

# Tema 2

## Introducción a Android

# ÍNDICE

## Tema 2

---

1. Aplicaciones Android.
2. Nuevo proyecto Android.
3. Estructura proyecto Android.
4. ¿Cómo probar un proyecto Android?

# 1. Aplicaciones Android

Las aplicaciones están compuestas por diferentes componentes enlazados mediante un archivo llamado **AndroidManifest.xml** que los describe e indica cómo interactúan.

- **Actividades:** son la capa de presentación de la aplicación. Gestionan la interacción con el usuario. Heredan de la clase Activity.
- **Servicios:** se ejecutan en segundo plano de la aplicación. Se utilizan para llevar a cabo procesamiento que debe ser realizado de manera regular. Incluso en el caso de que nuestras actividades no estén visibles o activas.
- **Proveedores de contenido:** almacenan y comparten datos entre aplicaciones. Incluye proveedores de contenidos que permiten acceder a datos del dispositivo móvil como contactos, calendario, imágenes...

# 1. Aplicaciones Android

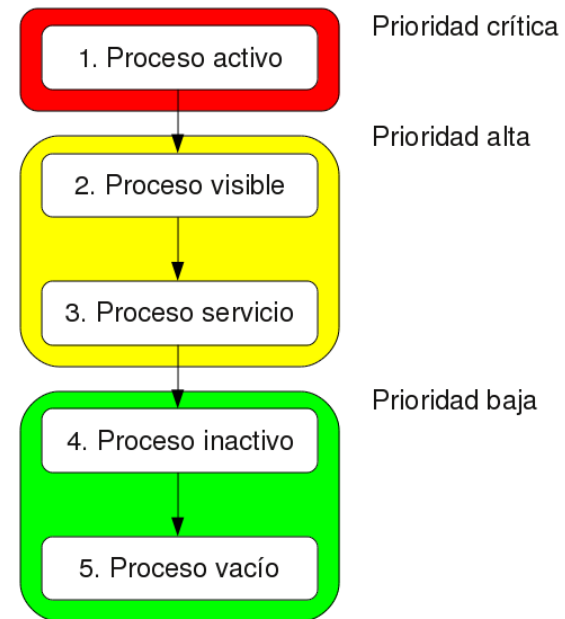
- **Intents:** se utilizan para la ejecución de acciones y permitir el paso de mensajes dentro de la misma aplicación o incluso entre distintas aplicaciones. Al emitir un Intent se declara la intención de que se lleve a cabo una determinada acción, por ejemplo. Cambiar de actividad, abrir la cámara, etc.
- **Receptores:** permiten a la aplicación hacerse cargo de determinadas acciones solicitadas mediante Intents. Se inician automáticamente cuando se lanza un Intent al que estaban escuchando.
- **Widgets:** componentes visuales.
- **Notificaciones:** permiten comunicarse con el usuario sin necesidad de robar el foco de la aplicación activa en ese momento. Por ejemplo, recibir un mensaje de texto, avisar al usuario mediante luces, sonidos, etc.

## 1.2. Ciclo de ejecución de una aplicación

Cada aplicación Android se ejecutará en su propio proceso.

El orden en el que los procesos son detenidos viene determinado por su prioridad, la cual se establece a partir del estado en el que se encuentre.

Ciclo de ejecución de los procesos en Android:



## 1.2. Ciclo de ejecución de una aplicación

- **Proceso activo:** aquellos que se encuentran interactuando con el usuario en ese preciso instante.
- **Proceso visible:** procesos visibles pero inactivos, ya sea porque su interfaz se está mostrando detrás de otras o porque no están respondiendo a ninguna entrada del usuario. Esto sucede cuando una actividad se encuentra parcialmente oculta por otra actividad, cuando aparece un diálogo por encima o cuando no ocupa toda la pantalla.
- **Proceso asociado a servicios en ejecución:** los servicios permiten que exista procesamiento sin necesidad de una interfaz de usuario.
- **Procesos inactivos:** procesos que albergan actividades que ni son visibles ni se encuentran realizando un procesamiento ni ejecutando un servicio.
- **Procesos vacíos:** son el resultado del intento de Android de retener aplicaciones en memoria una vez que están han terminado. Con esto se consigue que al lanzar de nuevo la aplicación de requiera menos tiempo.

