



Universidad Nacional  
Jorge Basadre Grohmann



# PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES: ANDROID

## Parte I

**MSc. ING. WILSON CÉSAR CALLISAYA CHOQUECOTA**

# TEMARIO

- Ejecutar Hola mundo en Android
- Estructura de un proyecto en Android
- Ejemplo de Vistas básicas
- Fragment TabHost
- Ciclo de Vida de Android
- Manipulacion de Elementos por Código
- Recycler View
- Añadiendo botón a recycler View
- 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.

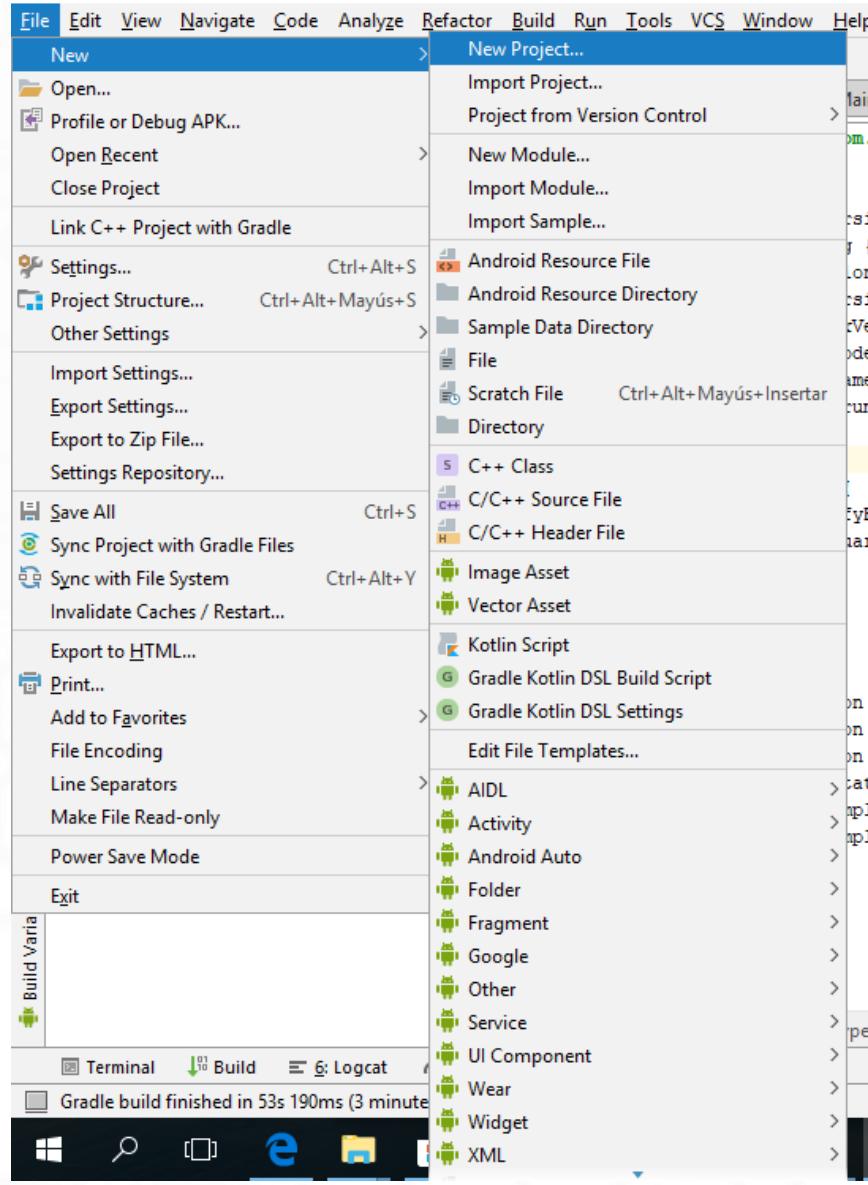
DESCARGA NO DISPONIBLE

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'

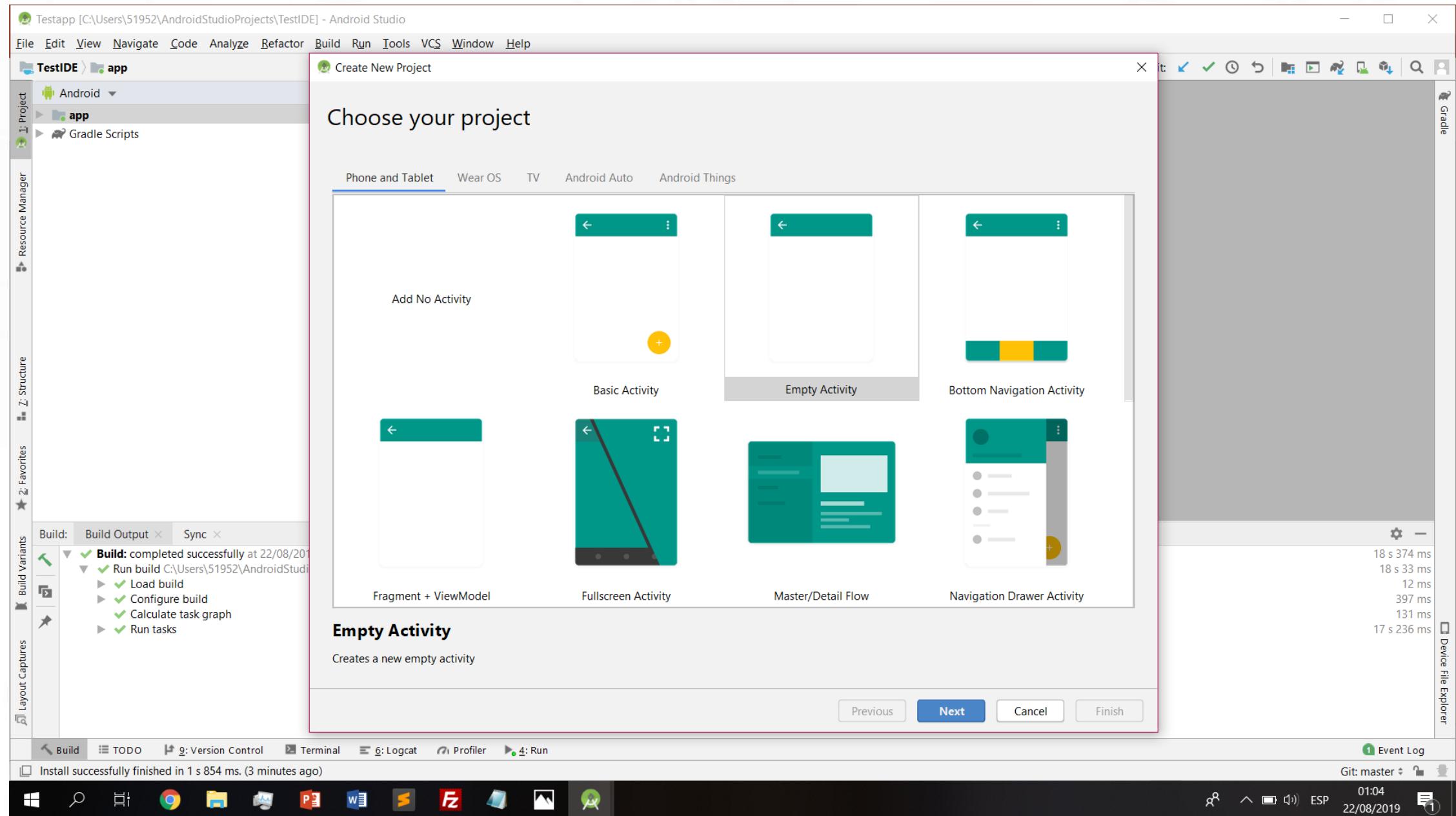
version 28
j {
ionId "com.example.toshiba.miprimeraapp"
ersion 15
Version 28
ode 1
ame "1.0"
umentationRunner "android.support.test.runner.AndroidJUnitRunner"

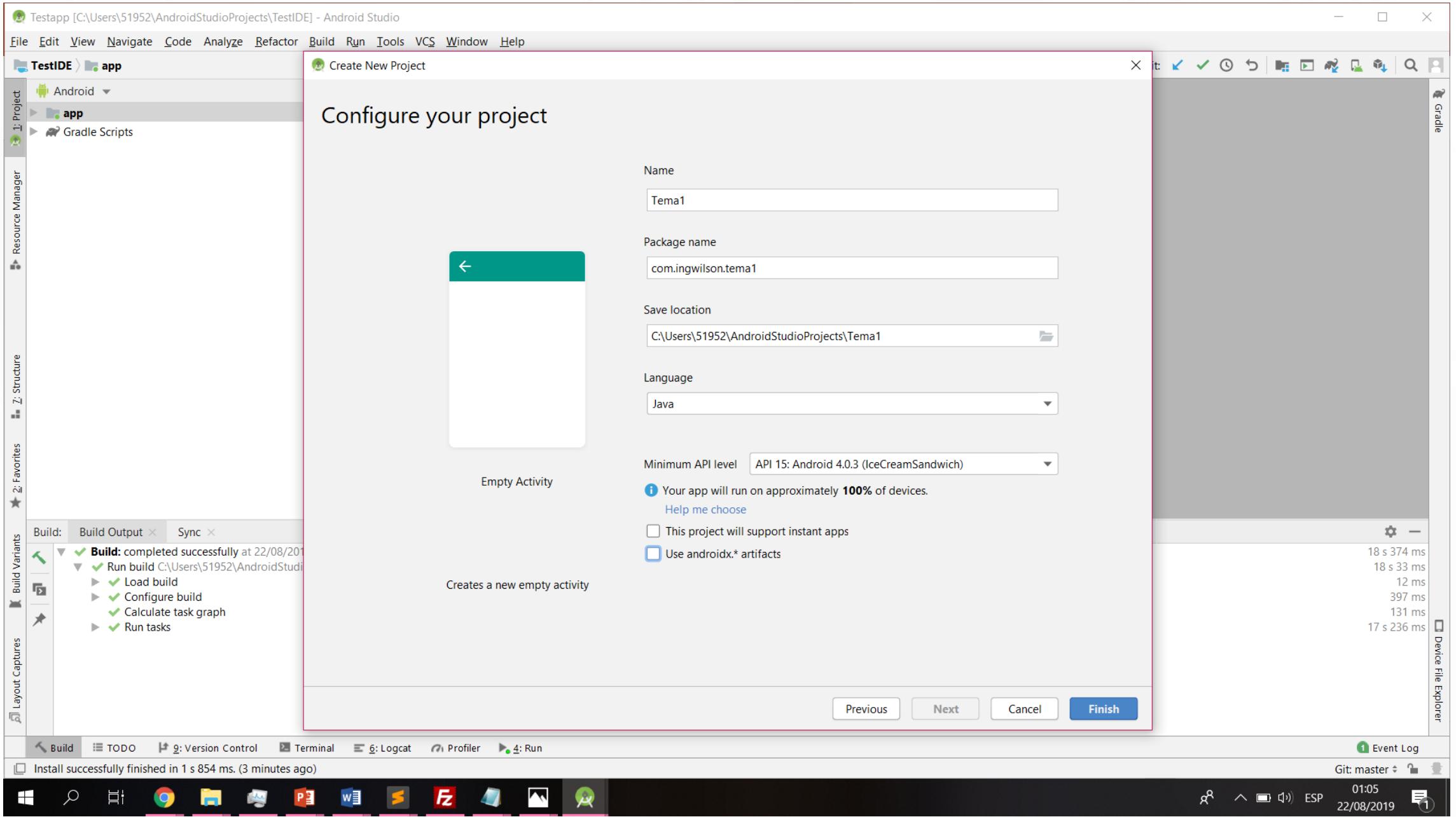
ayEnabled false
ardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

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
```

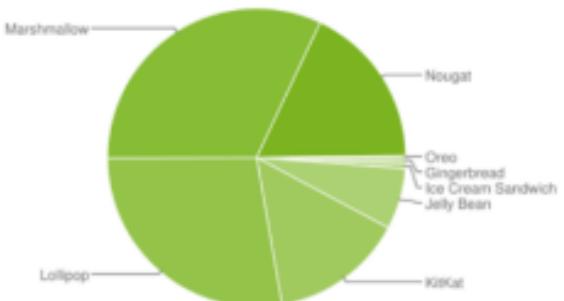






Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.6%
4.1.x	Jelly Bean	16	2.3%
4.2.x		17	3.3%
4.3		18	1.0%
4.4	KitKat	19	14.5%
5.0	Lollipop	21	6.7%
5.1		22	21.0%
6.0	Marshmallow	23	32.0%
7.0	Nougat	24	15.8%
7.1		25	2.0%
8.0	Oreo	26	0.2%

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)

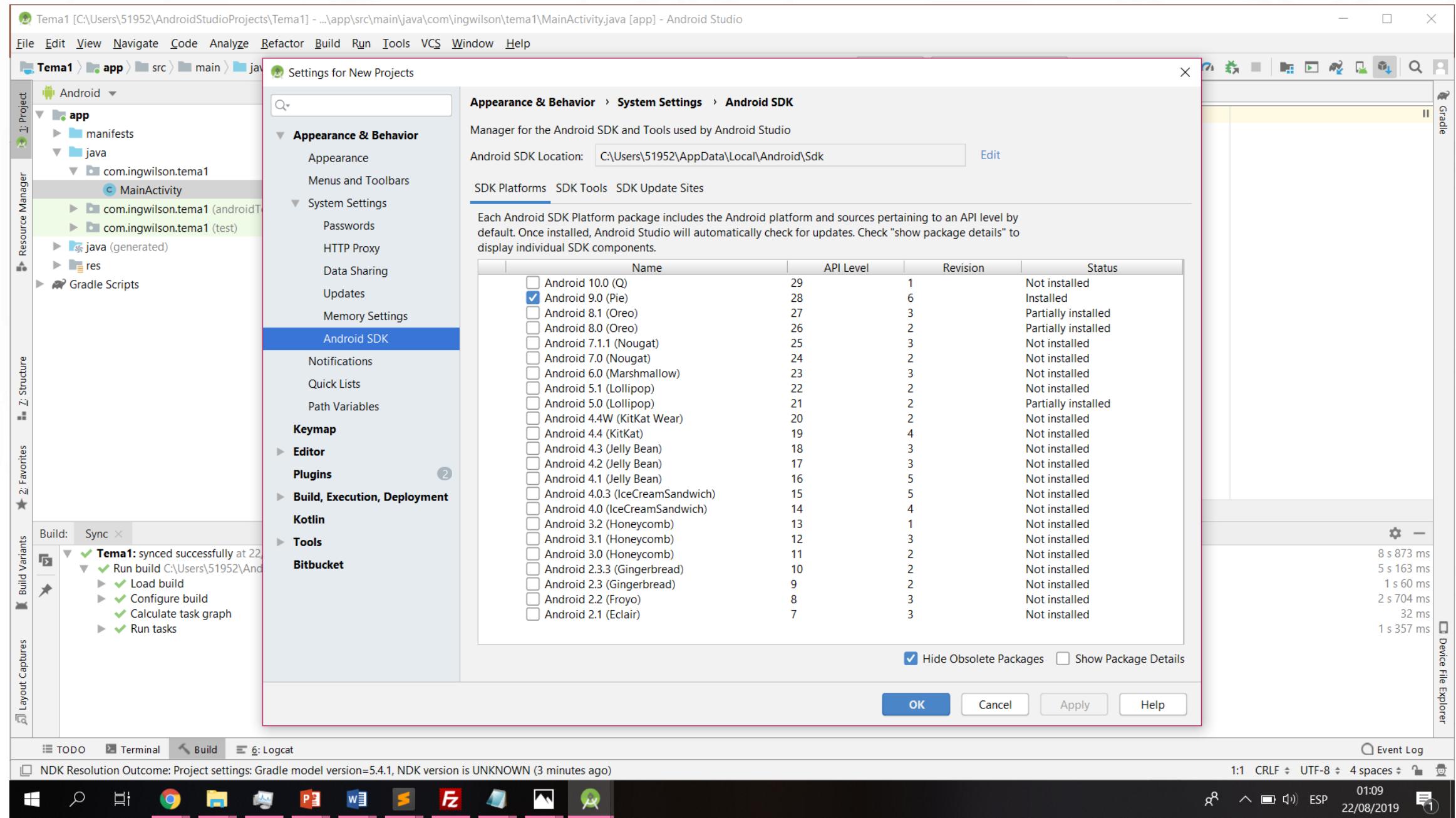
```
1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion 28
5     defaultConfig {
6         applicationId "com.example.toshiba.miprimeraapp"
7         minSdkVersion 15
8         targetSdkVersion 28
9         versionCode 1
10        versionName "1.0"
11        testInstrumentationRunner "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 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

