

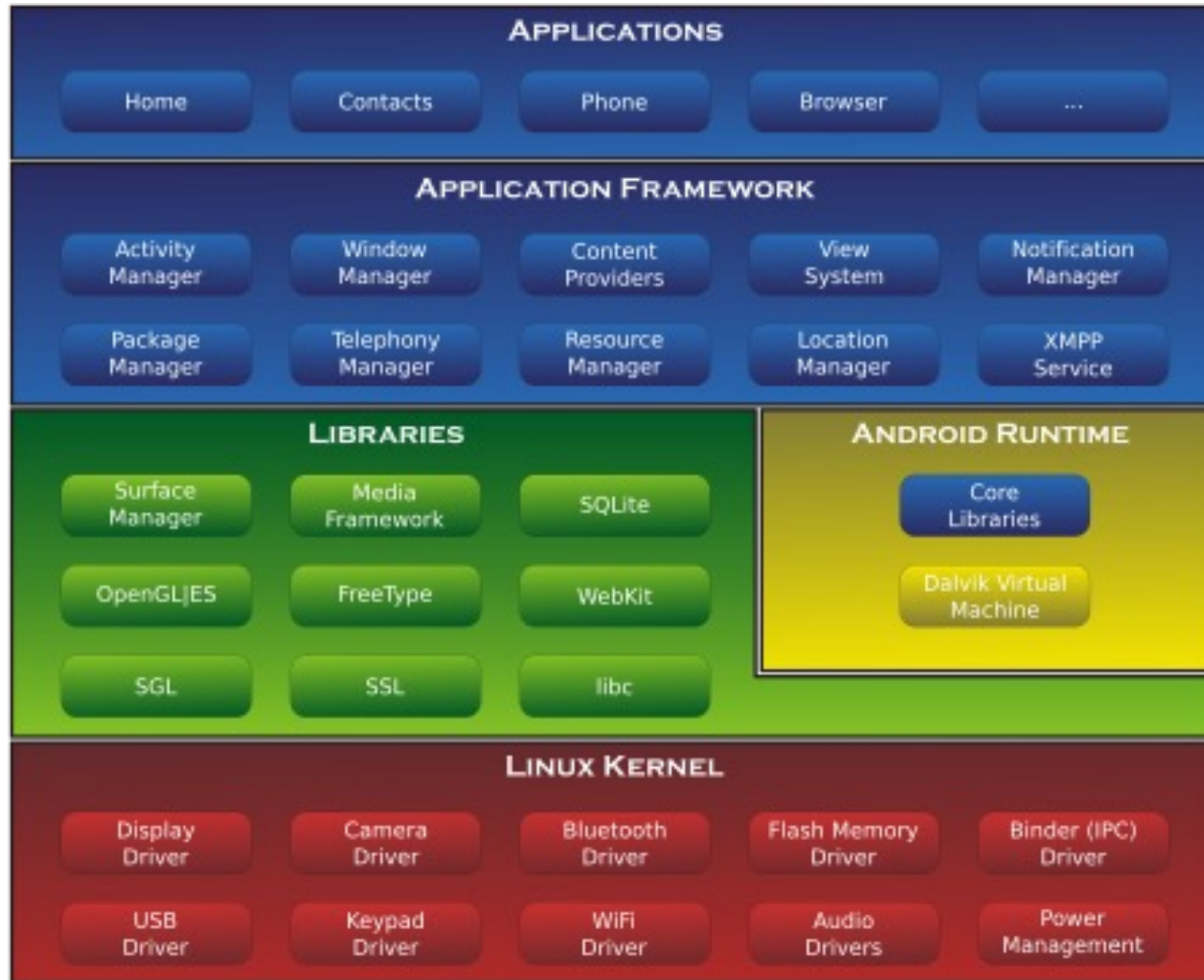
Conceptos Básicos

Desarrollo para plataformas móviles



Borja Martin Herrera
Borja.herrera@u-tad.com

Radiografía Android



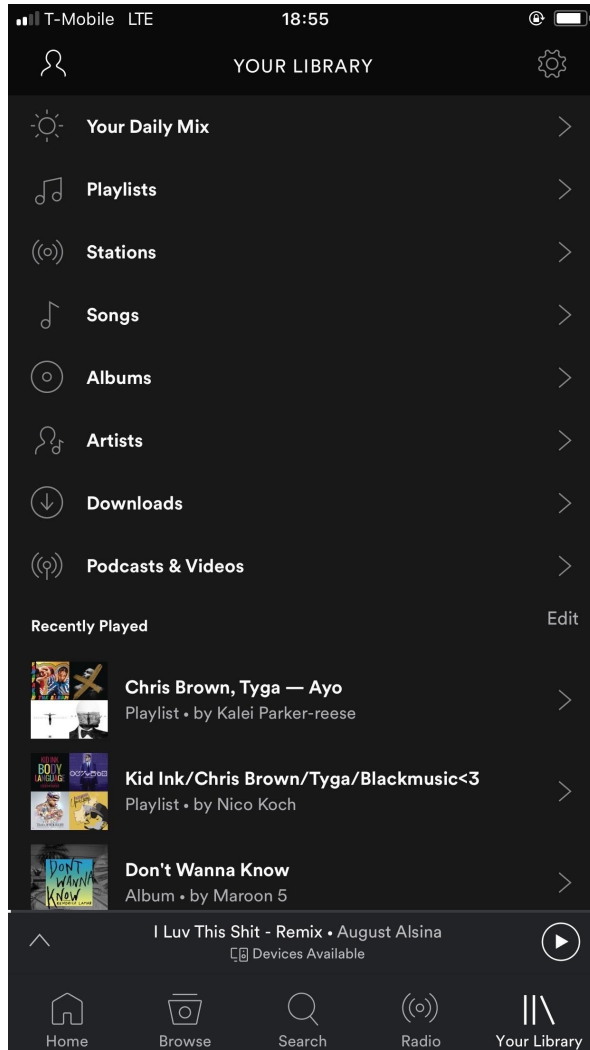
Fuente: Wikipedia

Radiografía Android

- Application : Las aplicaciones incluidas en Android como un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros.
- Application Framework : Son una serie de API del entorno de trabajo que podemos usar a la hora de realizar nuestras apps.
- Libraries: Bibliotecas de C/C++ usadas por varios componentes del sistema. Tenemos: System C library (implementación biblioteca C estándar), bibliotecas de gráficos, 3D y SQLite, entre otras.
- Runtime de Android: Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik.
- Núcleo Linux: Android depende de Linux para seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores.