# ÍNDICE

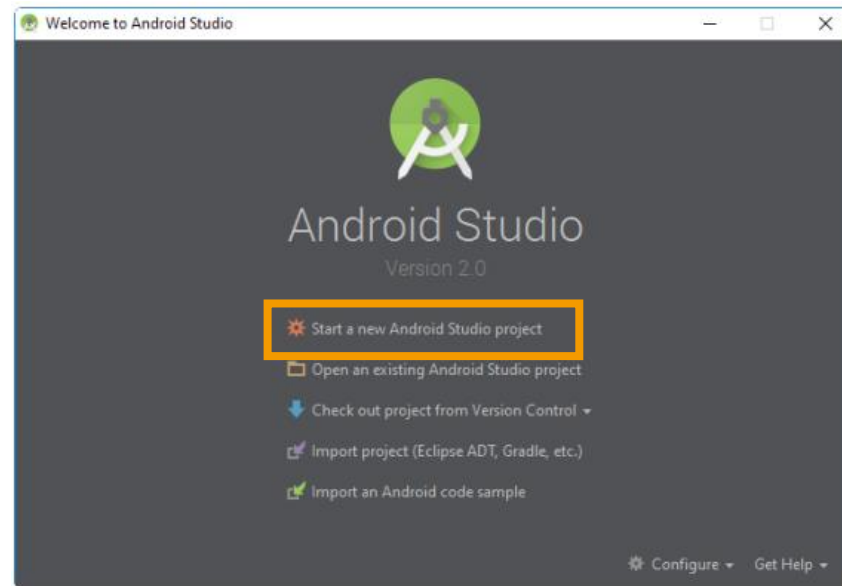
## Tema 2

---

1. Aplicaciones Android.
2. Nuevo proyecto Android.
3. Estructura proyecto Android.
4. ¿Cómo probar un proyecto Android?

## 2. Nuevo proyecto Android

Para crear un nuevo proyecto ejecutaremos Android Studio y desde la pantalla de bienvenida pulsaremos la opción "Start a new Android Studio project" para iniciar el asistente de creación de un nuevo proyecto:

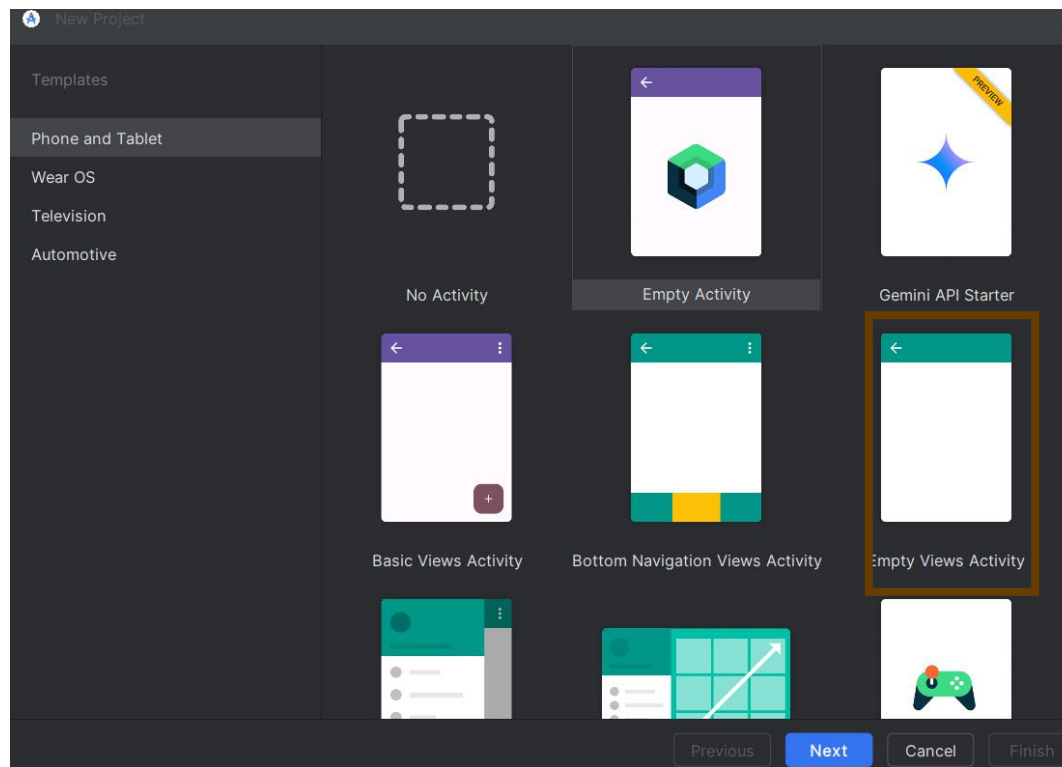


**Si ya habíamos abierto anteriormente Android Studio, accederemos al menú "File / New project..." -> Empty activity para crear el nuevo proyecto.**



## 2. Nuevo proyecto Android

Si ya habíamos abierto anteriormente Android Studio, accederemos al menú "File / New project..." -> Empty views activity para crear el nuevo proyecto.