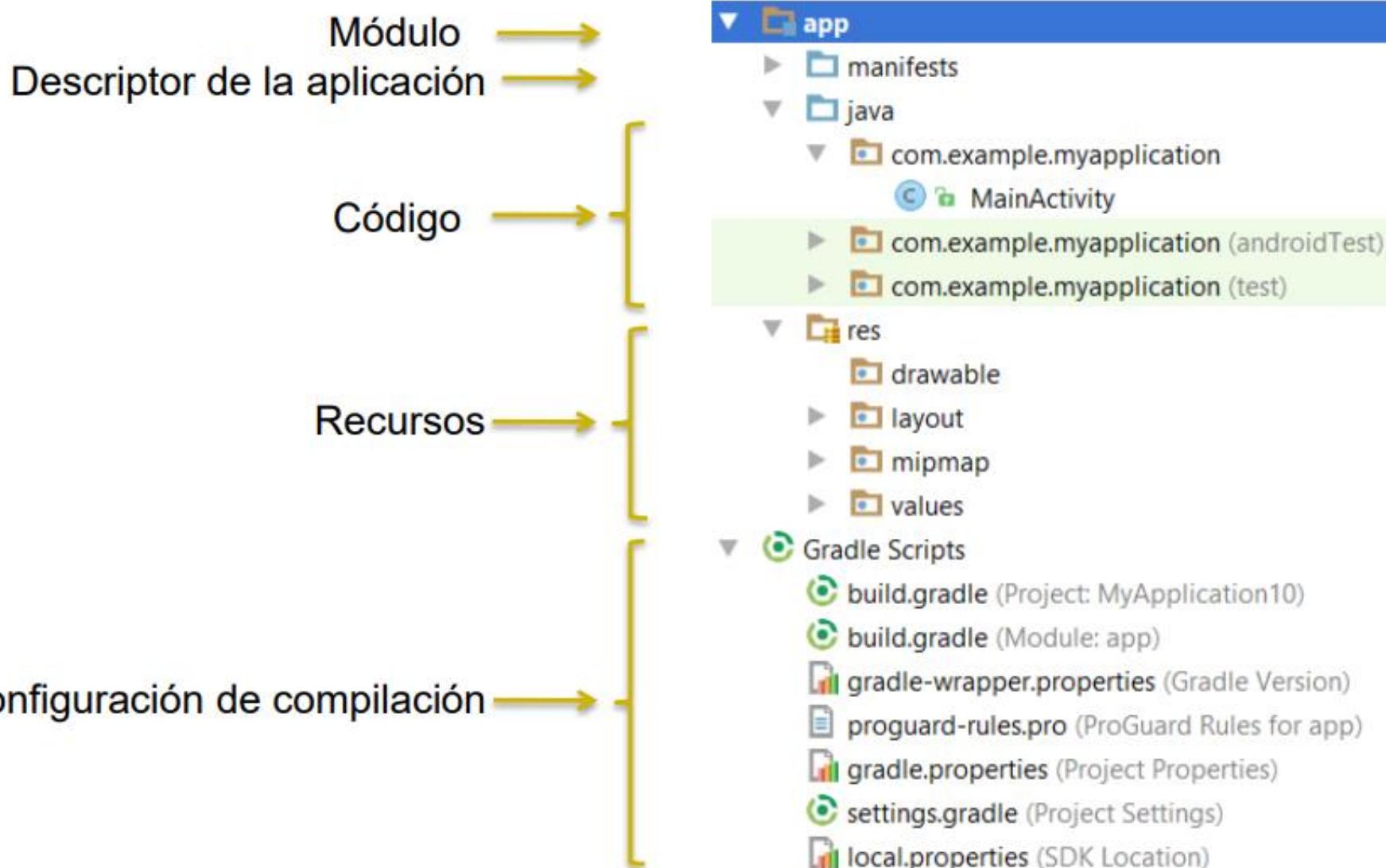
ASKS v1.3\_161228

#### 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

activity\_main.xml MainActivity.java app

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns: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" />

</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

1.1.2: manifests

1.1.3: AndroidManifest.xml

1.1.4: java

1.1.5: com.example.toshiba.miprimeraapp

1.1.6: MainActivity

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

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

1.1.9: res

1.1.10: drawable

1.1.11: layout

1.1.12: activity\_main.xml

1.1.13: mipmap

1.1.14: values

1.1.15: Gradle Scripts

1.1.16: build.gradle (Project: miprimeraapp)

1.1.17: build.gradle (Module: app)

1.1.18: gradle-wrapper.properties (Gradle Version)

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

1.1.20: gradle.properties (Project Properties)

1.1.21: settings.gradle (Project Settings)

1.1.22: local.properties (SDK Location)

1.1.23: Captures

1.1.24: Favorites

1.1.25: Build Variants

1.1.26: Device File Explorer

1.1.27: Gradle

1.1.28: 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"
```

The screenshot shows the Android Studio interface with the project 'miprimeraapp' open. The left sidebar displays the project structure, including the 'app' module with its manifest, Java files (MainActivity), and resource folders (res). The 'Gradle Scripts' section lists build.gradle files for the project and module, along with other configuration files like gradle-wrapper.properties and proguard-rules.pro.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.toshiba.miprimeraapp"
        minSdkVersion 15
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
}
```

# 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. The XML code is as follows:

```
<?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">

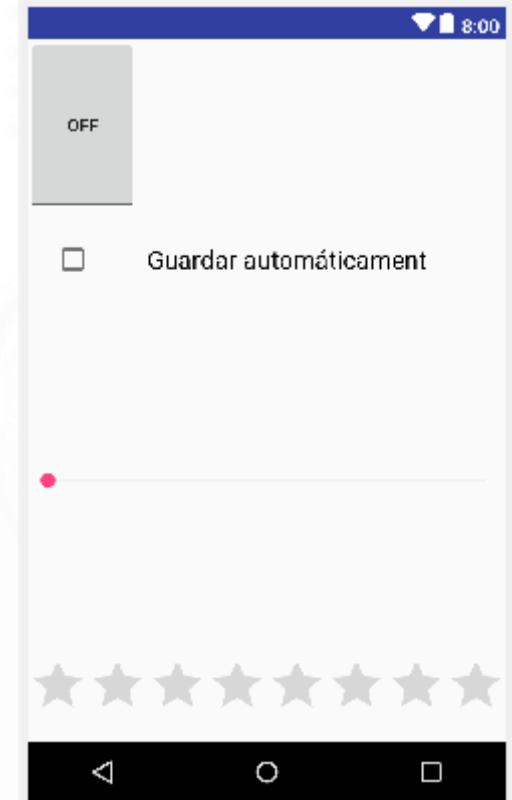
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
   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

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

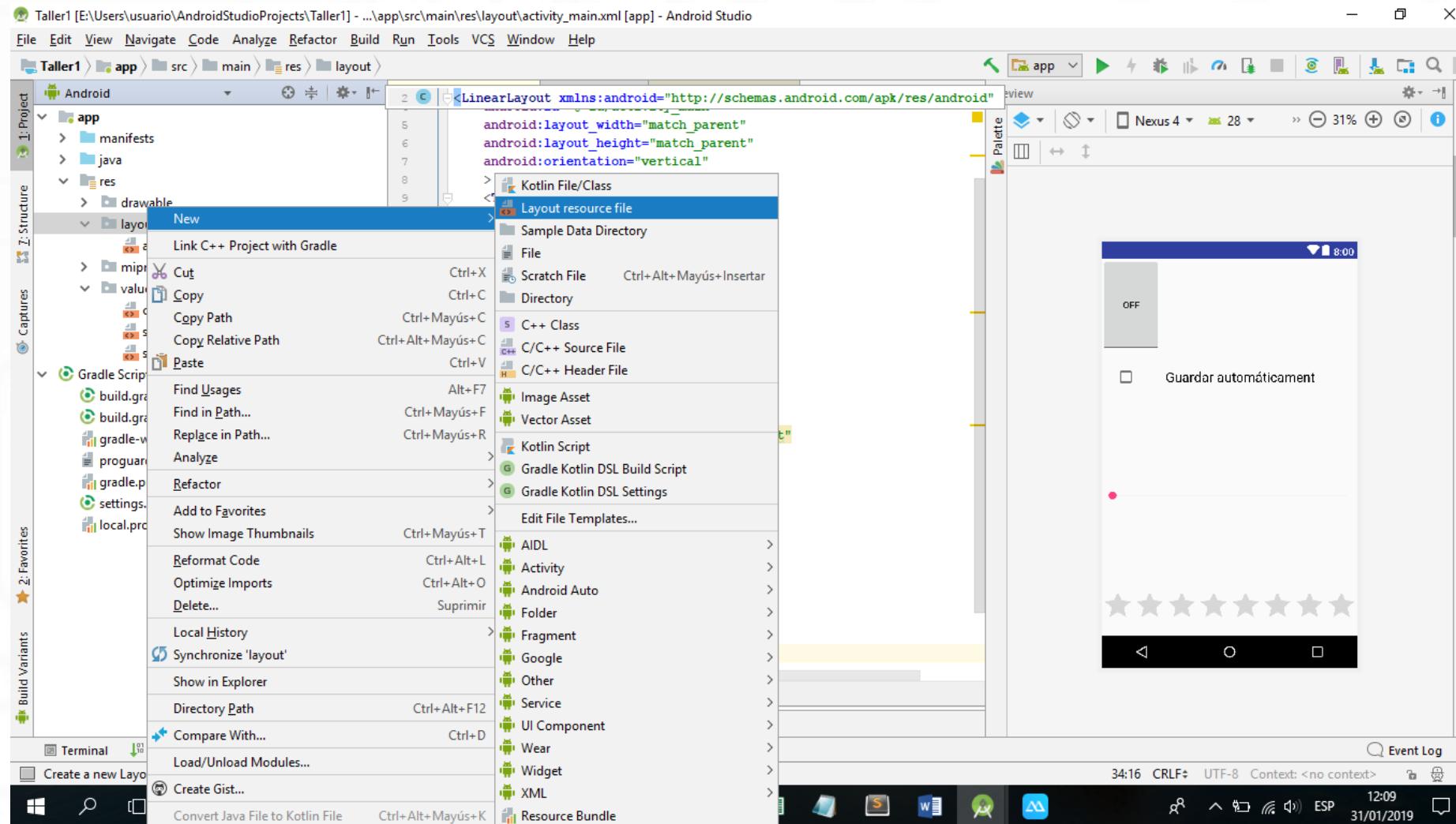
The code editor has tabs for 'Design' and 'Text'. The bottom status bar shows the terminal output: 'Gradle build finished in 18s 896ms (2 minutes ago)'. The system tray at the bottom right shows the date and time as '31/01/2019'.

# Ejemplo 1 – Reemplazar el activity\_main.xml

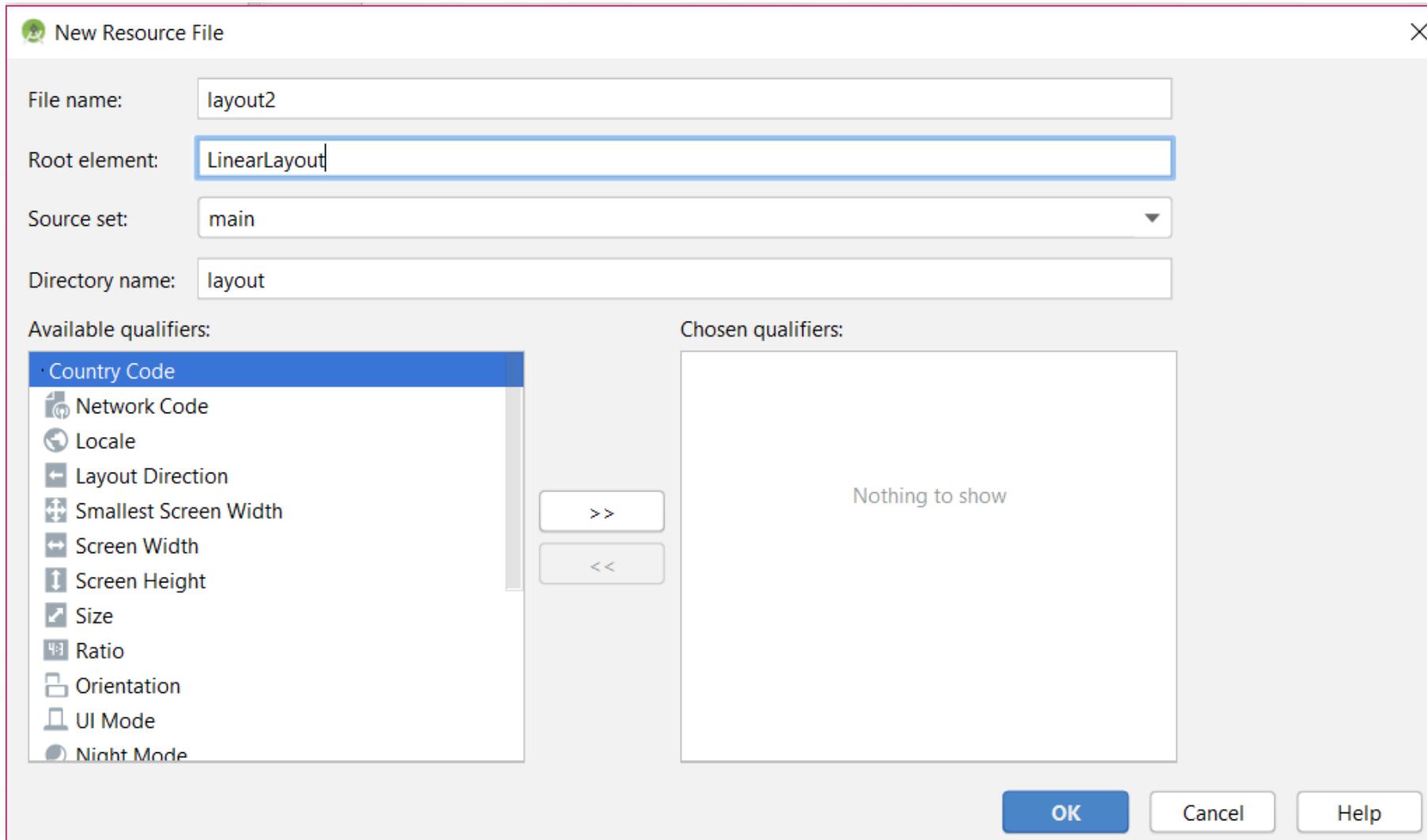
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
>
    <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 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>

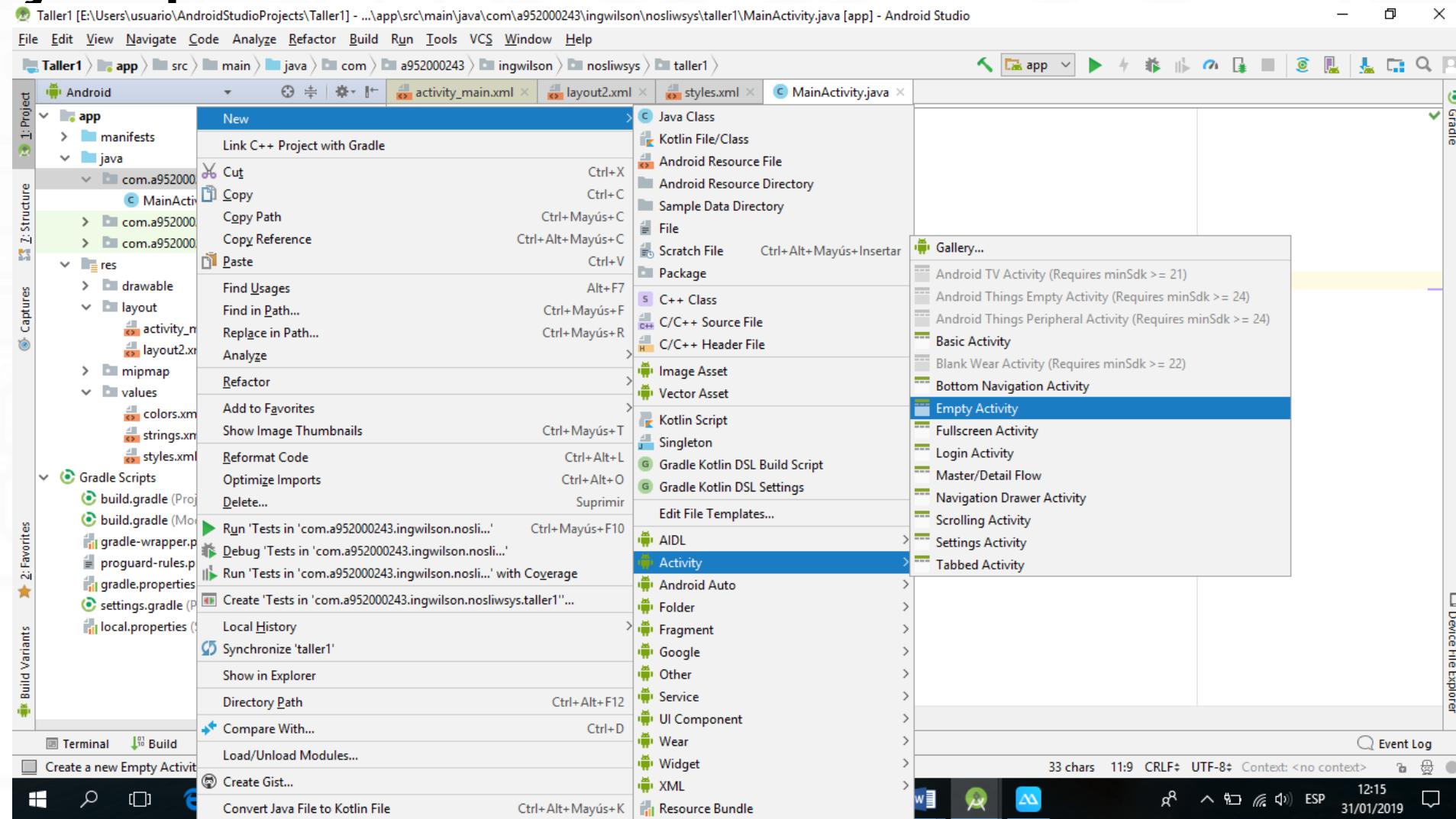
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Un botón" />

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

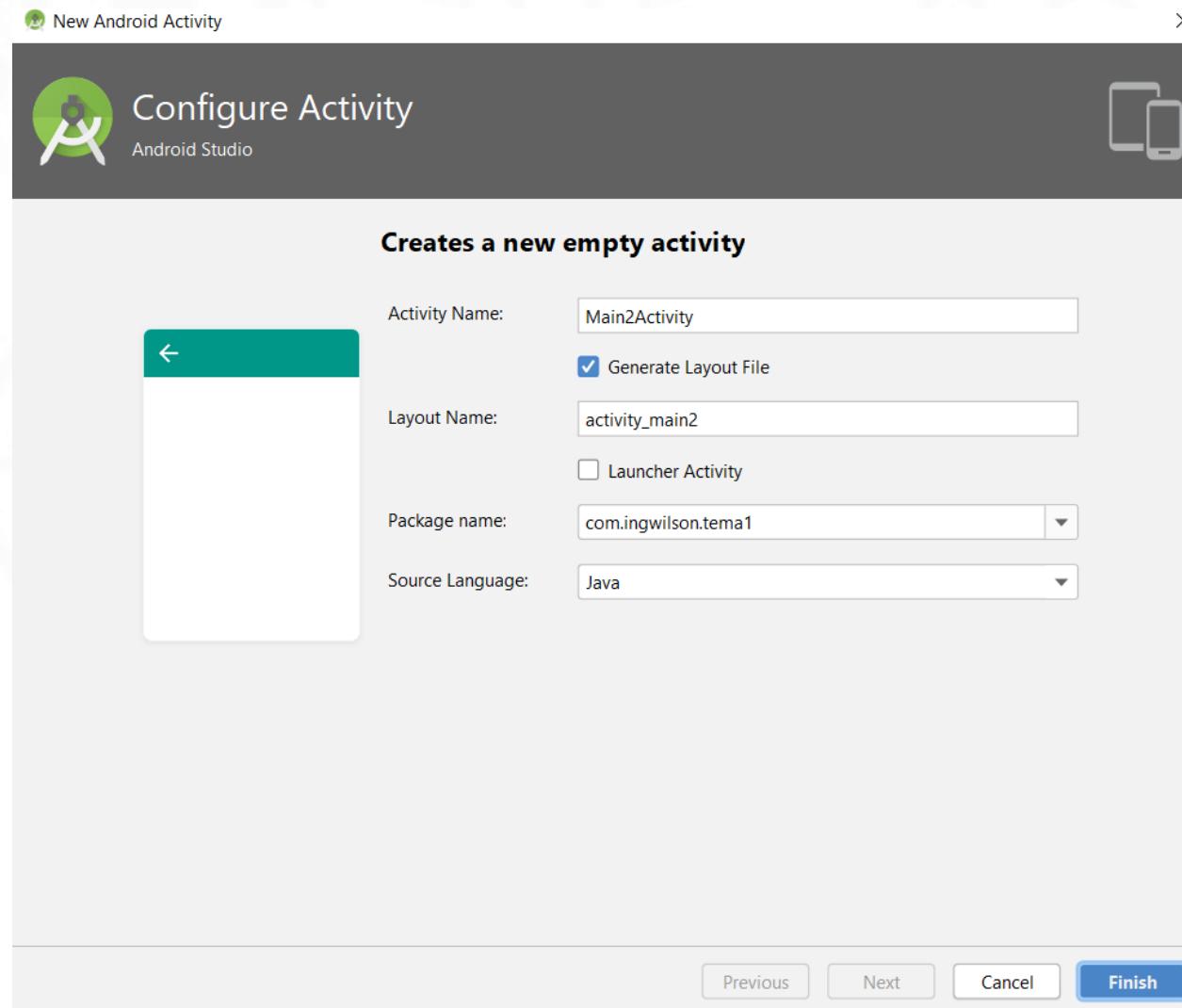
# Ejemplo 2 – Seleccionar layout2 en el MainActivity.class