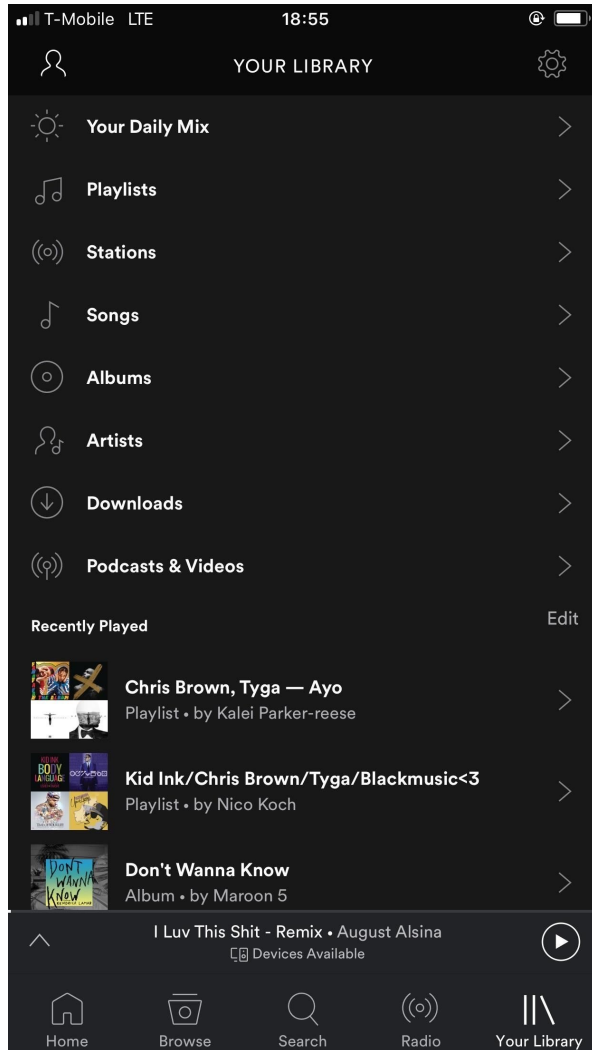
Fuente: Wikipedia

Radiografía App



- Las app de Android se dividen en pantallas.
- Cada pantalla tiene un código fuente que dice que tiene que hacer y un xml que dice lo que se tiene que pintar en ella.
- Layout:
 - Es lo que se muestra en la pantalla y la manera de hacerlo.
 - Viene definido en un xml.
- Activity:
 - Es la funcionalidad de lo que se muestra.
 - Es una clase de Java.