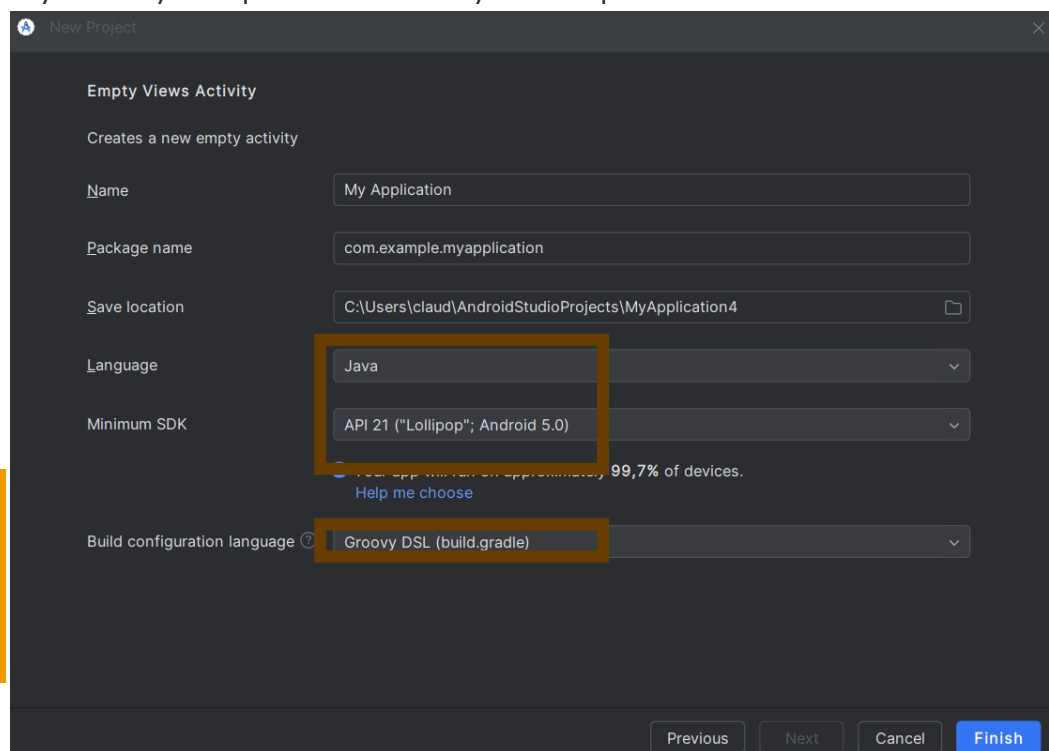


## 2. Nuevo proyecto Android

En la primera pantalla indicaremos, por este orden, el nombre de la aplicación, el dominio de la compañía, la ruta donde crear el proyecto y las plataformas y APIs que va a utilizar nuestra aplicación:

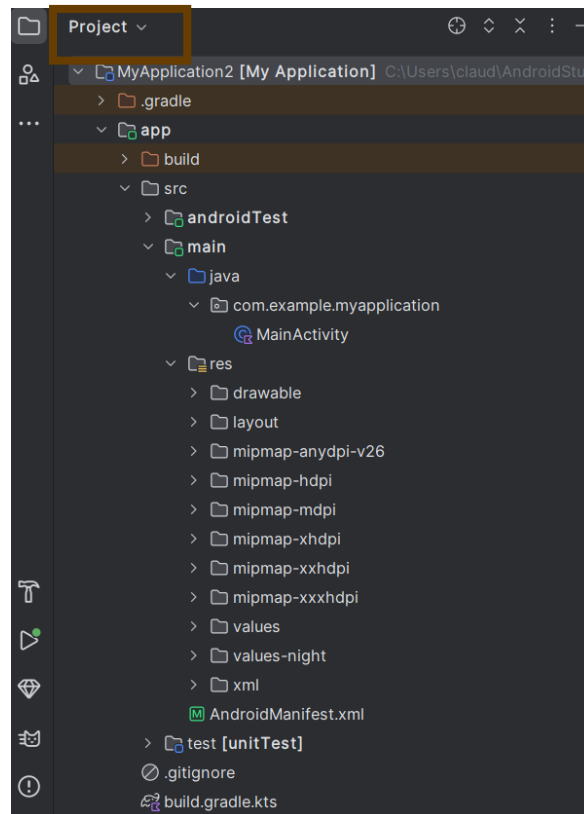
El enlace "**Help me choose**", que mostrará el porcentaje de dispositivos que ejecutan actualmente cada versión de Android.