```
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"  
        />  
    <Button android:id="@+id/button02"  
        android:layout_height="wrap_content"  
        android:layout_width="match_parent"  
        android:text="@string/opcion2"/>  
    <Button android:id="@+id/button04"  
        android:layout_height="wrap_content"  
        android:layout_width="match_parent"  
        android:text="@string/opcion3"/>  
    <Button android:id="@+id/button03"  
        android:layout_height="wrap_content"  
        android:layout_width="match_parent"  
        android:text="@string/opcion4"/>  
    <Button android:id="@+id/button05"  
        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">Fragment TabHost</string>
    <string name="opcion3">Manipulando elementos por codigo</string>
    <string name="opcion4">Recycler View</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

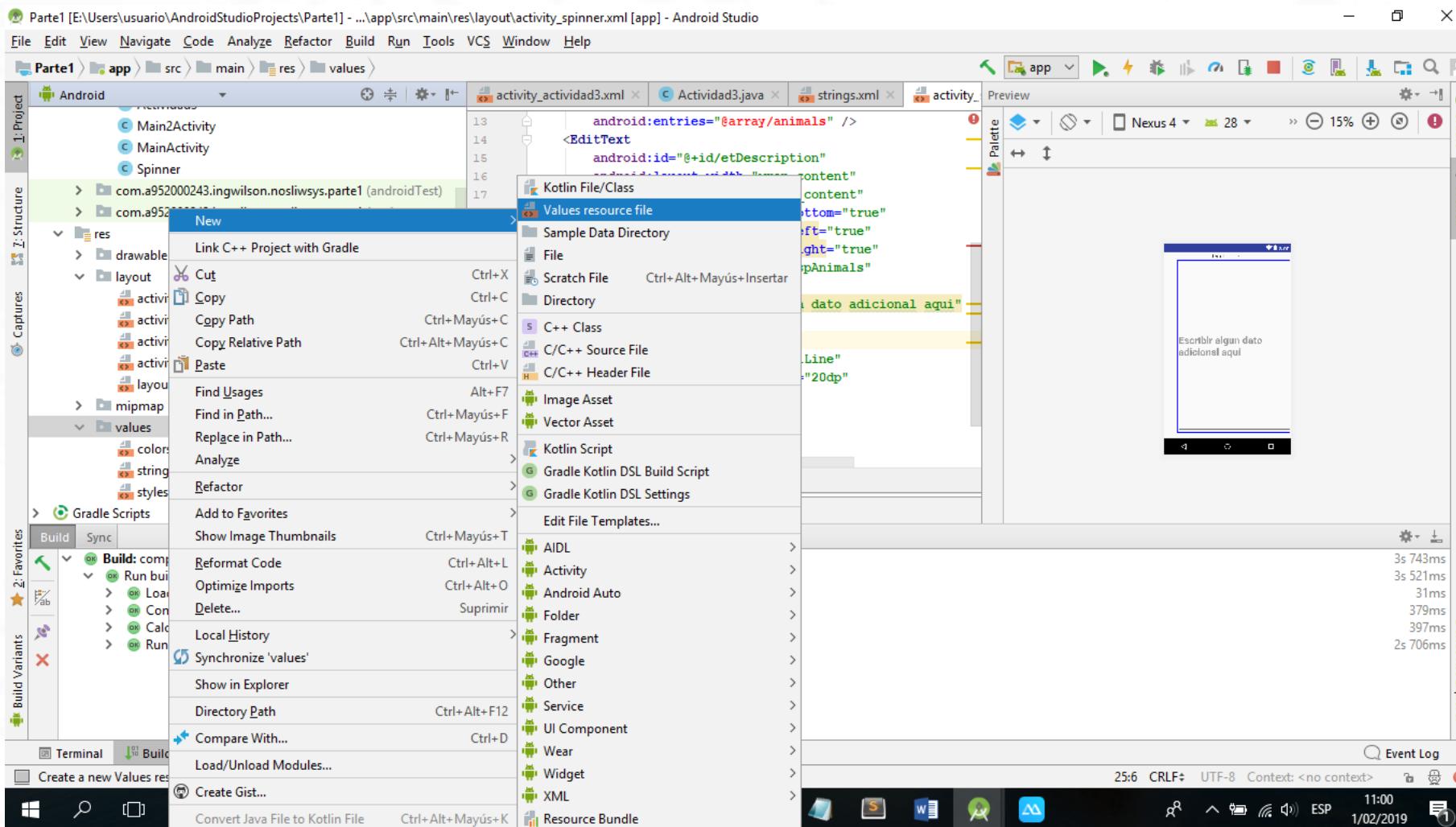
# Ejemplo 5 – activity\_mi\_spinner.xml

```
<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" />

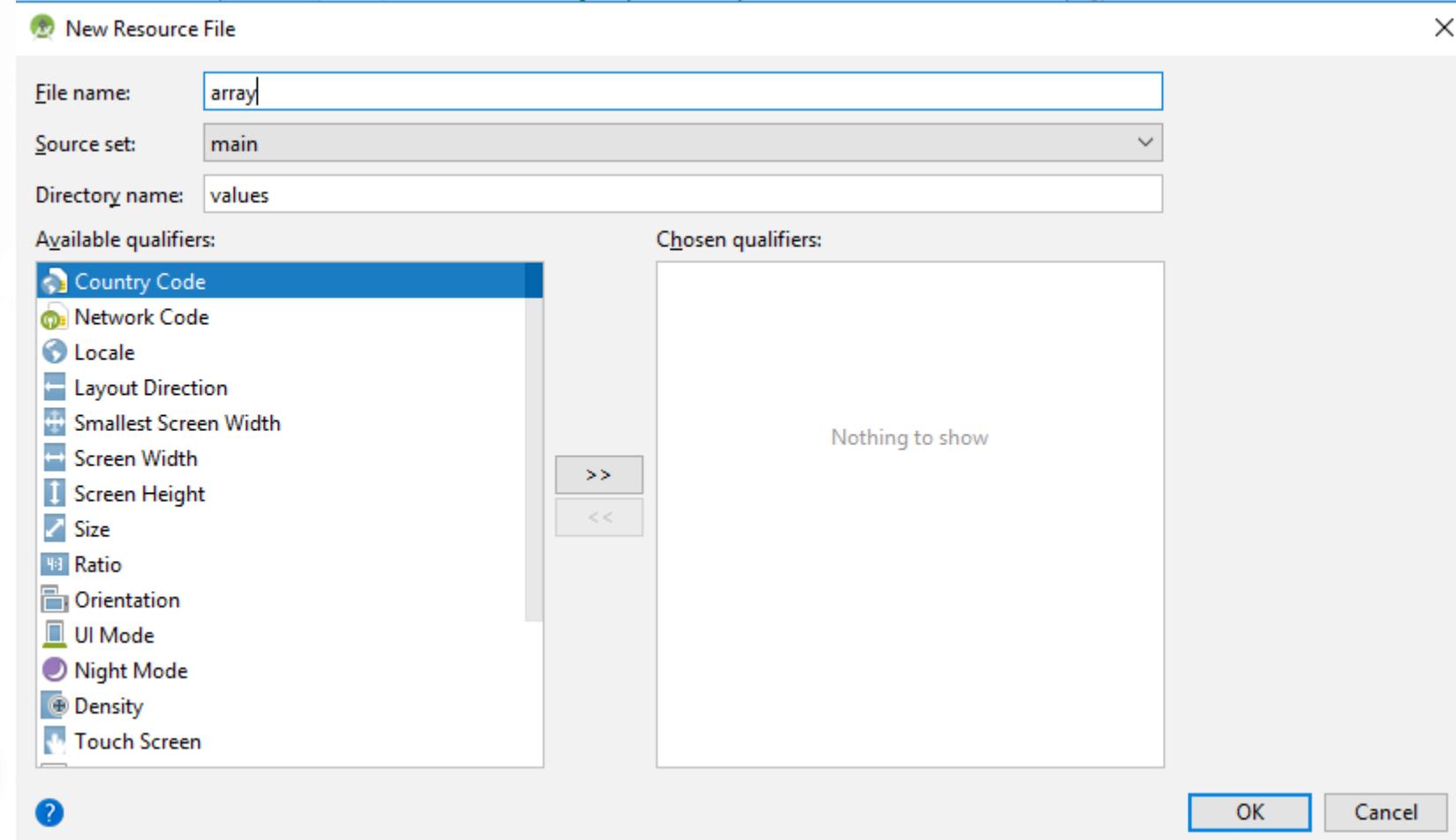
    <EditText
        android:id="@+id/etDescription"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/spAnimals"
        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>
```