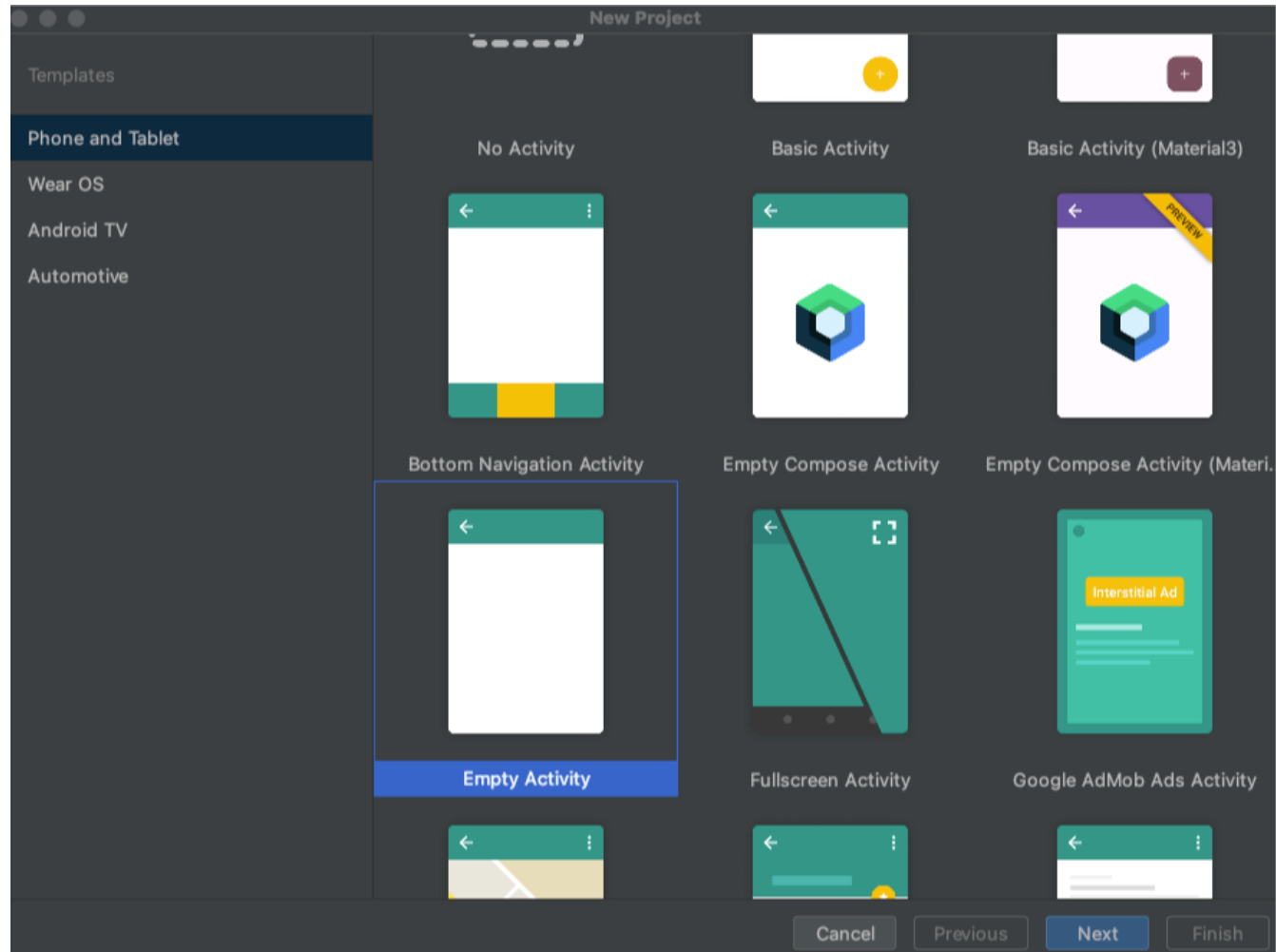
Funcionamiento de una App



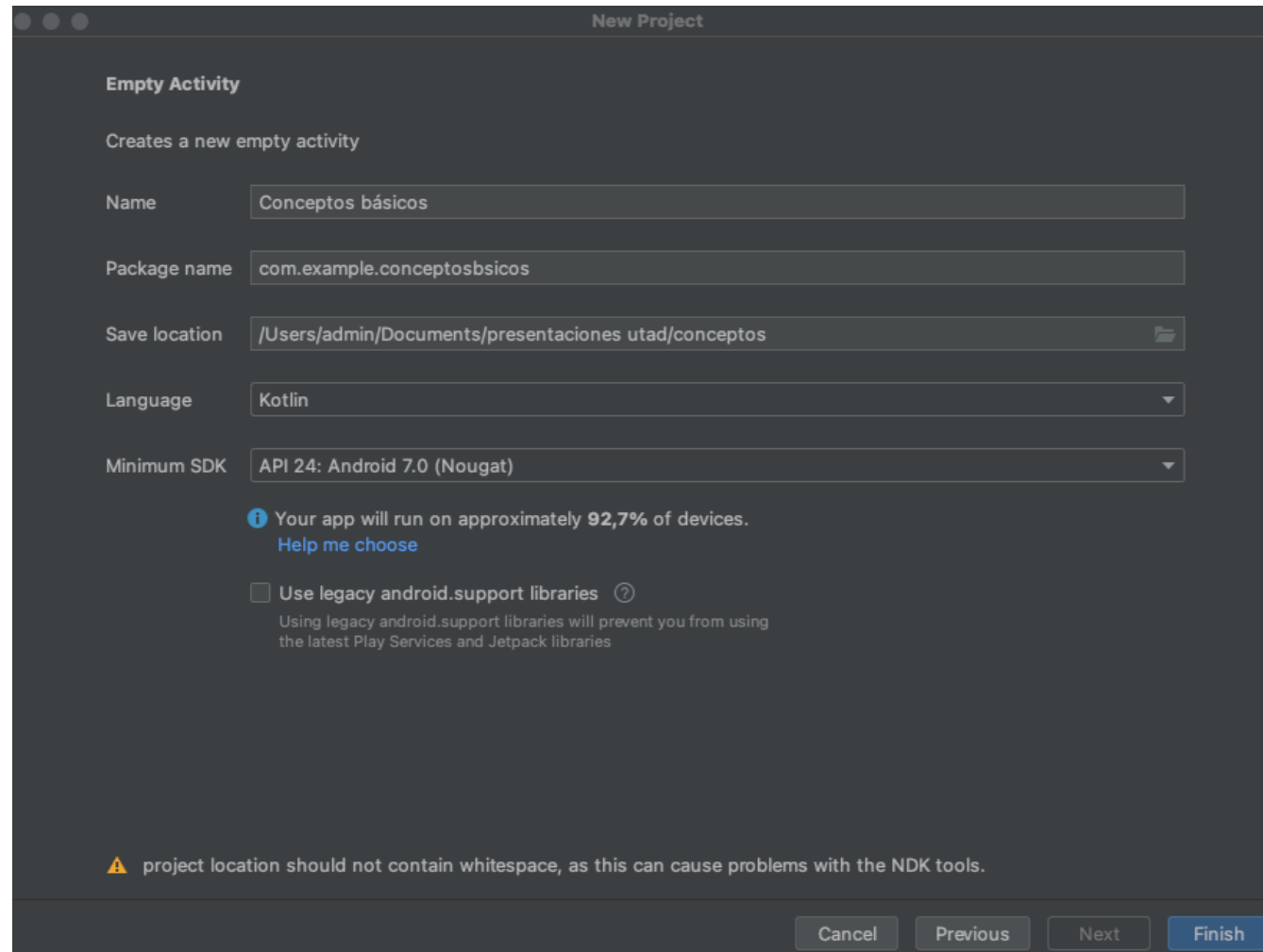
- Toda app de Android empieza creando un objeto del tipo de Actividad que se defina como principal.
- Está actividad define el *layout* que tiene asociado y lo crea mostrándolo en el dispositivo.
- El usuario interactúa con ese *layout*. La actividad recibe esos eventos y decide que es lo que se tiene que hacer.
- Por último la actividad actualiza el *layout*.

Ejemplo de App

- Creamos un nuevo proyecto



Ejemplo de App



New Project

Empty Activity

Creates a new empty activity

Name: Conceptos básicos

Package name: com.example.conceptosbsicos

Save location: /Users/admin/Documents/presentaciones utad/conceptos

Language: Kotlin

Minimum SDK: API 24: Android 7.0 (Nougat)

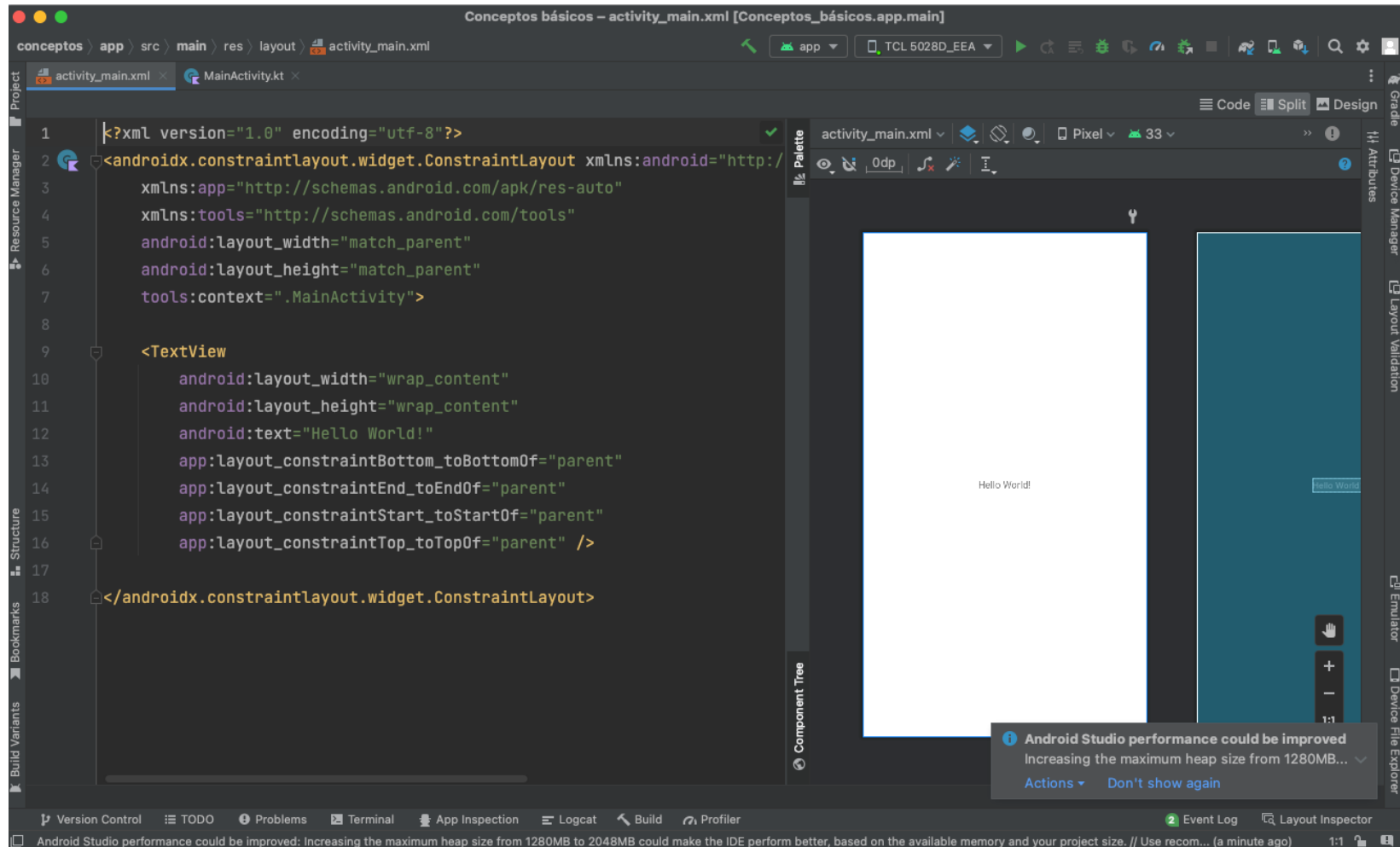
i Your app will run on approximately **92,7%** of devices.
[Help me choose](#)

☐ Use legacy android.support libraries **?**
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

⚠ project location should not contain whitespace, as this can cause problems with the NDK tools.