Cuanto menor sea ésta, a más dispositivos podrá llegar nuestra aplicación, pero más complicado será conseguir que se ejecute correctamente en todas las versiones de Android



## 2. Nuevo proyecto Android

En la parte izquierda pulsaremos sobre la lista desplegable situada en la parte superior izquierda, y cambiaremos la vista de proyecto al modo "Project":



# ÍNDICE

## Tema 2

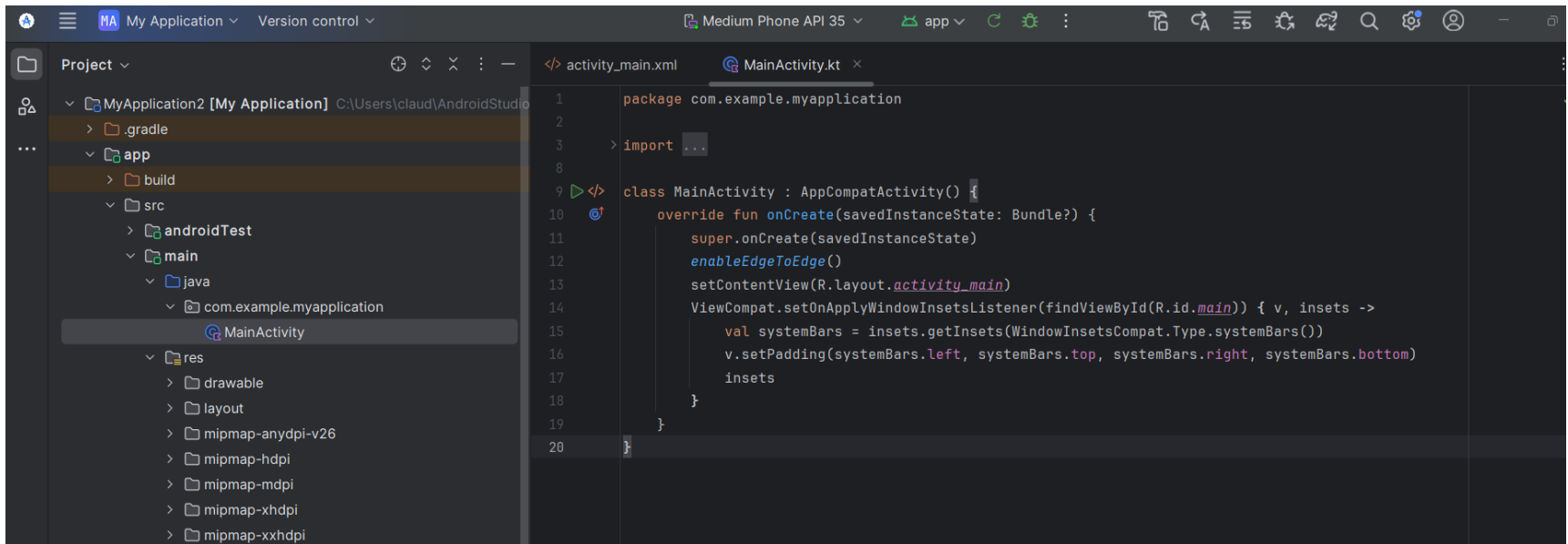
---

1. Aplicaciones Android.
2. Nuevo proyecto Android.
3. Estructura proyecto Android.
4. ¿Cómo probar un proyecto Android?

### 3. Estructura proyecto Android

#### Carpeta /app/src/main/java

Esta carpeta contendrá todo el código fuente de la aplicación, clases auxiliares, etc. Inicialmente, Android Studio creará por nosotros el código básico de la pantalla (actividad o activity) principal de la aplicación, que recordemos que en nuestro caso era **MainActivity**, y siempre bajo la estructura del paquete java definido durante la creación del proyecto.