# Fragment Tabhost

MSc. Ing. Wilson César Callisaya Choquecota

# FragmentTabhost

- Generamos una actividad llamada MiFragmentTabhost
- Añadir el Onclick respectivo

# activity\_mi\_fragment\_tab\_host.xml

## No copiar

```
<android.support.v4.app.FragmentTabHost
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tabhost"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >
        <TabWidget
            android:id="@+id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="0"
            android:orientation="horizontal" />
        <FrameLayout
            android:id="@+id/tabcontent"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1" />
    </LinearLayout>
</android.support.v4.app.FragmentTabHost>
```

# activity\_mi\_fragment\_tab\_host.xml

<https://developer.android.com/jetpack/androidx/migrate>

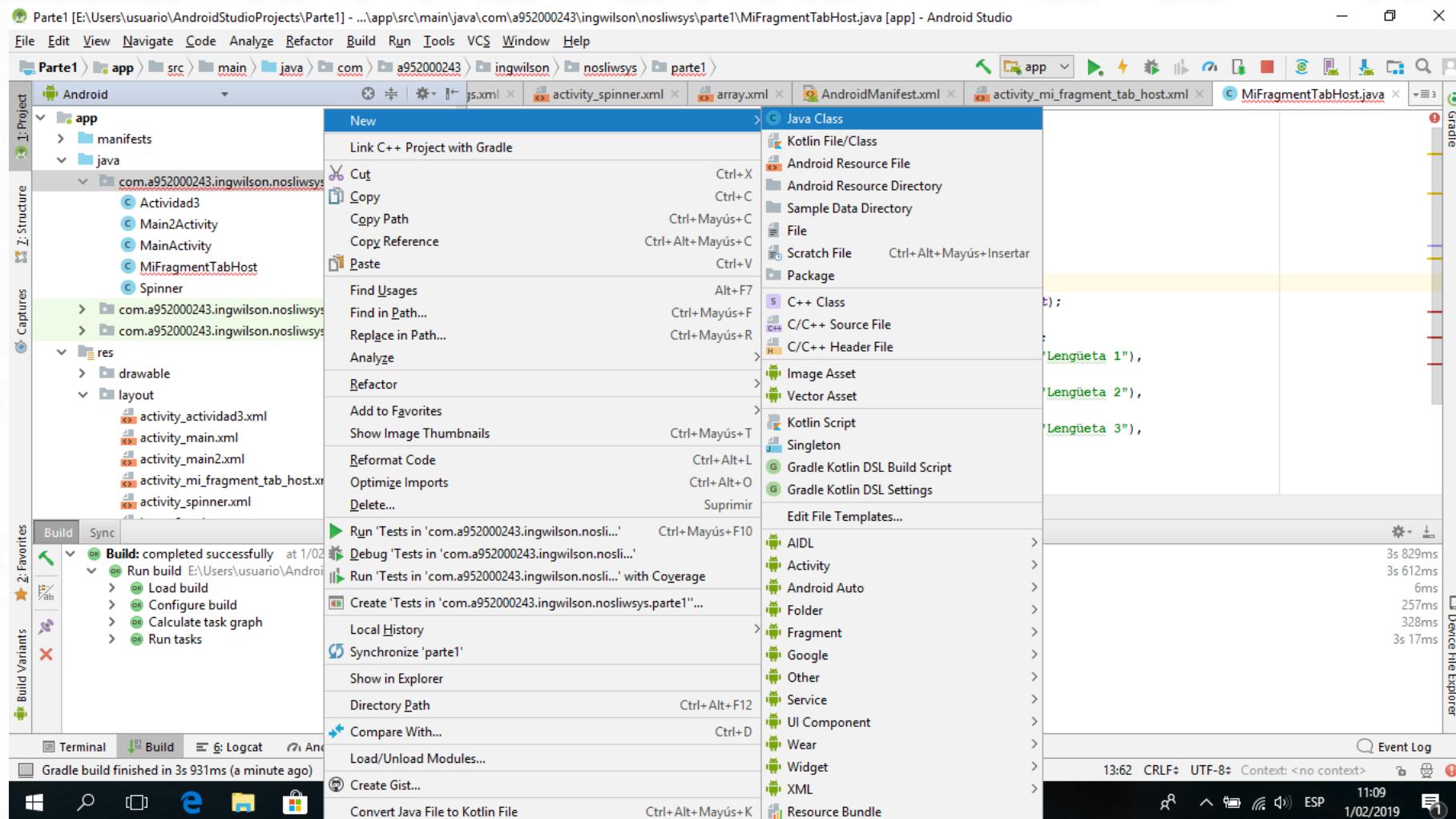
```
<androidx.fragment.app.FragmentTabHost

    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tabhost"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >
        <TabWidget
            android:id="@+id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="0"
            android:orientation="horizontal" />
        <FrameLayout
            android:id="@+id/tabcontent"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1" />
    </LinearLayout>
</androidx.fragment.app.FragmentTabHost>
```

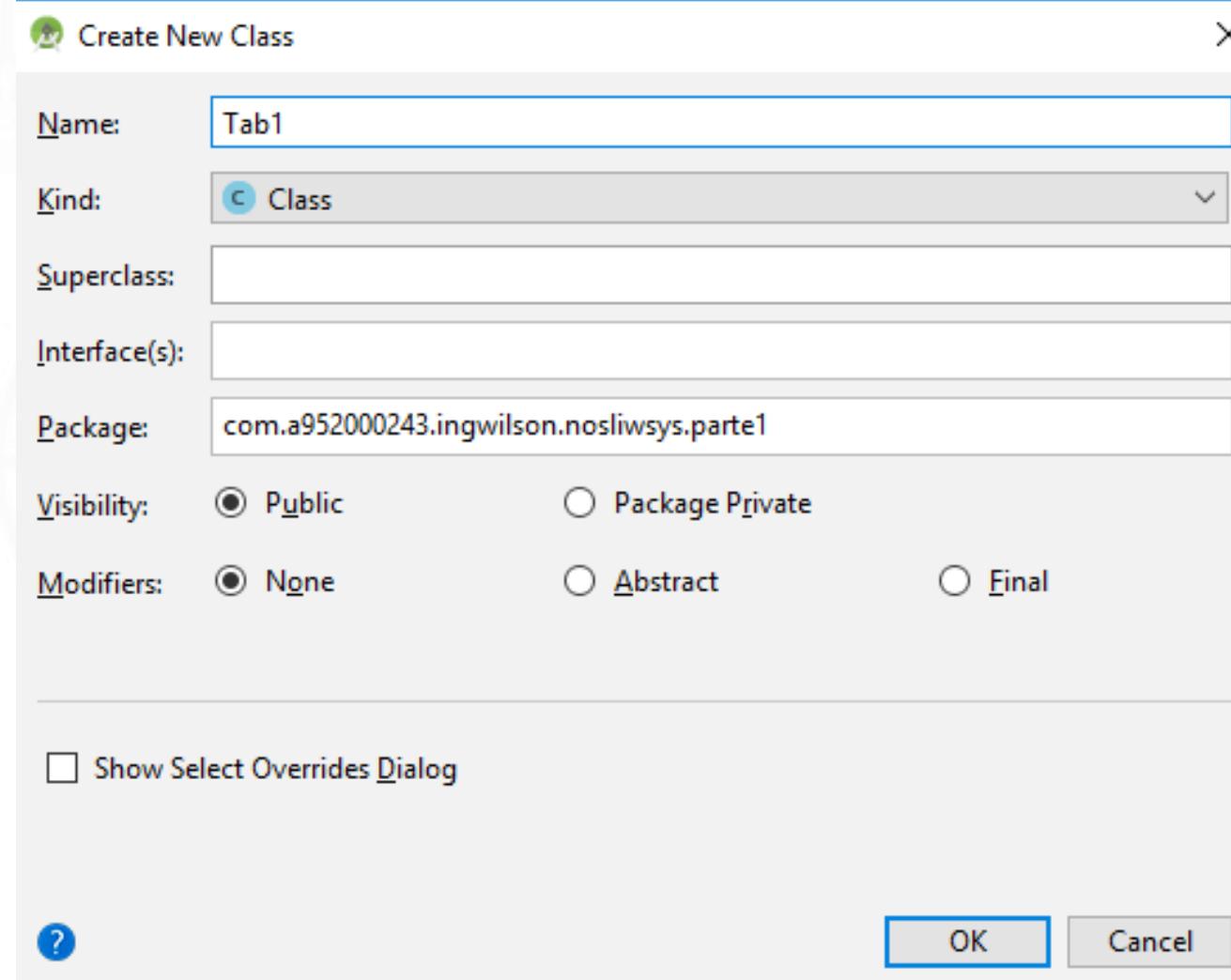
# MiFragmentTabhost.class

```
public class MiFragmentTabHost extends FragmentActivity {
    private FragmentTabHost tabHost;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mi_fragment_tab_host);
        tabHost = (FragmentTabHost) findViewById(android.R.id.tabhost);
        tabHost.setup(this,
                getSupportFragmentManager(), android.R.id.tabcontent);
        tabHost.addTab(tabHost.newTabSpec("tab1").setIndicator("Lengüeta 1"),
                Tab1.class, null);
        /*      tabHost.addTab(tabHost.newTabSpec("tab2").setIndicator("Lengüeta 2"),
                Tab2.class, null);
        tabHost.addTab(tabHost.newTabSpec("tab3").setIndicator("Lengüeta 3"),
                Tab3.class, null);*/
    }
}
```

# Creamos la clase Tab1



# Creamos la clase Tab1



# Tab1.class

```
public class tab1 extends Fragment {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState) {
        return inflater.inflate(R.layout.tab1, container, false);
    }
}
```

# Tab1.class

```
activity_mi_fragment_tabhost.xml × c MiFragmentTabhost.java × tab1.xml × c Tab1.java × activity_mi_spinner.xml × c MiSpinner.java ×  
1 package com.ingwilson.primerapractica;  
2  
3 public class Tab1 extends Fragment {  
4     @Override  
5     public void onCreate(Bundle savedInstanceState)  
6         super.onCreate(savedInstanceState);  
7     }  
8     @Override  
9     public View onCreateView(LayoutInflater inflater, ViewGroup container,  
10                             Bundle savedInstanceState) {  
11         return inflater.inflate(R.layout.tab1, container, false);  
12     }  
13 }
```

Class to Import

- Fragment (androidx.fragment.app) Gradle: androidx.fragment:fragment:1.0.0@aar (classes.jar)
- Fragment (android.app) < Android API 28 Platform > (android.jar)

# Creamos el layout tab1.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/text"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center_vertical|center_horizontal"
        android:text="Lengüeta 1"
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>
```

# FragmentTabhost

- Generamos Tab2 y Tab3 similar a Tab1
- Cambiar el llamado de layout de Tab2 por el del ejemplo1 (activity\_main)
- Cambiar el llamado de layout de Tab3 por el del ejemplo2 (layout2)
- Retiramos el comentario de los llamados al Tab2 y Tab3

# 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

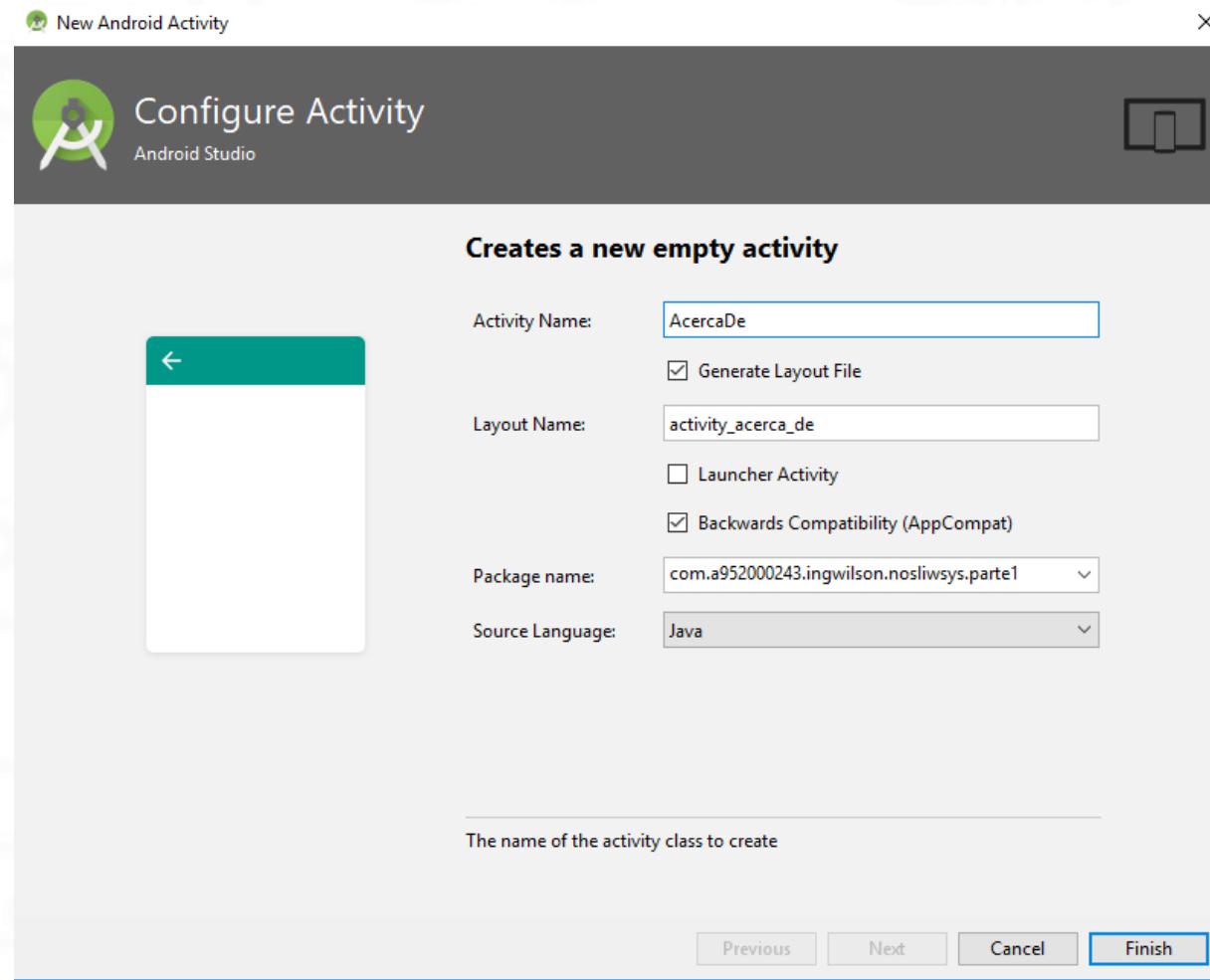
# Añadimos los siguientes botones al xml del ejemplo4(Actividad3) y añadimos el string

