCNosliwsys/TestIDE.git>

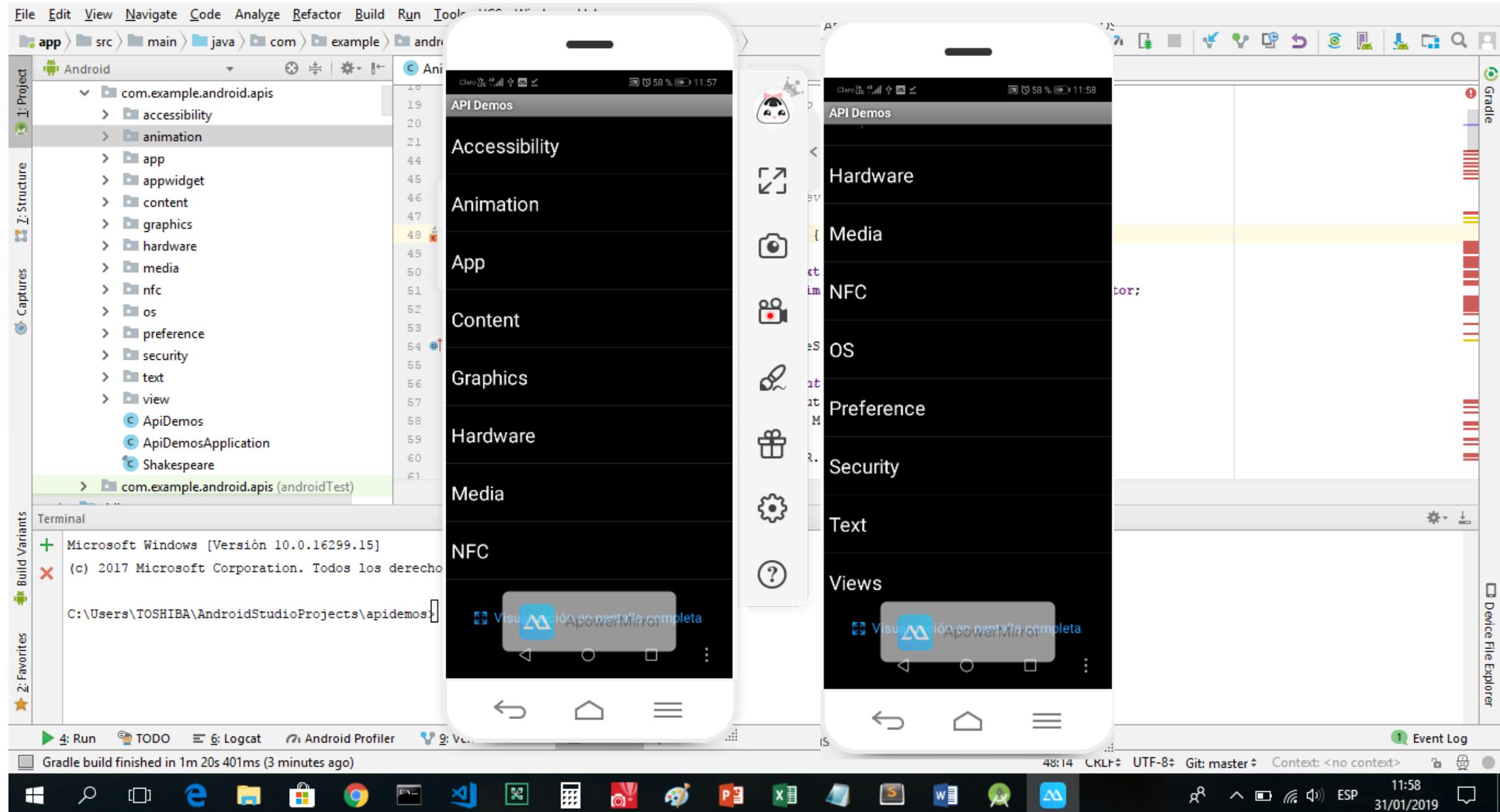




<https://github.com/WCNosliwsys>

<https://github.com/WCNosliwsys/TestIDE.git>

<https://github.com/WCNosliwsys/apidemos.git>



<http://www.androidcurso.com/index.php/recursos/proyectos/87-proyectos-android-fundamentos/pfund2017>

MÁSTER ▾ MOOC ▾ RECURSOS ▾ ASIGNATURAS ▾ LIBROS ▾ CONTACTO

INICIO | PROYECTOS ANDROID FUNDAMENTOS 8^a ED. 2017

Proyectos Android Fundamentos 8^a Ed. 2017

ArtesApp



"FINALISTA AL MEJOR PROYECTO"

Descripción: Aplicación para comprar o mandar a realizar, productos hechos con caña flecha. Diseña tu propio articulo y te lo creamos.

Alumno: Aguas Barraza, Dairo

Memoria: Artesapp.pdf

Aplicación: Artesapp.apk

Ejercicio Propuesto

- Generar la visualización y funcionalidad de una calculadora
- Generar formulario de datos de un alumno (considerar Nombre, apellido, grado de instrucción, sexo, fecha de nacimiento), usar spinner para los datos sexo y grado de instrucción.
- Botón que envíe del formulario de datos del alumno a otro donde diga bienvenido con su nombre y apellido juntos.
- Paso a paso en su propio github

Detalle de Presentación de Trabajo Encargado

El Trabajo encargado de Alternativas Multiplataforma Xamarin, Ionic, React Native, Native Script o Flutter ya asignado en la primera semana del cursos será presentado en un Video en Youtube se recomienda usar una herramienta como <https://www.apowersoft.es/grabador-de-pantalla-gratis>

Se debe presentar el link del video en youtube al correo
nosliwsys@gmail.com con asunto UPT SuNombre SolucionesMoviles
TrabajoEncargadoU2

Duración del video: menos de 10 minutos

A tratar en el video Indicar sobre lo necesario para su instalación de cada alternativa asignada, ventajas de las alternativas multiplataformas asignadas, como realizo los 2 ejercicios asignados, en el video debe mostrar la compilación de ambos ejercicios en un celular (para duplicar la pantalla del celular en el computador puede usar apowermirror o el de su preferencia), no olvidar el video **debe estar la voz del estudiante en un video sin editar.**



UNIVERSIDAD PRIVADA DE TACNA



Gracias por su atención.

MSc. Ing. Wilson César Callisaya Choquecota

nosliwsys@gmail.com

952000243 – Tacna Perú