### 3. Estructura proyecto Android

#### Carpeta `/app/src/main/res/`

Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, layouts, cadenas de texto, etc. Los diferentes tipos de recursos se pueden distribuir entre las siguientes subcarpetas:

Carpeta	Descripción
<code>/res/drawable/</code>	<p>Contiene las imágenes y otros elementos gráficos usados por la aplicación. Para poder definir diferentes recursos dependiendo de la resolución y densidad de la pantalla del dispositivo se suele dividir en varias subcarpetas:</p> <ul style="list-style-type: none"><li>• <code>/drawable</code> (recursos independientes de la densidad)</li><li>• <code>/drawable-ldpi</code> (densidad baja)</li><li>• <code>/drawable-mdpi</code> (densidad media)</li><li>• <code>/drawable-hdpi</code> (densidad alta)</li><li>• <code>/drawable-xhdpi</code> (densidad muy alta)</li><li>• <code>/drawable-xxhdpi</code> (densidad muy muy alta :)</li><li>• ...</li></ul>

### 3. Estructura proyecto Android

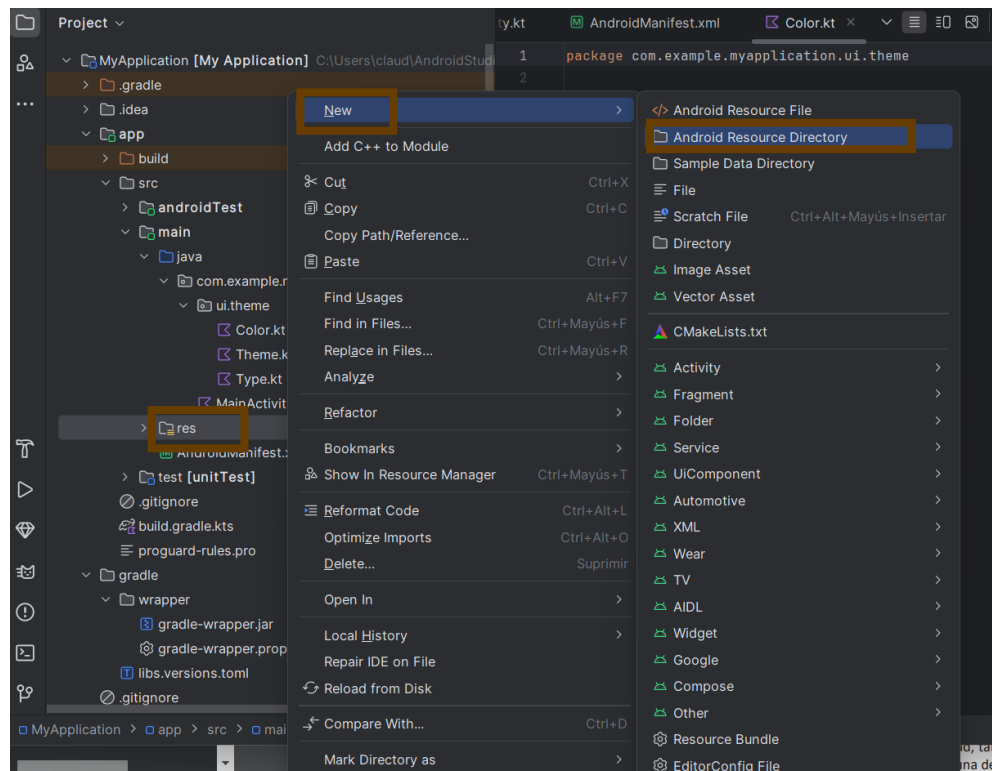
#### Carpeta `/app/src/main/res/`

No todas estas carpetas tienen por qué aparecer en cada proyecto Android, tan sólo las que se necesiten. Iremos viendo durante el curso qué tipo de elementos se pueden incluir en cada una de ellas y cómo se utilizan.