Cancel Previous Next Finish

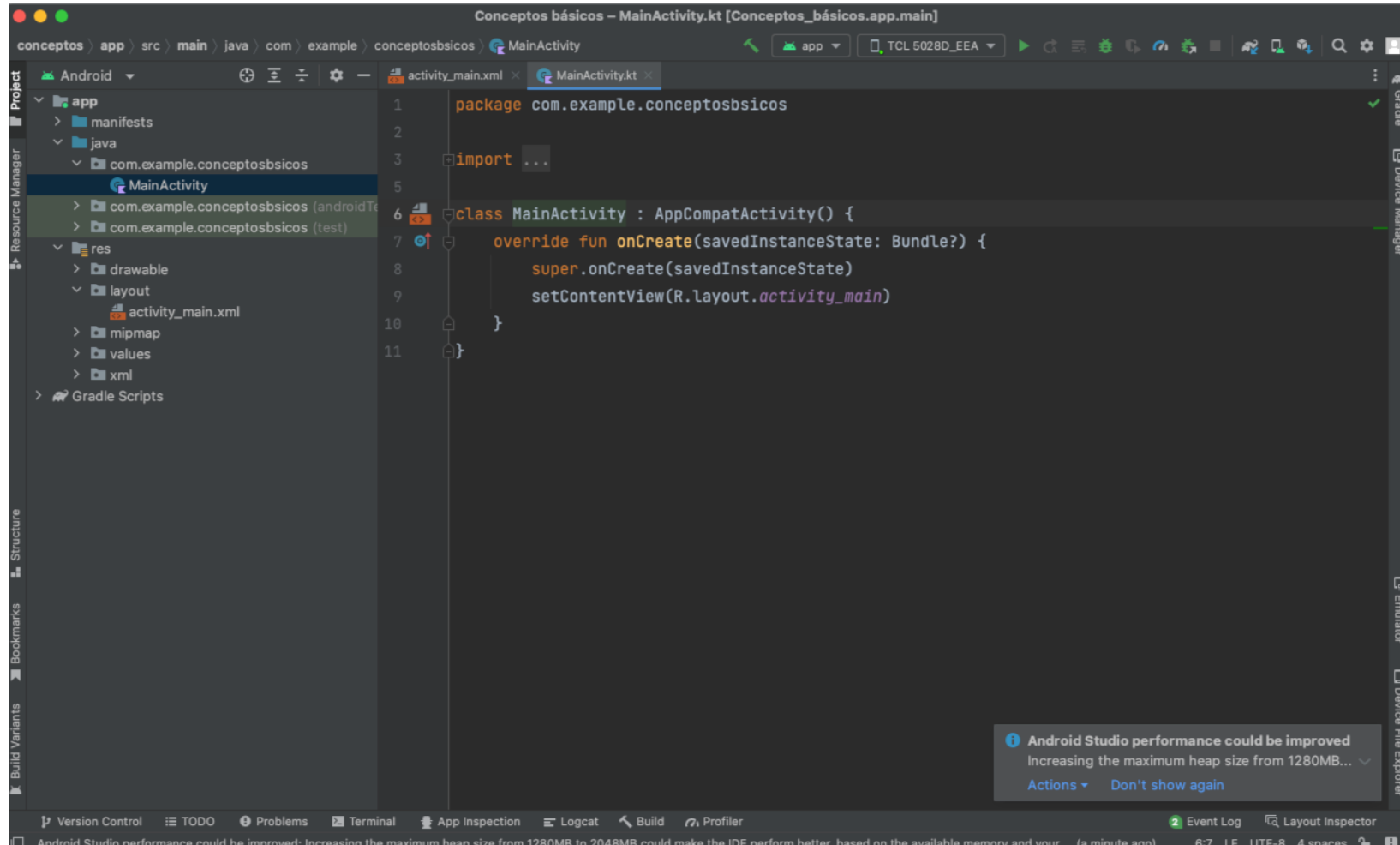
Ejemplo de App



Modo texto

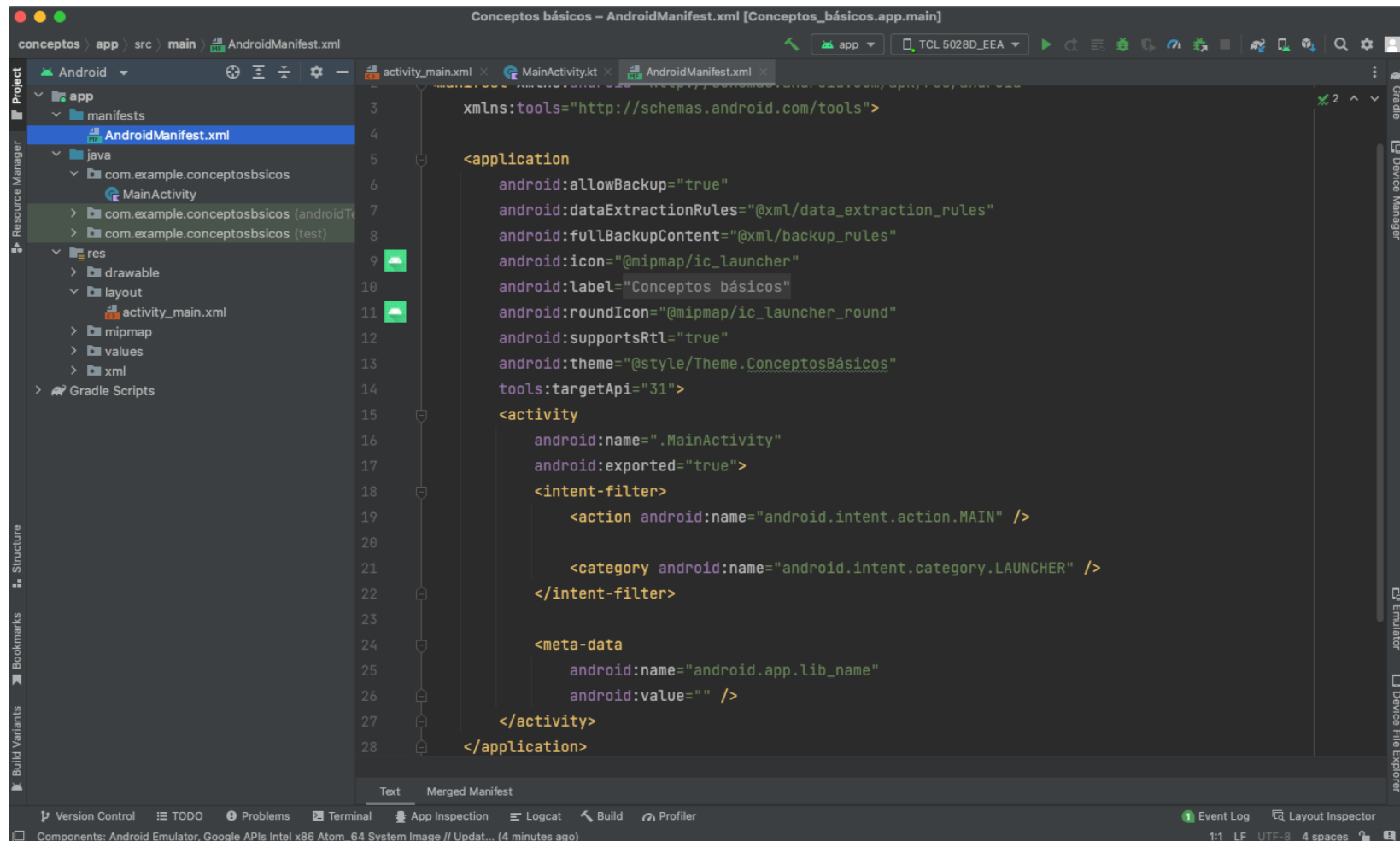
Ejemplo de App

La actividad (MainActivity)