```
<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

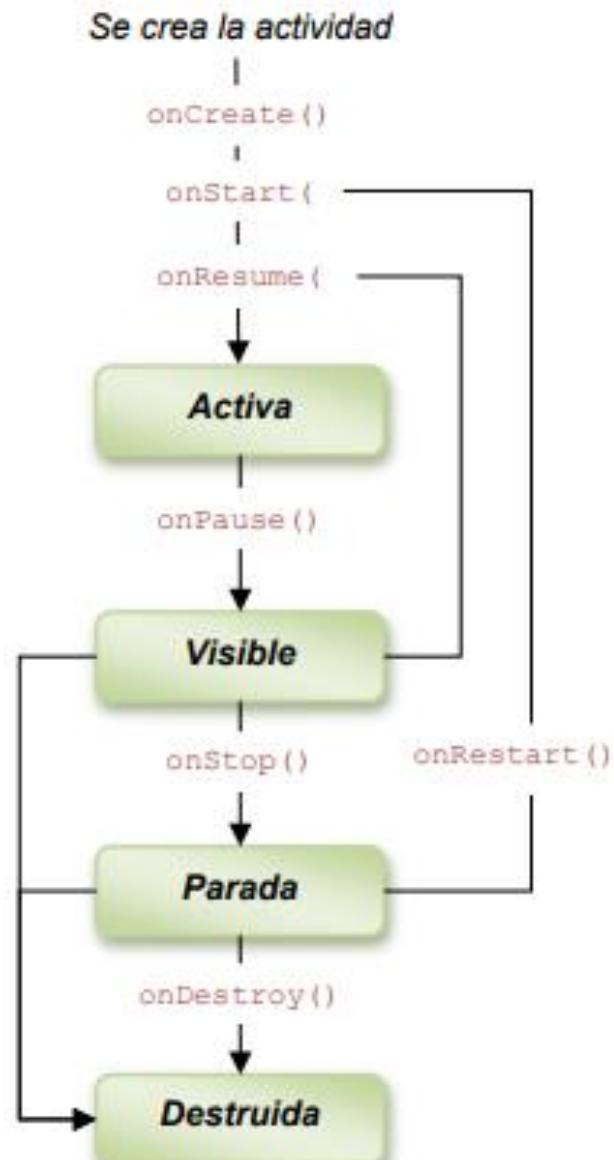
```
public class AcercaDe extends AppCompatActivity {
```

por

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

# Modificamos el Manifest

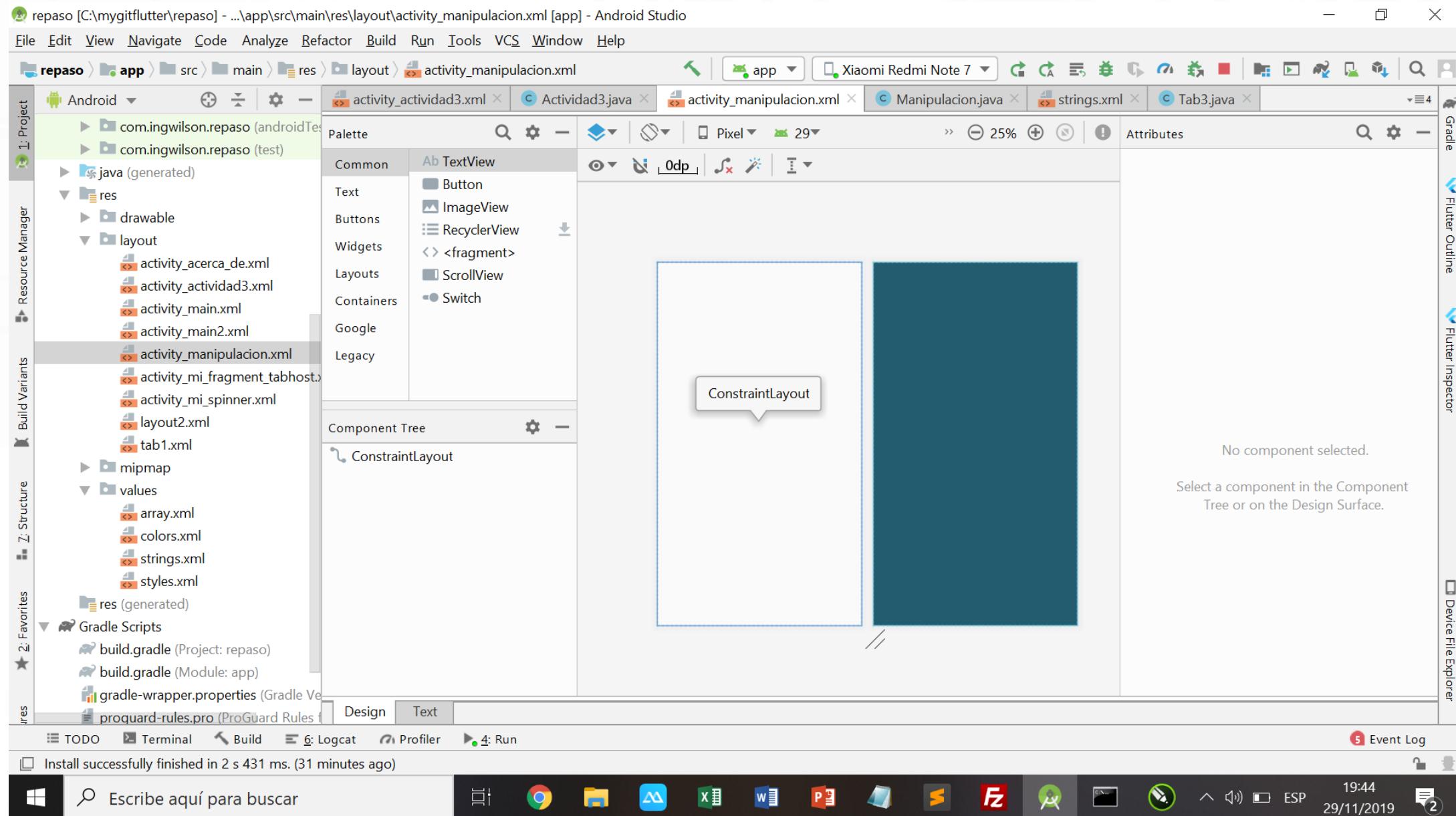
```
<activity android:name=".AcercaDe"
    android:label="Acerca de . . ."
    android:theme="@android:style/Theme.Dialog"></activity>
```

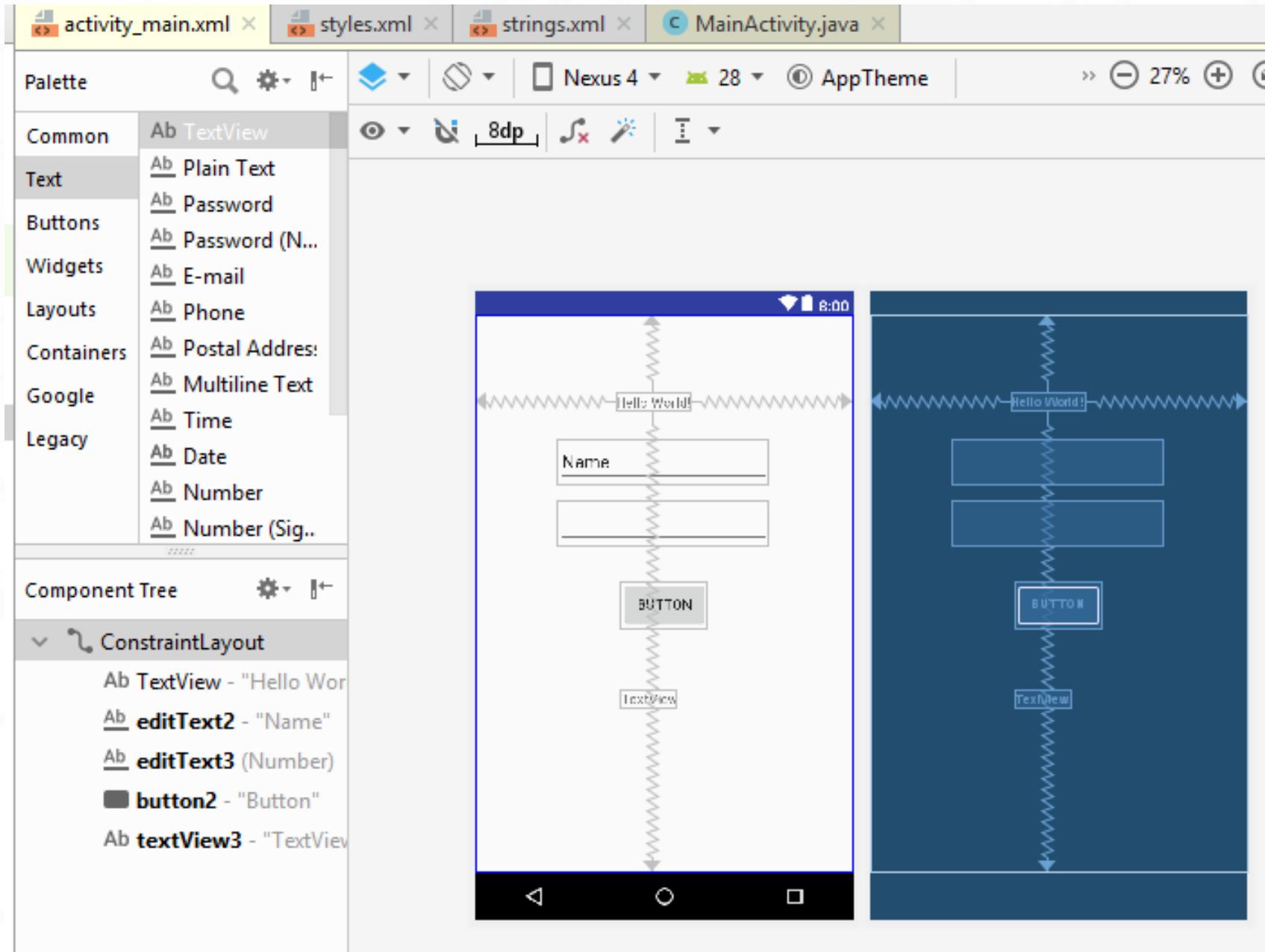


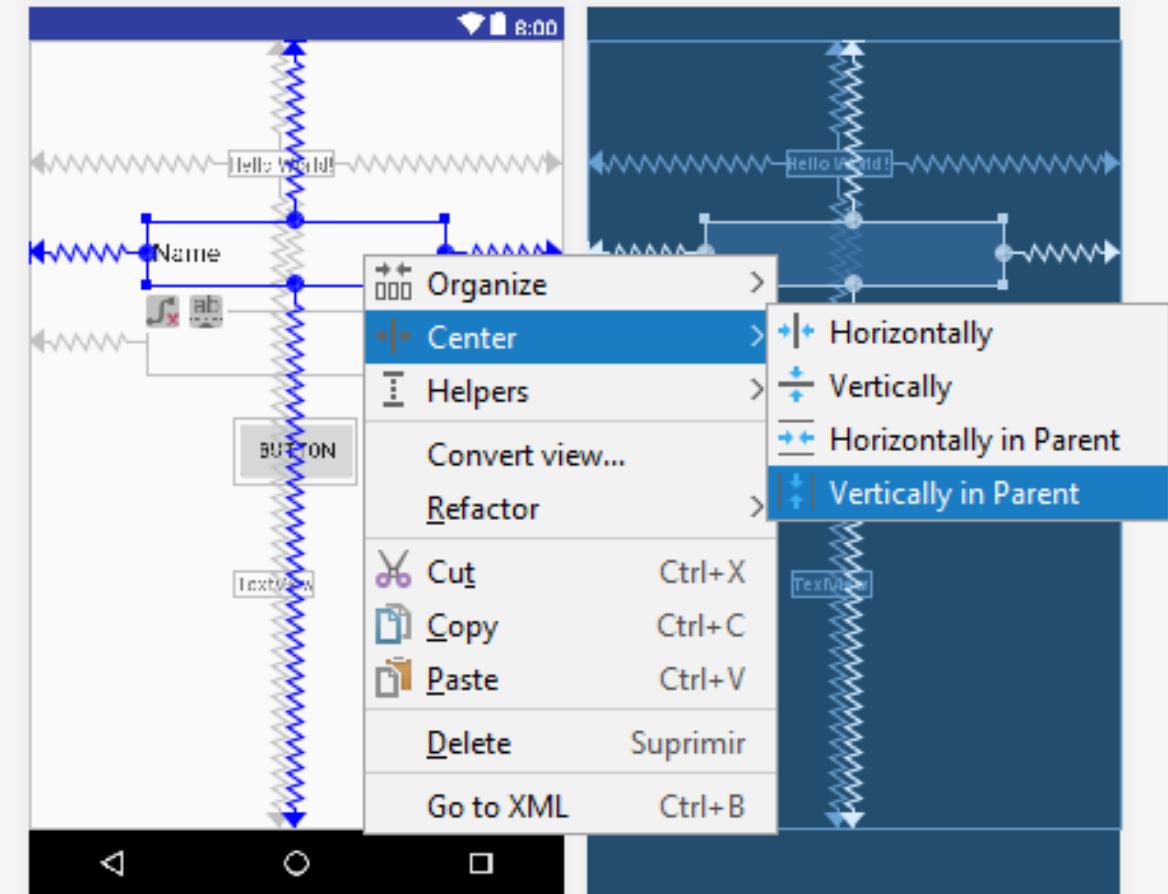
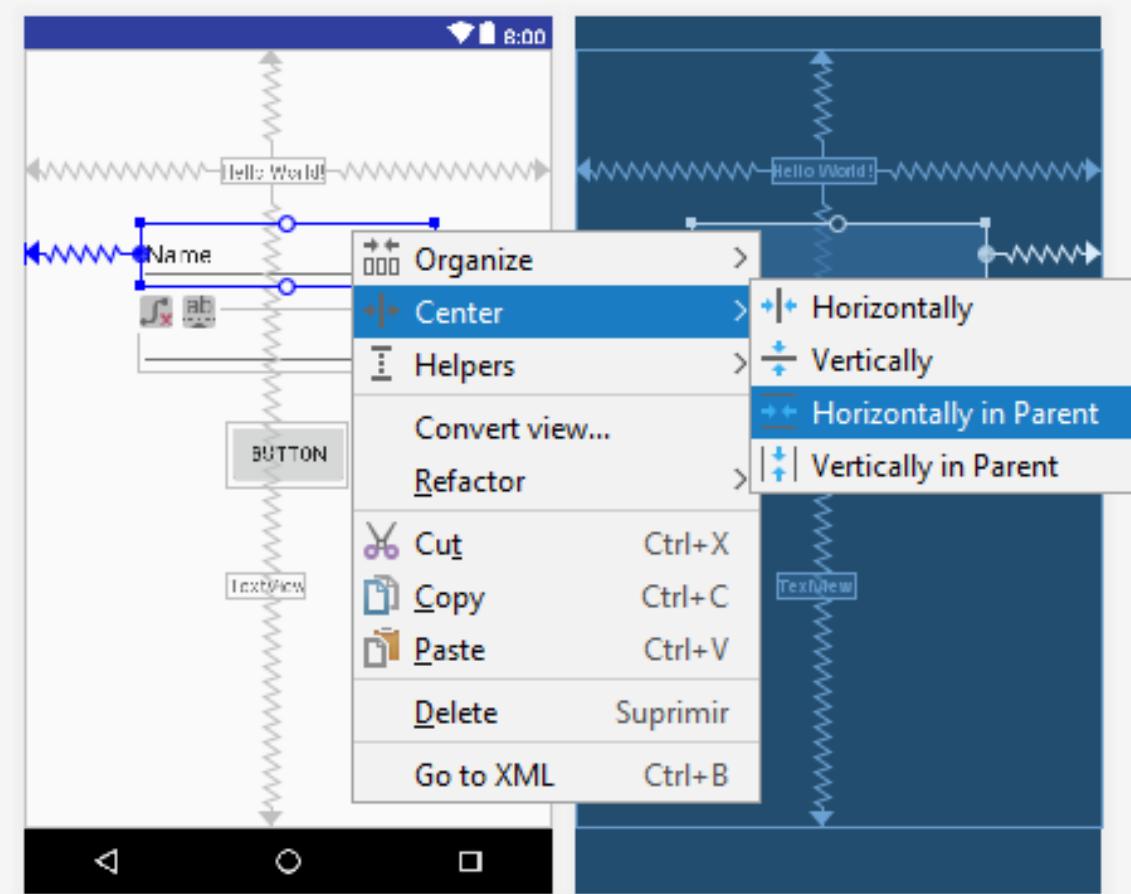
# Manipulación de elementos por código