Carpeta	Descripción
<code>/res/mipmap/</code>	Contiene los iconos de lanzamiento de la aplicación (el icono que aparecerá en el menú de aplicaciones del dispositivo) para las distintas densidades de pantalla existentes. Al igual que en el caso de las carpetas <code>/drawable</code> , se dividirá en varias subcarpetas dependiendo de la densidad de pantalla: <ul style="list-style-type: none"><li>• <code>/mipmap-mdpi</code></li><li>• <code>/mipmap-hdpi</code></li><li>• <code>/mipmap-xhdpi</code></li><li>• ...</li></ul>
<code>/res/layout/</code>	Contiene los ficheros de definición XML de las diferentes pantallas de la interfaz gráfica. Para definir distintos layouts dependiendo de la orientación del dispositivo se puede dividir también en subcarpetas: <ul style="list-style-type: none"><li>• <code>/layout</code> (vertical)</li><li>• <code>/layout-land</code> (horizontal)</li></ul>
<code>/res/anim/</code> <code>/res/animator/</code>	Contienen la definición de las animaciones utilizadas por la aplicación.
<code>/res/color/</code>	Contiene ficheros XML de definición de colores según estado.
<code>/res/menu/</code>	Contiene la definición XML de los menús de la aplicación.
<code>/res/xml/</code>	Contiene otros ficheros XML de datos utilizados por la aplicación.
<code>/res/raw/</code>	Contiene recursos adicionales, normalmente en formato distinto a XML, que no se incluyan en el resto de carpetas de recursos.
<code>/res/values/</code>	Contiene otros ficheros XML de recursos de la aplicación, como por ejemplo cadenas de texto ( <code>strings.xml</code> ), estilos ( <code>styles.xml</code> ), colores ( <code>colors.xml</code> ), arrays de valores ( <code>arrays.xml</code> ), tamaños ( <code>dimens.xml</code> ), etc.