Ejemplo de App

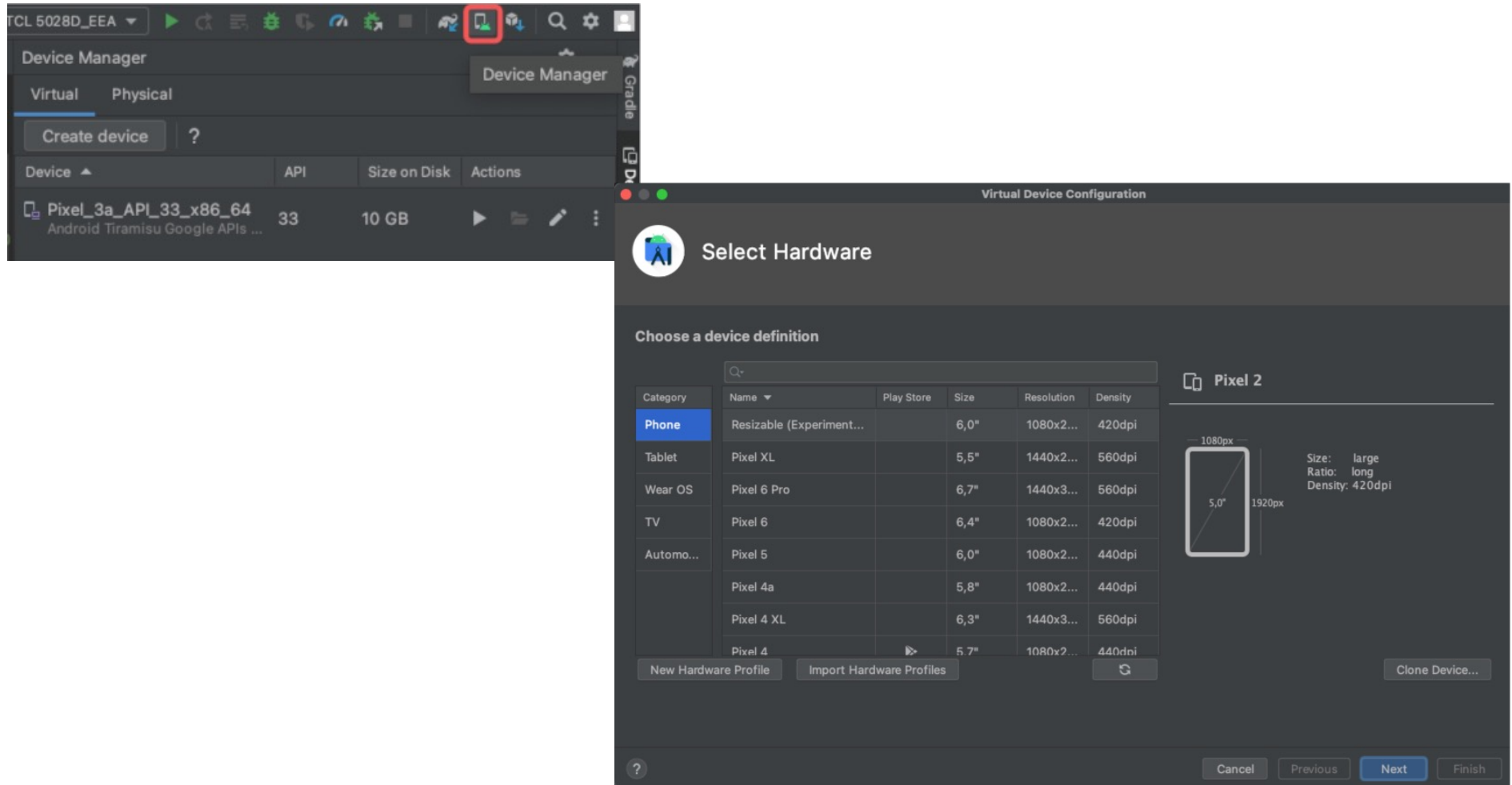
- En el archivo AndroidManifest.xml definimos las actividades que tiene la app y cual es la principal.



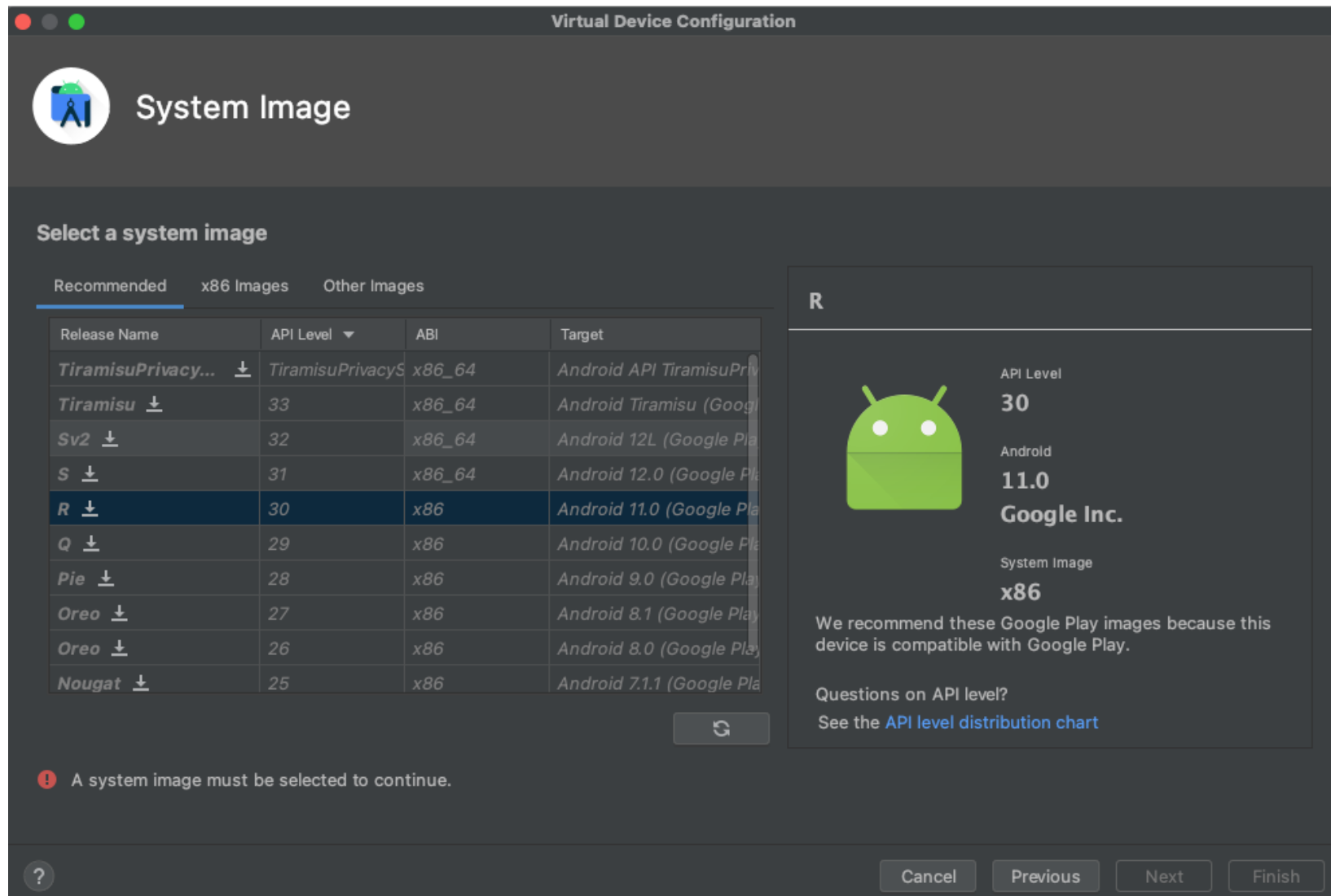
```
1  <?xml xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools">
3
4      <application
5          android:allowBackup="true"
6          android:dataExtractionRules="@xml/data_extraction_rules"
7          android:fullBackupContent="@xml/backup_rules"
8          android:icon="@mipmap/ic_launcher"
9          android:label="Conceptos básicos"
10         android:roundIcon="@mipmap/ic_launcher_round"
11         android:supportRtl="true"
12         android:theme="@style/Theme.ConceptosBásicos"
13         tools:targetApi="31">
14
15         <activity
16             android:name=".MainActivity"
17             android:exported="true">
18             <intent-filter>
19                 <action android:name="android.intent.action.MAIN" />
20
21                 <category android:name="android.intent.category.LAUNCHER" />
22             </intent-filter>
23
24             <meta-data
25                 android:name="android.app.lib_name"
26                 android:value="" />
27
28             </activity>
29         </application>
30     </xml>
```

Ejemplo de App


Creamos un emulador de un dispositivo en concreto.