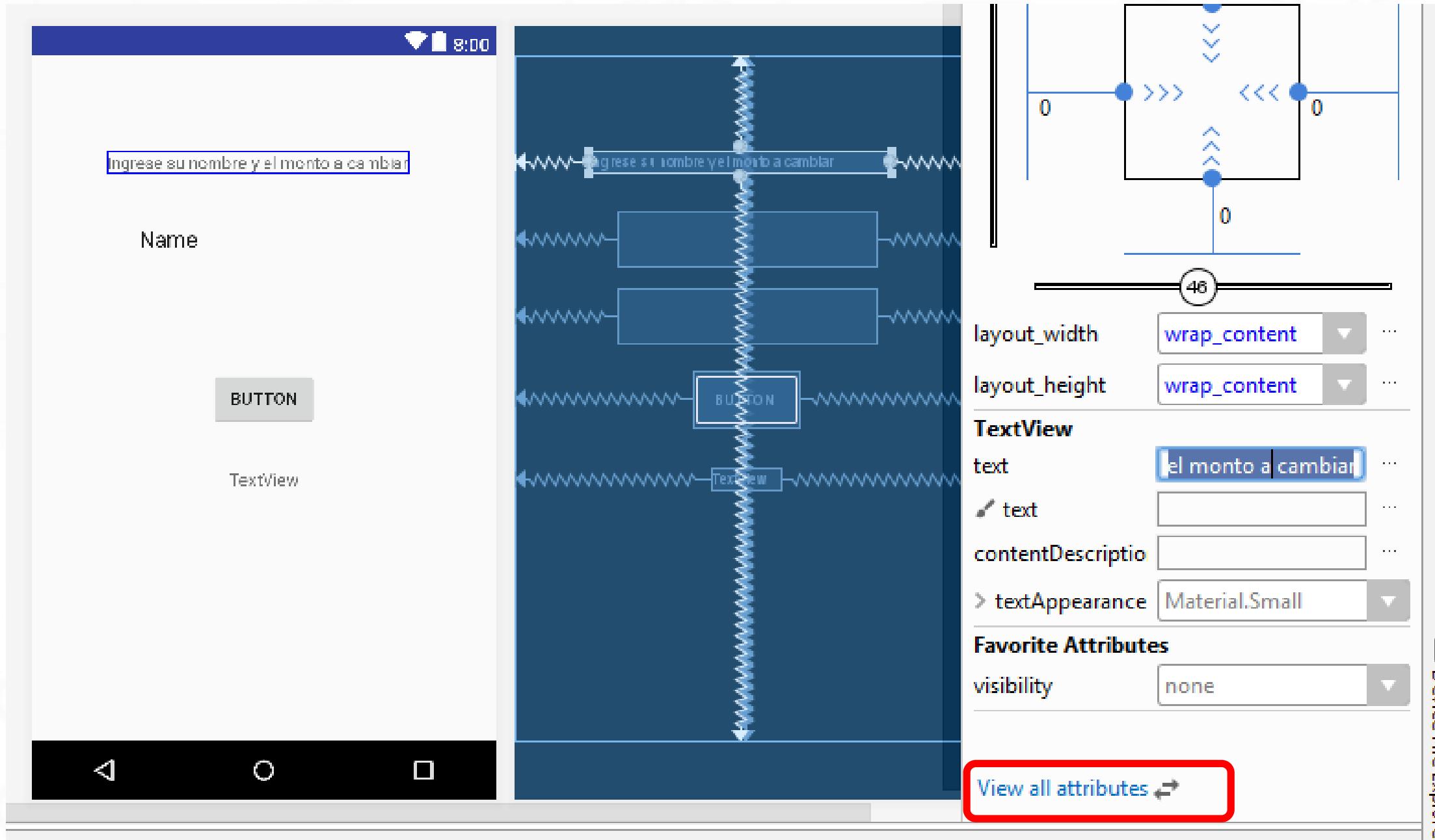
# Manipulando Elementos por Código

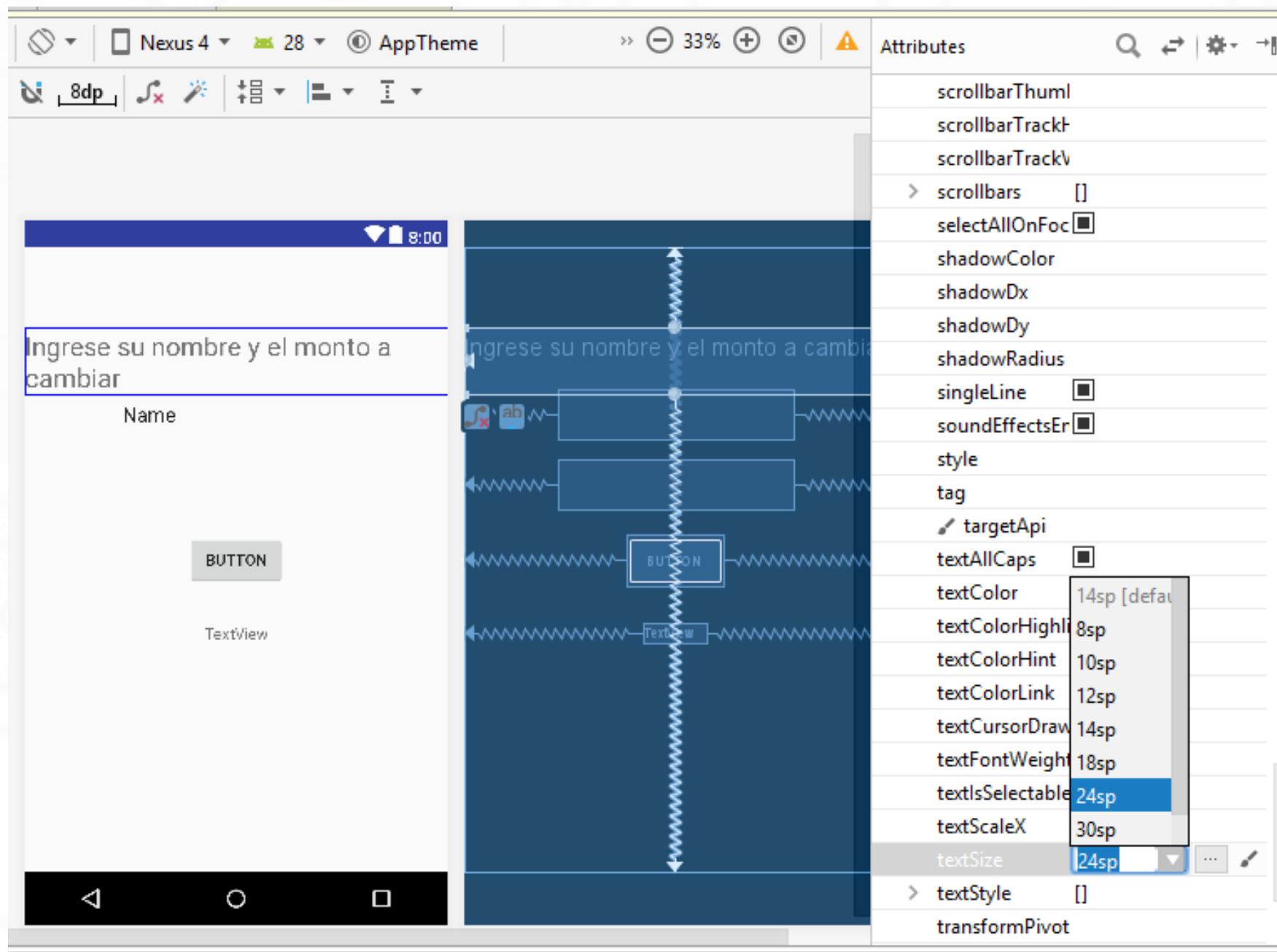
- Creamos la Actividad Manipulacion
- Generamos el Onclick respectivo en la botonera de la actividad principal (Actividad3)
- Ejecutamos para verificar el llamado
- Posteriormente usamos el modo Design para construir su layout según lo que se muestra a continuación.