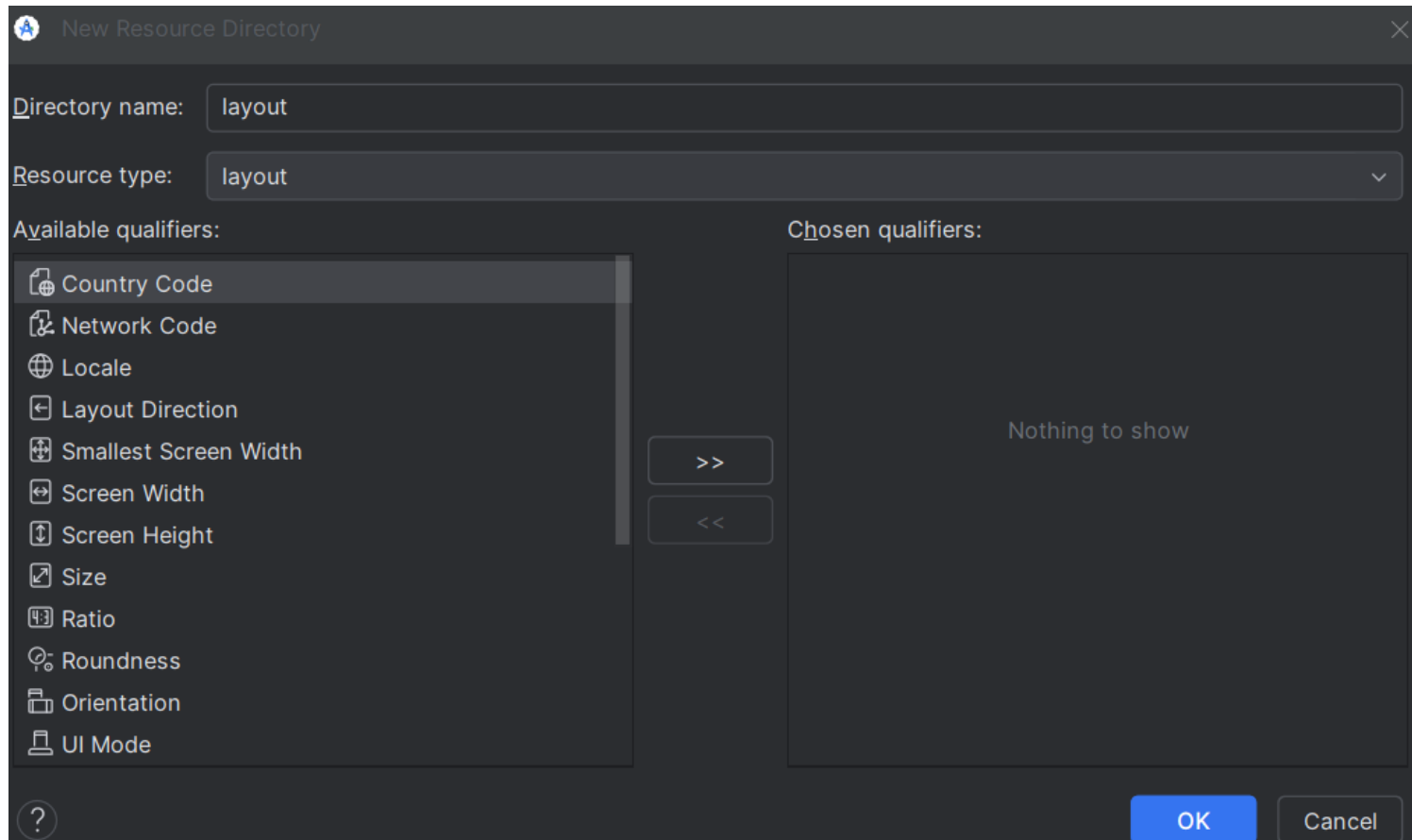
## 3.1. Fix activity\_main.xml

En ocasiones no aparece la carpeta `/res/layout/`: contiene los ficheros de definición XML de las diferentes pantallas de la interfaz gráfica