Ejemplo de App




Ejemplo de App

**Android Virtual Device (AVD)**
Android Studio


Verify Configuration

AVD Name

 Nexus 5X

5.2 1080x1920 xxhdpi


Change...


 Lollipop

Android 5.0 armeabi-v7a

Change...

Startup orientation

 Portrait

 Landscape

Emulated Performance


Graphics:

Device Frame ☒ Enable Device Frame

Show Advanced Settings

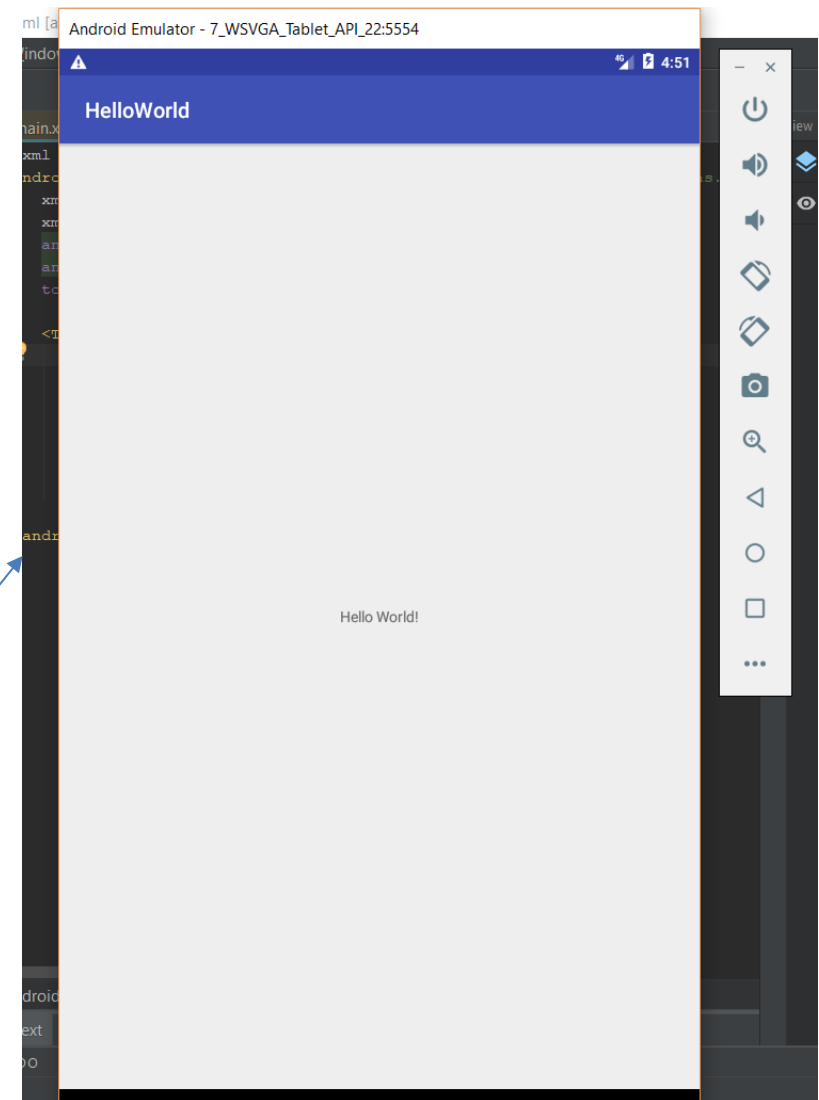
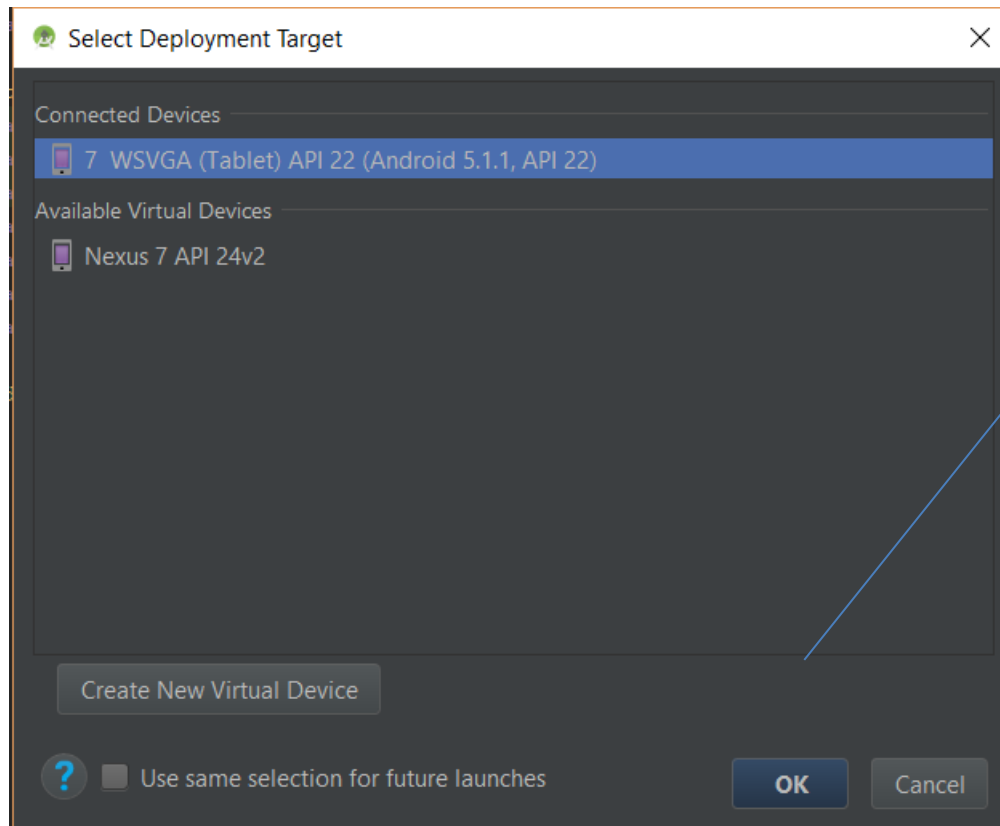
AVD Name
The name of this AVD.

Recommendation
Consider using an x86 system image on an x86 host for better emulation performance.