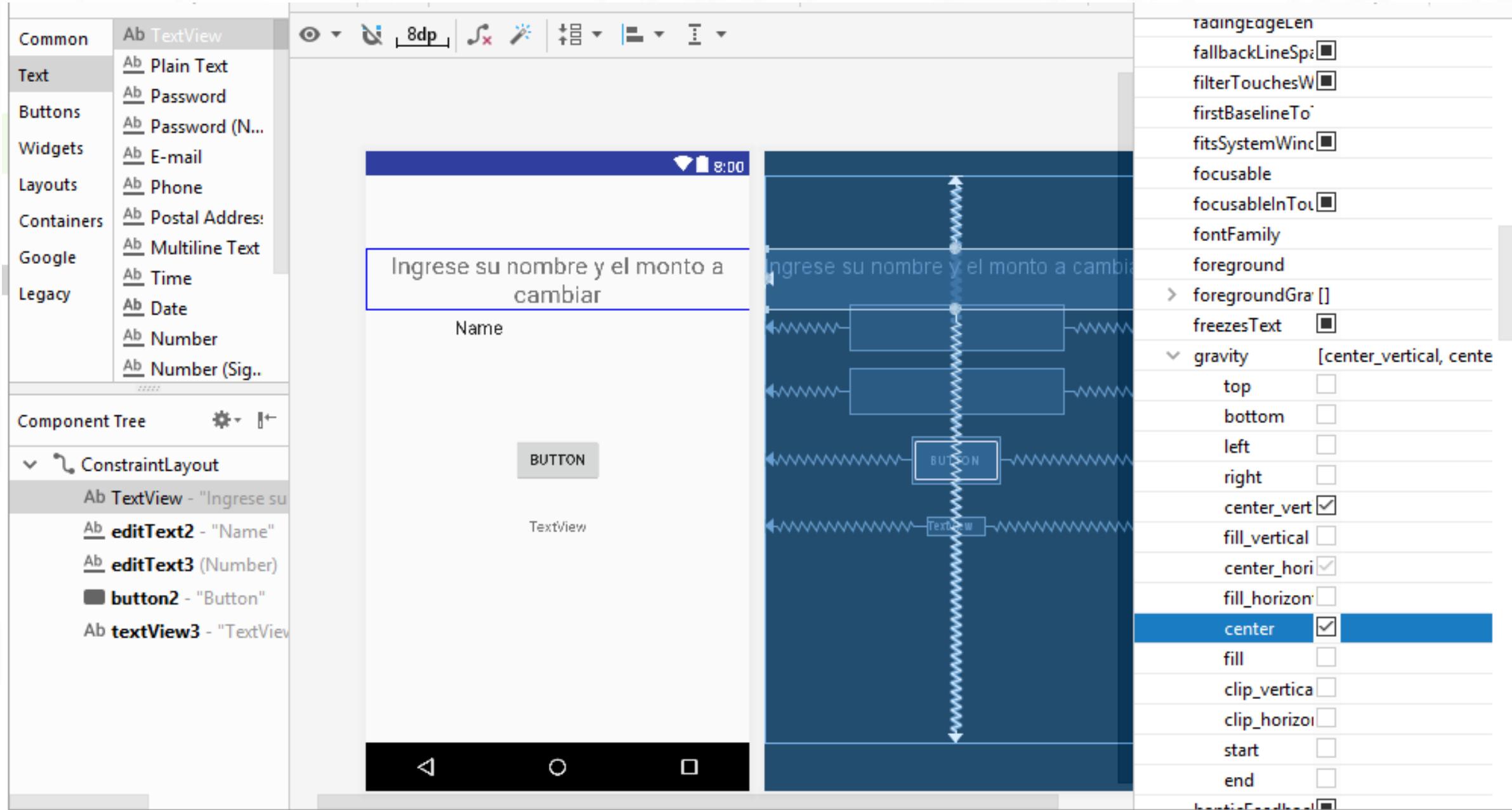


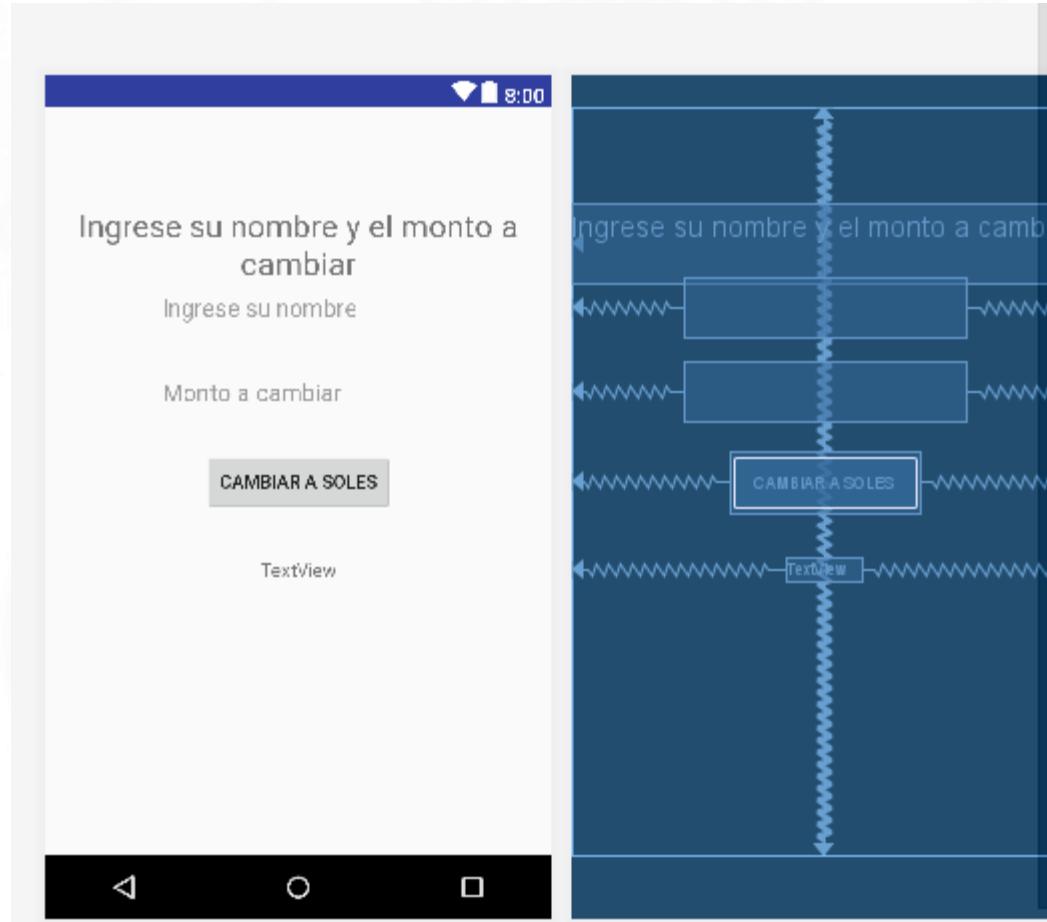












# Modificamos los identificadores de sus vistas

```
<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
                }
            });
    }
}
```

# Recycler View



- Welcome to VoicIn  
**The VoicIn** De hace 190 días  
VoicIn Headquarters
- 
- Necesitas que te ayude?  
**Support** De hace 173 días  
VoicIn Headquarters, Barcelona, Spain
- 
- Jose L.** De hace 93 días

# Pasos

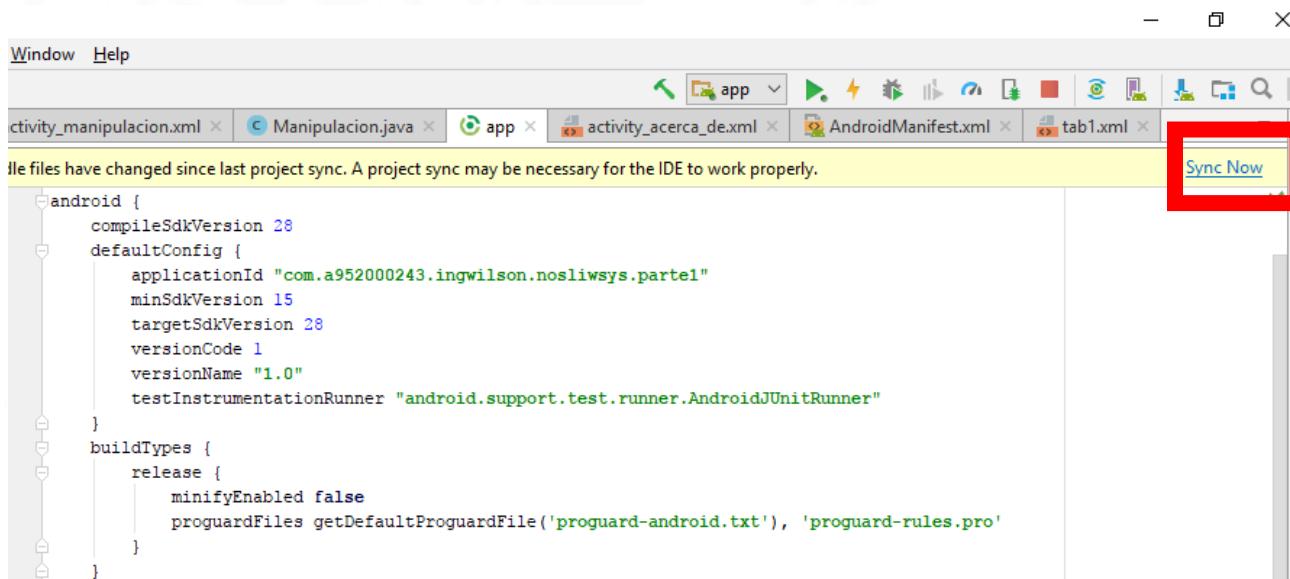
- 1. Diseñar un Layout que contiene el RecyclerView.
- 2. Implementar la actividad que lo visualice el RecyclerView.
- 3. Diseñar un layout individual que se repetirá en la lista.
- 4. Personalizar cada una de los layouts individuales según nuestros datos utilizando un adaptador.
- 5. Definir como queremos que se posicione los elementos en las vistas. Por ejemplo en forma de lista o de cuadricula.

# Ejemplo Recycler View

- Creamos la actividad MiRecyclerView
- Añadimos el onclick respectivo
- Creamos el layout miitem
- Creamos la clase Java MiAdaptador

# Pasos - Gradle

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'androidx.appcompat:appcompat:1.0.2'  
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'androidx.test:runner:1.1.1'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'  
    implementation 'androidx.recyclerview:recyclerview:1.0.0'  
}
```



# Pasos – Cambiamos su XML

```
<androidx.recyclerview.widget.RecyclerView  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/recycler_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_marginLeft="10dp"  
    android:layout_marginRight="10dp" />
```

# MiRecyclerView.class

```
public class MiRecyclerView extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.LayoutManager layoutManager;
    private MiAdaptador adaptador;
    private Vector<String> misdatos;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mirecyler);
        recyclerView = findViewById(R.id.recyclerview_view);

        misdatos = new Vector<String>();
        misdatos.add("123000 Wilson Callisaya");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("111000 Pedro Martinez");
        misdatos.add("011000 Paco Pérez");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("111000 Pedro Martinez");
        misdatos.add("011000 Paco Pérez");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("111000 Pedro Martinez");
        misdatos.add("011000 Paco Pérez");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("123000 Pepito Domingez");
        misdatos.add("111000 Pedro Martinez");
        misdatos.add("011000 Paco Pérez");

        adaptador = new MiAdaptador(this,
            misdatos);
        recyclerView.setAdapter(adaptador);
        layoutManager = new LinearLayoutManager(this);
        recyclerView.setLayoutManager(layoutManager);
    }
}
```

# Pasos - miitem

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingTop="4dp"
    android:paddingBottom="4dp"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/icono"
        android:layout_width="?android:attr/listPreferredItemHeight"
        android:layout_height="wrap_content"
        android:src="@mipmap/ic_launcher" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView android:id="@+id/titulo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            />
        <TextView android:id="@+id/subtitulo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Otro Texto"
            android:gravity="center"/>
        <Button
            android:id="@+id/mibotonitem"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="miboton"
            android:visibility="gone"
            />
    </LinearLayout>
</LinearLayout>
```

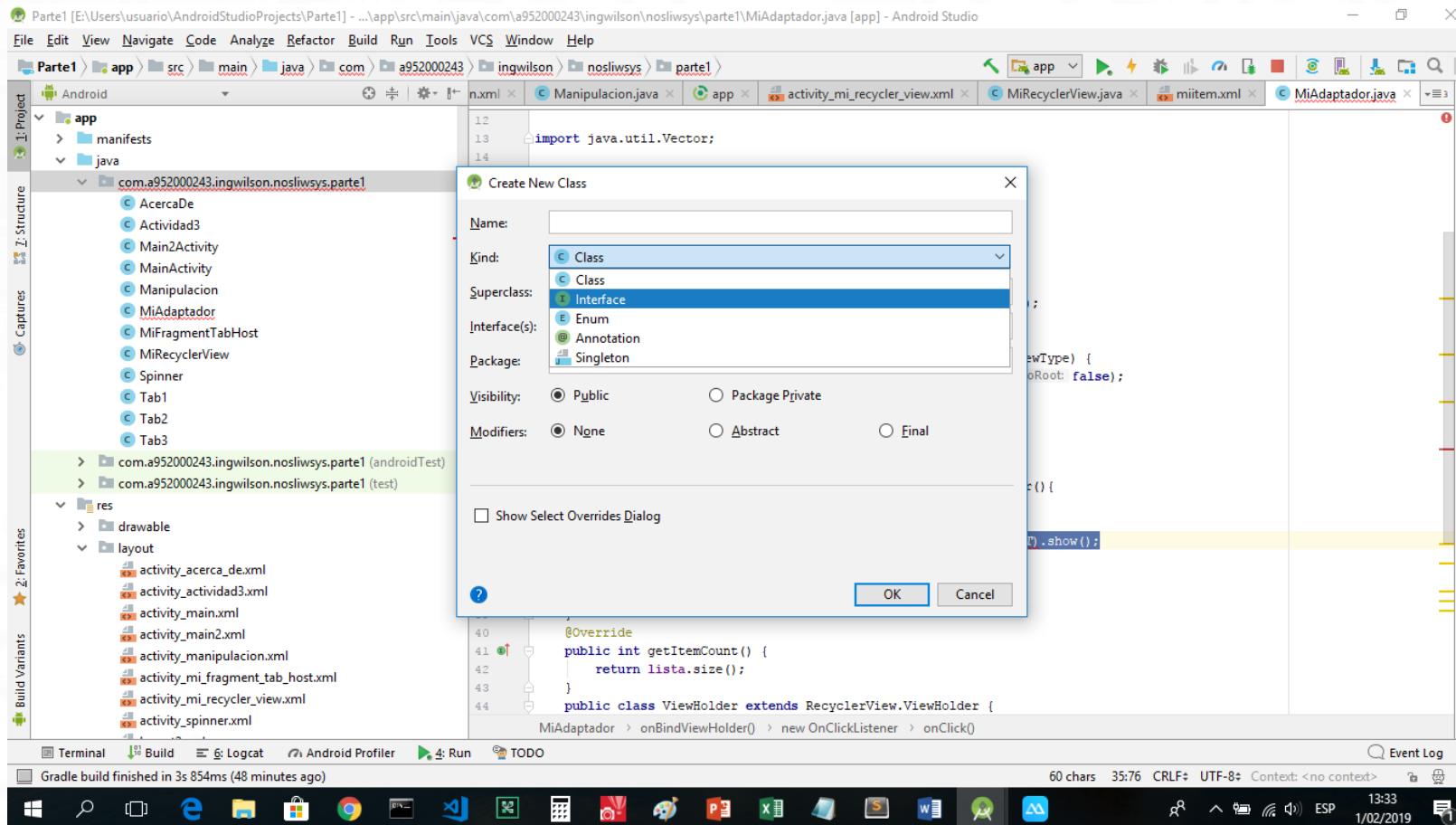
# Pasos - MiAdaptador

```
public class MiAdaptador extends
    RecyclerView.Adapter<MiAdaptador.ViewHolder> {
    private LayoutInflater inflador;
    private Vector<String> lista;
    public MiAdaptador(Context context, Vector<String> lista) {
        this.lista = lista;
        inflador = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View v = inflador.inflate(R.layout.miitem, parent, false);
        return new ViewHolder(v);
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int i) {
        holder.titulo.setText(lista.get(i));
    }
    @Override
    public int getItemCount() {
        return lista.size();
    }
    public class ViewHolder extends RecyclerView.ViewHolder {
        public TextView titulo, subtítulo;
        public ImageView icon;
        ViewHolder(View itemView) {
            super(itemView);
            titulo = (TextView)itemView.findViewById(R.id.titulo);
            subtítulo = (TextView)itemView.findViewById(R.id.subtítulo);
            icon = (ImageView)itemView.findViewById(R.id.icono);
        }
    }
}
```

# Añadiendo un botón con acciones al Recycler View

# Generamos la siguiente Interface similar a la creación de una Java Class con el nombre InterfaceToast



# Moficamos la interface

```
public interface InterfaceToast {  
    void mensaje(String mensaje);  
}
```

# Moficar Adapter

```
public class MiAdaptador extends RecyclerView.Adapter<MiAdaptador.ViewHolder> {
    private LayoutInflater inflador;
    private Vector<String> lista;
    Context micontext;
    public MiAdaptador(Context context, Vector<String> lista) {
        this.lista = lista;
        micontext=context;
        inflador = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View v = inflador.inflate(R.layout.mitem, parent, false);
        return new ViewHolder(v);
    }
    @Override
    public void onBindViewHolder(ViewHolder holder, final int i) {
        holder.titulo.setText(lista.get(i));
        holder.miboton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ((InterfaceToast)micontext).mensaje("El item de la posición: "+ i + " Su valor es: " + lista.get(i));
            }
        });
    }
    @Override
    public int getItemCount() {
        return lista.size();
    }
    public class ViewHolder extends RecyclerView.ViewHolder {
        public TextView titulo, subtituto;
        public ImageView icon;
        public Button miboton;
        ViewHolder(View itemView) {
            super(itemView);
            titulo = (TextView)itemView.findViewById(R.id.titulo);
            subtituto = (TextView)itemView.findViewById(R.id.subtituto);
            icon = (ImageView)itemView.findViewById(R.id.icono);
            miboton= itemView.findViewById(R.id.mibotonitem);
        }
    }
}
```

# Implementamos la interface

```
public class MiRecyclerView extends AppCompatActivity implements InterfaceToast {
```

Añadimos el siguiente método al final

```
@Override  
public void mensaje(String mensaje) {  
    Toast.makeText(this, mensaje, Toast.LENGTH_SHORT).show();  
}
```

# Cambiamos a visible el botón de miiitem

```
<Button  
    android:id="@+id/mibotonitem"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="miboton"  
    android:visibility="visible"  
/>
```

# GITHUB ANDROID Y APIDEMOS

# 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

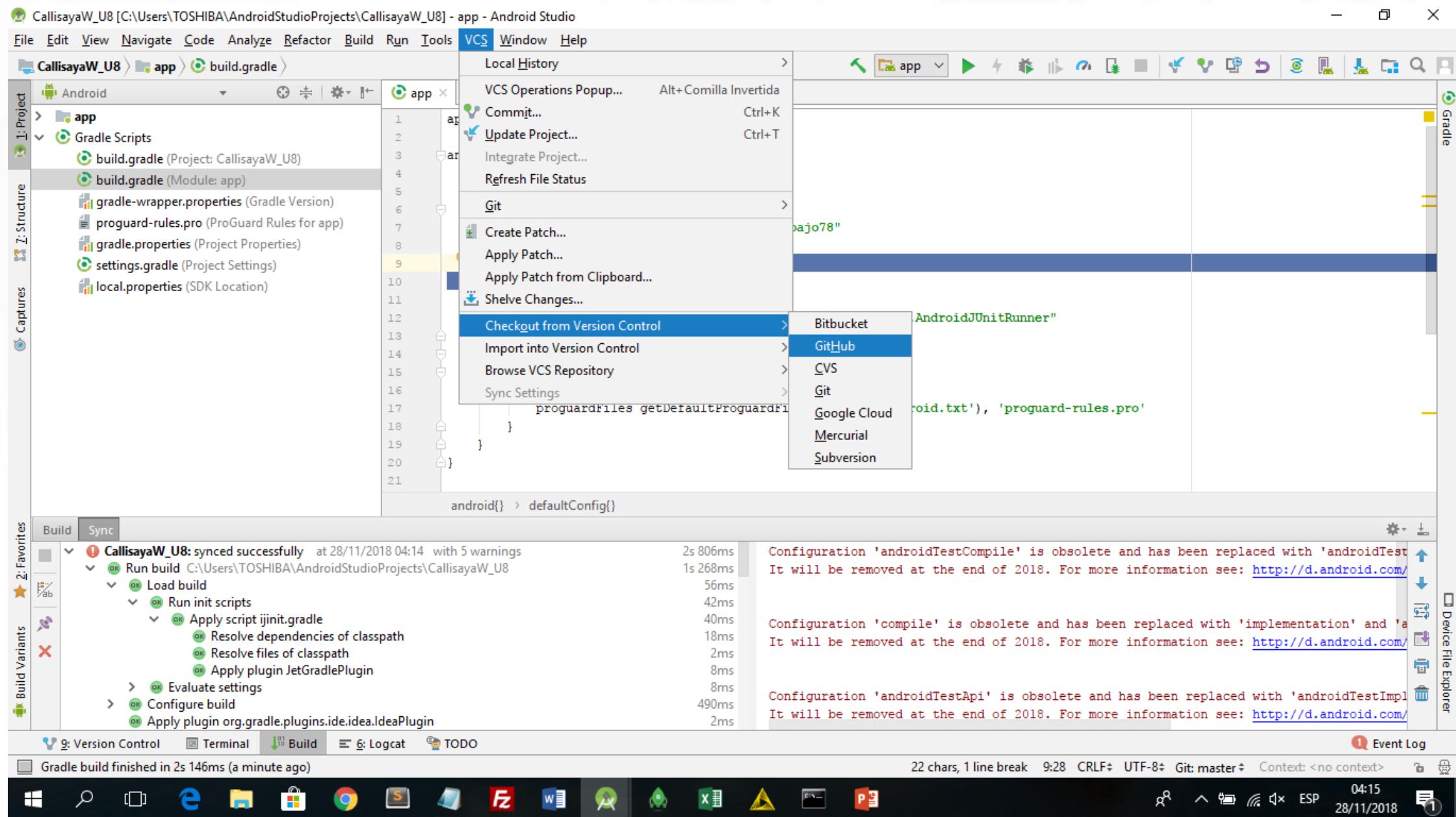
subir archivos al repositorio

git status –s

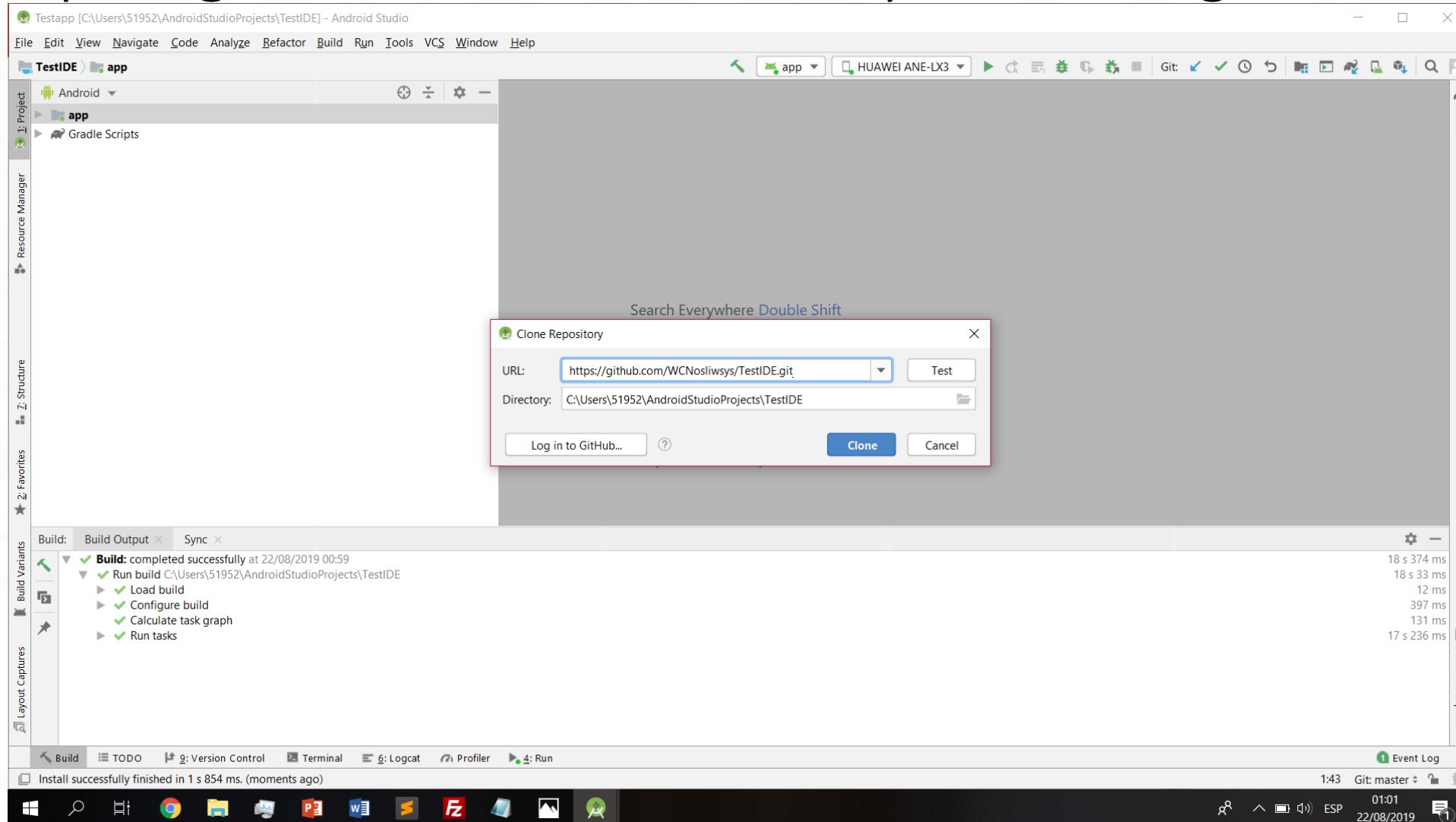
para ver el estado de los archivos

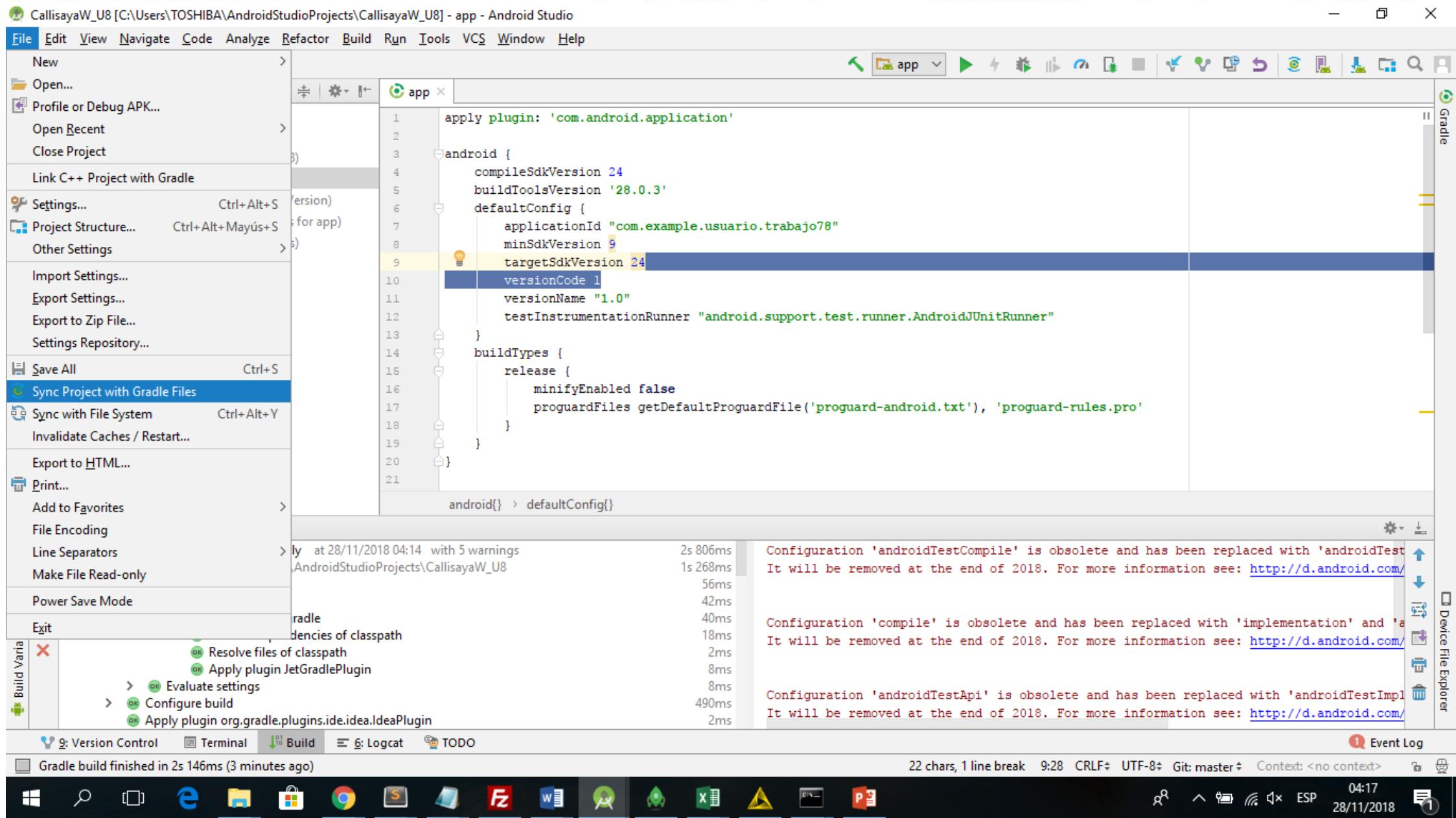
<https://github.com/>

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>

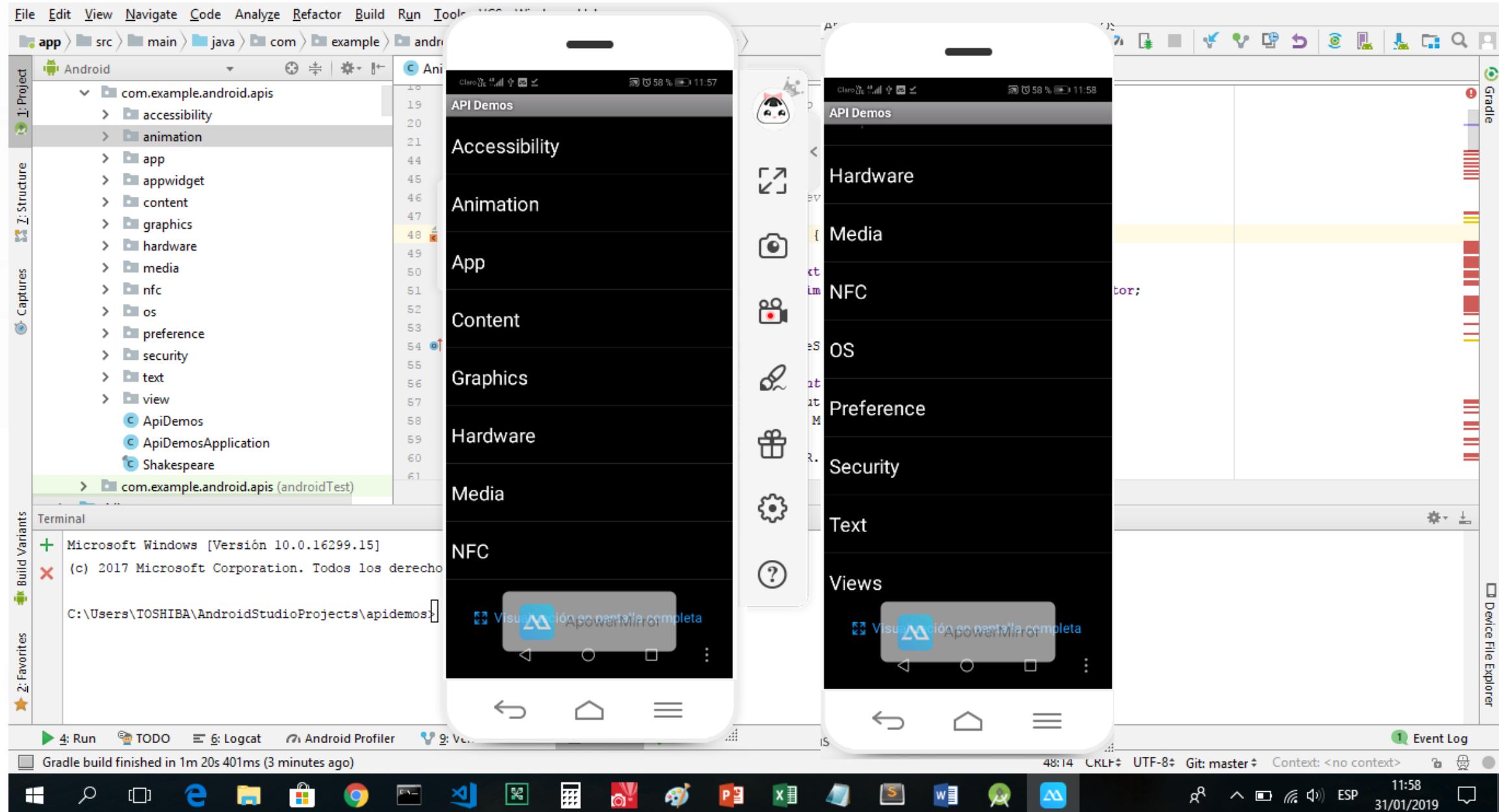




<https://github.com/WCNosliwsys>

<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<sup>a</sup> ED. 2017

## Proyectos Android Fundamentos 8<sup>a</sup> 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



Universidad Nacional  
Jorge Basadre Grohmann



# Gracias por su atención.

MSc. Ing. Wilson César Callisaya Choquecota

[nosliwsys@gmail.com](mailto:nosliwsys@gmail.com)

952000243 – Tacna Perú