Ejemplo de App

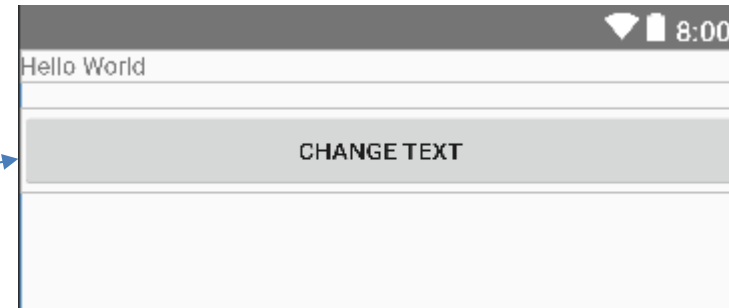
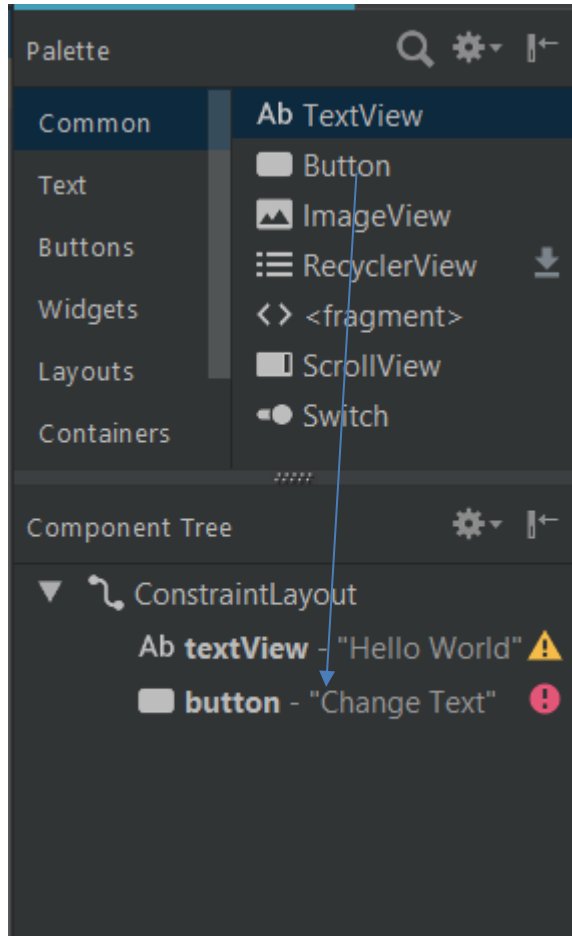
Una vez creado, podemos lanzar la app tanto en el emulador creado como en el móvil físico.



Ejemplo de App

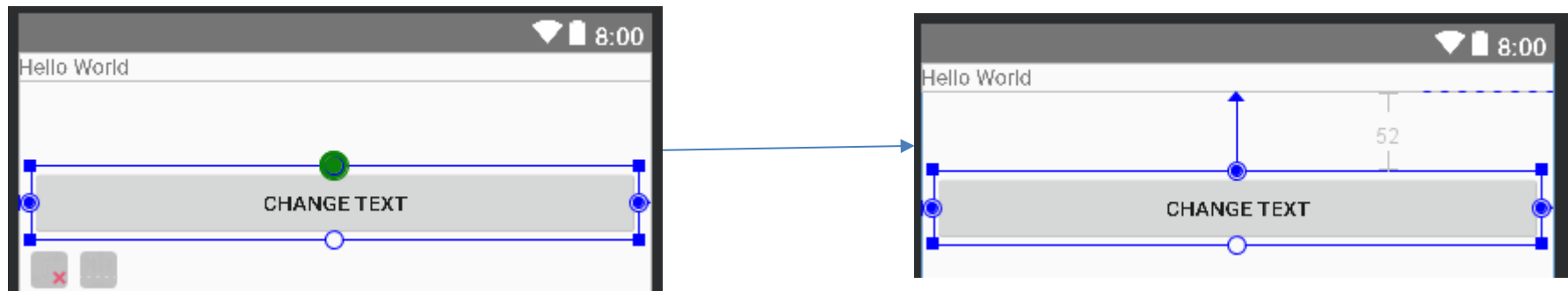
Vamos a añadir un botón que modifique el texto de "HelloWorld":

- Añadimos el botón



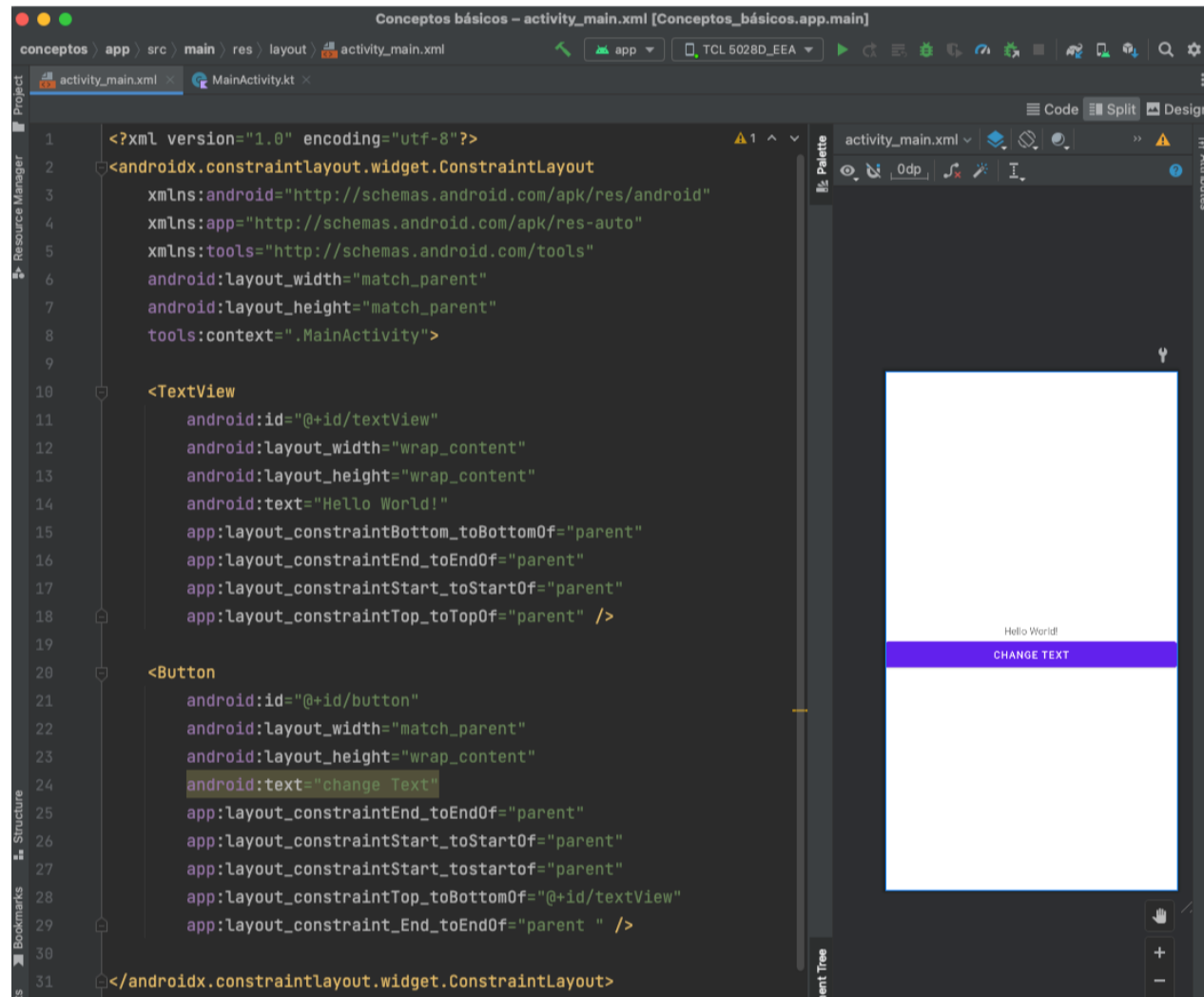
Ejemplo de App

Fijamos la posición del botón al Layout:



Ejemplo de App

El XML nos quedará parecido a este:



Ejemplo de App

Vemos el textview en detalle:

```
<TextView
```

```
    android:id="@+id/textView"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Hello World" />
```

Identificador del elemento

Cadena de texto

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"
```



```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"
```



```
android:layout_width="match_parent"  
android:layout_height="match_parent"
```



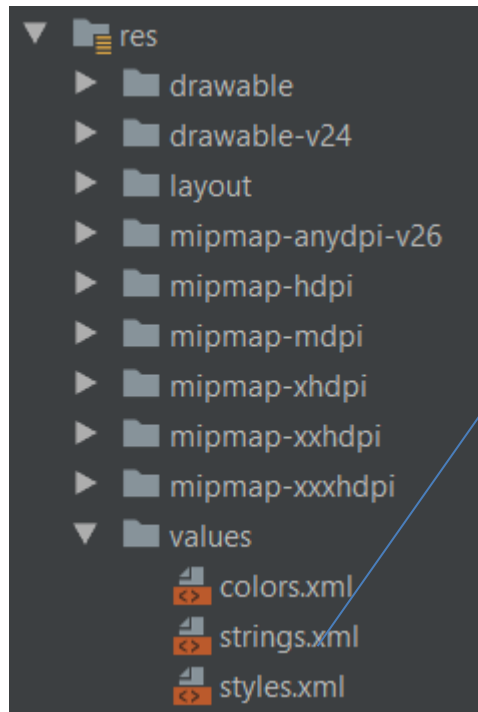
Ejemplo de App

Cadenas de texto:

```
<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Hello World" />
```

No es recomendable “hardcodear” los textos en los Layouts.

Cada cadena de texto se debe añadir al fichero “strings.xml”

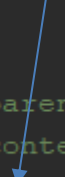


```
<resources>
    <string name="app_name">Test</string>
    <string name="hola_mundo">Hello World!</string>
</resources>
```

Ejemplo de App

Para poder utilizar el texto, hacemos referencia a su id en el Layout.

```
<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/hola_mundo" />
```



Ejemplo de App

Ahora vamos a darle funcionalidad al botón:

```
public class MainActivity extends AppCompatActivity {  
  
    private TextView m_text;  
    private Button m_button;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        m_text = findViewById(R.id.textView);  
        m_button = findViewById(R.id.button);  
  
        m_button.setOnClickListener(new View.OnClickListener() {  
            public void onClick(View v) {  
                m_text.setText("Adios");  
            }  
        });  
    }  
}
```

Referencias a los elementos del Layout

Asignamos las referencias buscando el id de los elementos.

R.id.nombreDelElemento

Ejemplo de App

Ahora vamos a darle funcionalidad al botón:

```
public class MainActivity extends AppCompatActivity {  
  
    private TextView m_text;  
    private Button m_button;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        m_text = findViewById(R.id.textView);  
        m_button = findViewById(R.id.button);  
  
        m_button.setOnClickListener(new View.OnClickListener() {  
            public void onClick(View v) {  
                m_text.setText("Adios");  
            }  
        });  
    }  
}
```

Escribimos el código que va a ejecutarse cuando se haga Click en el botón.

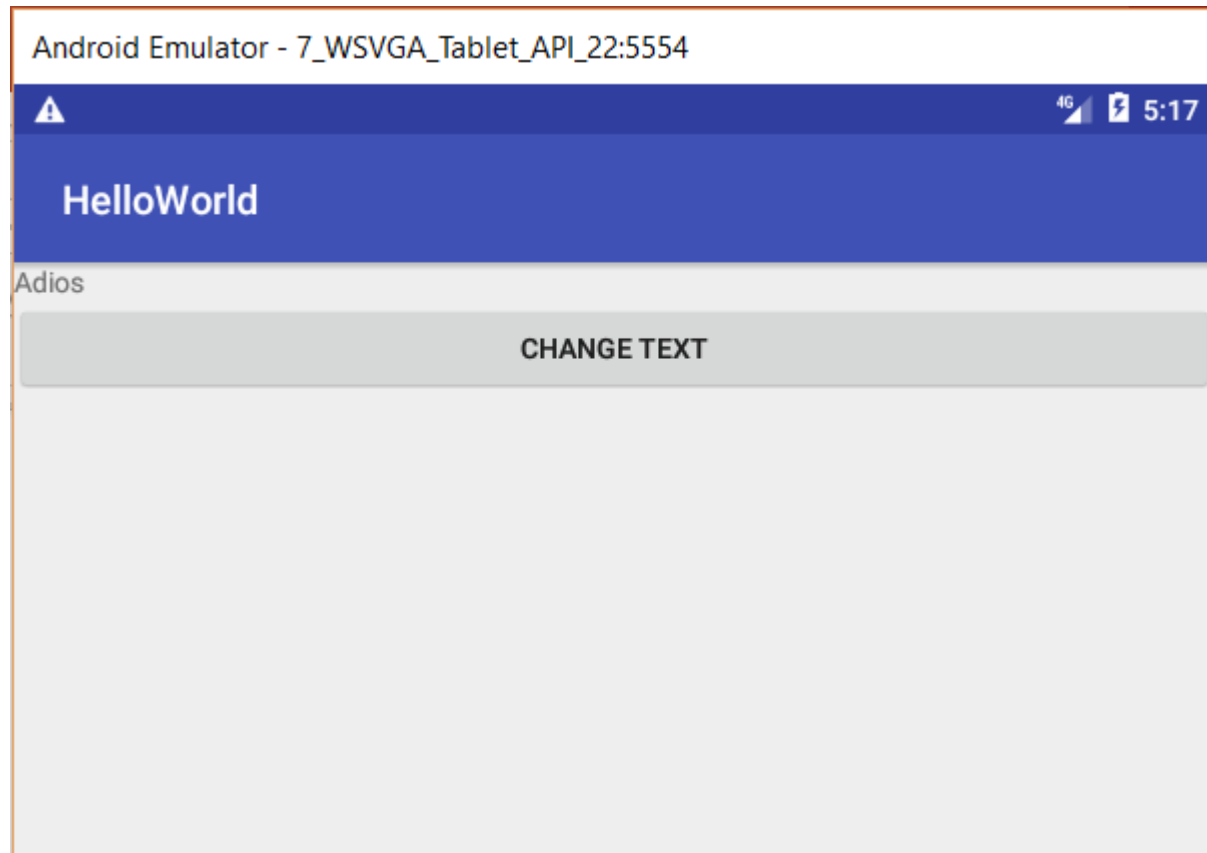
Esto se hace añadiendo un Listener al evento onClick con el método `setOnClickListener`.

Por parámetro pasamos un nuevo objeto de la clase `View.OnClickListener` con su método `onClick`.

Dentro de `onClick` añadimos el código de funcionamiento del botón.

Ejemplo de App

La app debería verse así:



Ejemplo de App

Ejercicio: Realizad una calculadora.