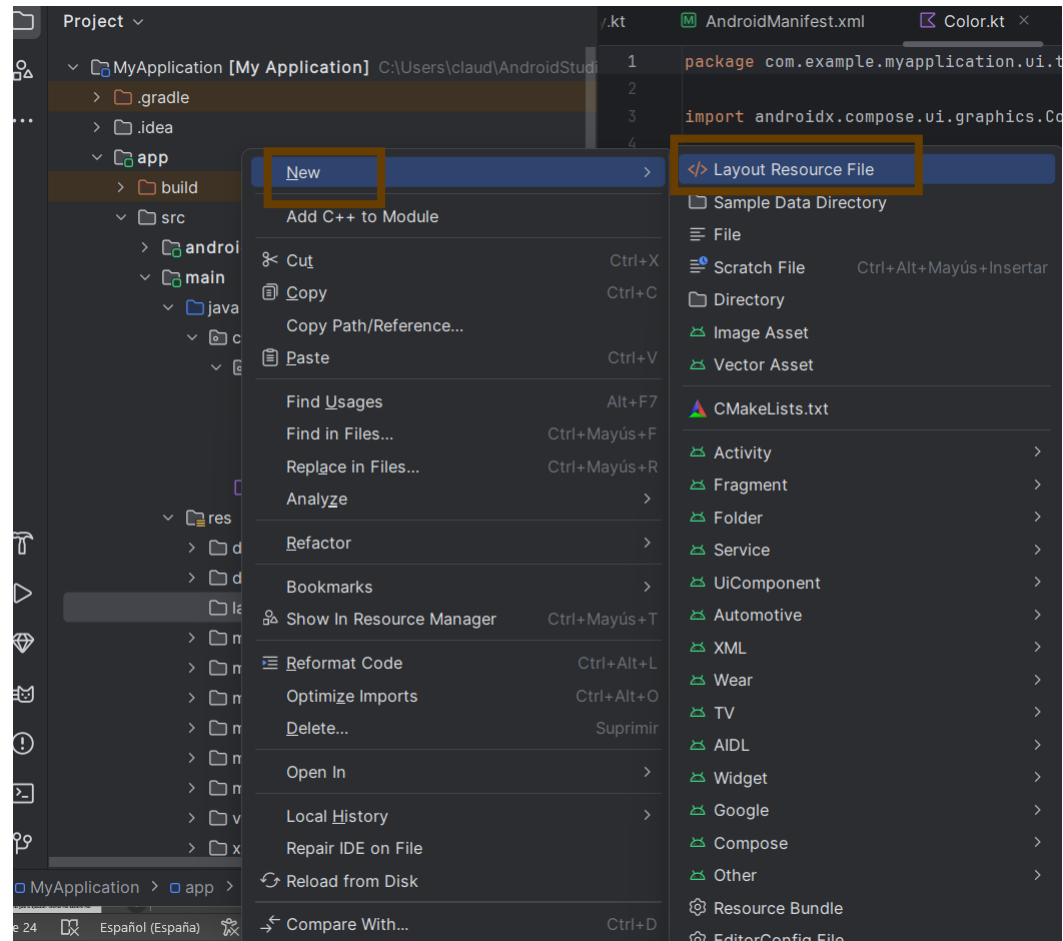




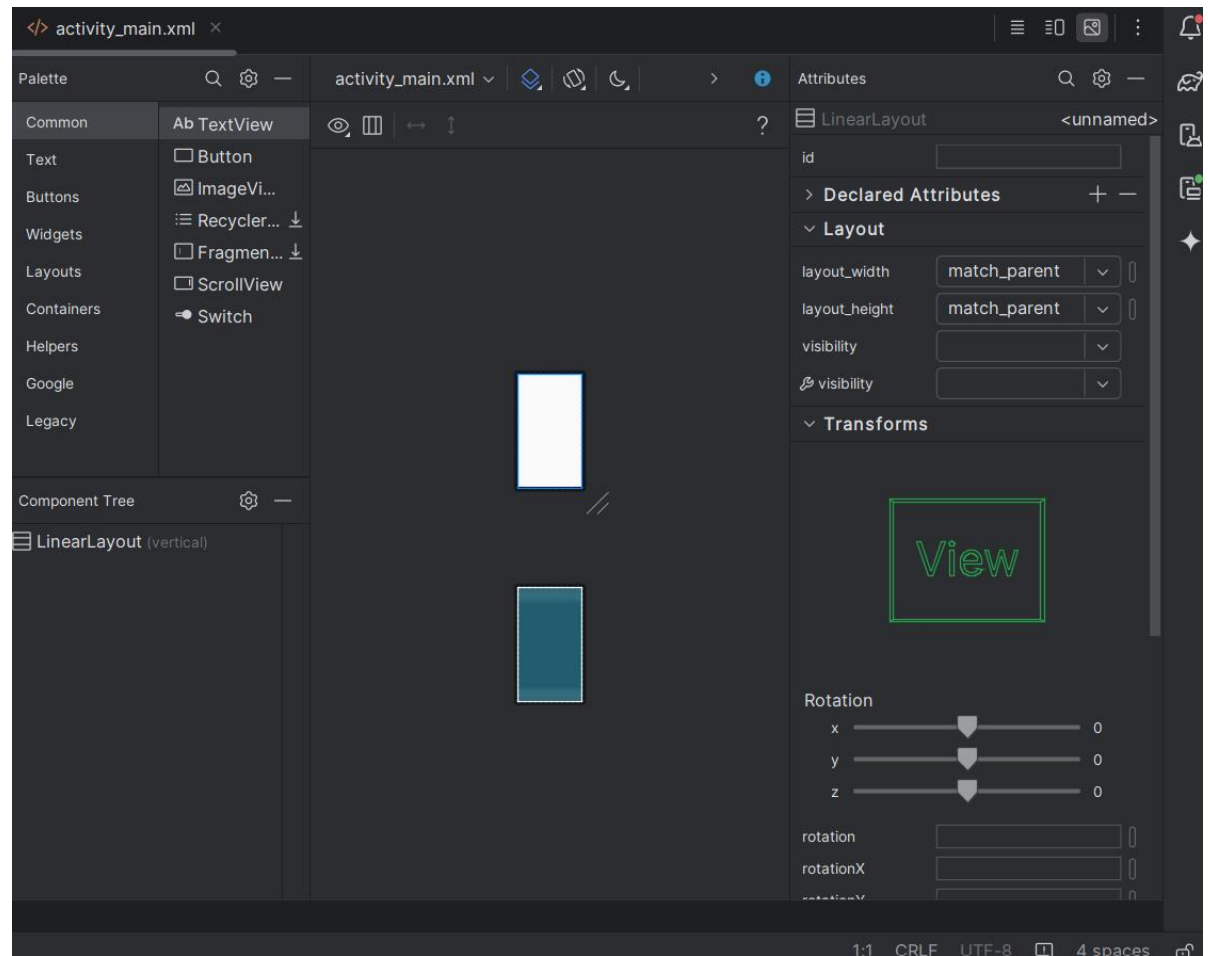
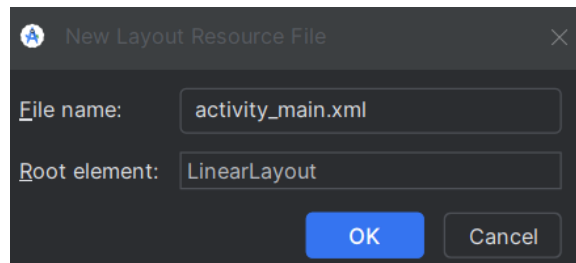
### 3.1. Fix activity\_main.xml



## 3.1. Fix activity\_main.xml



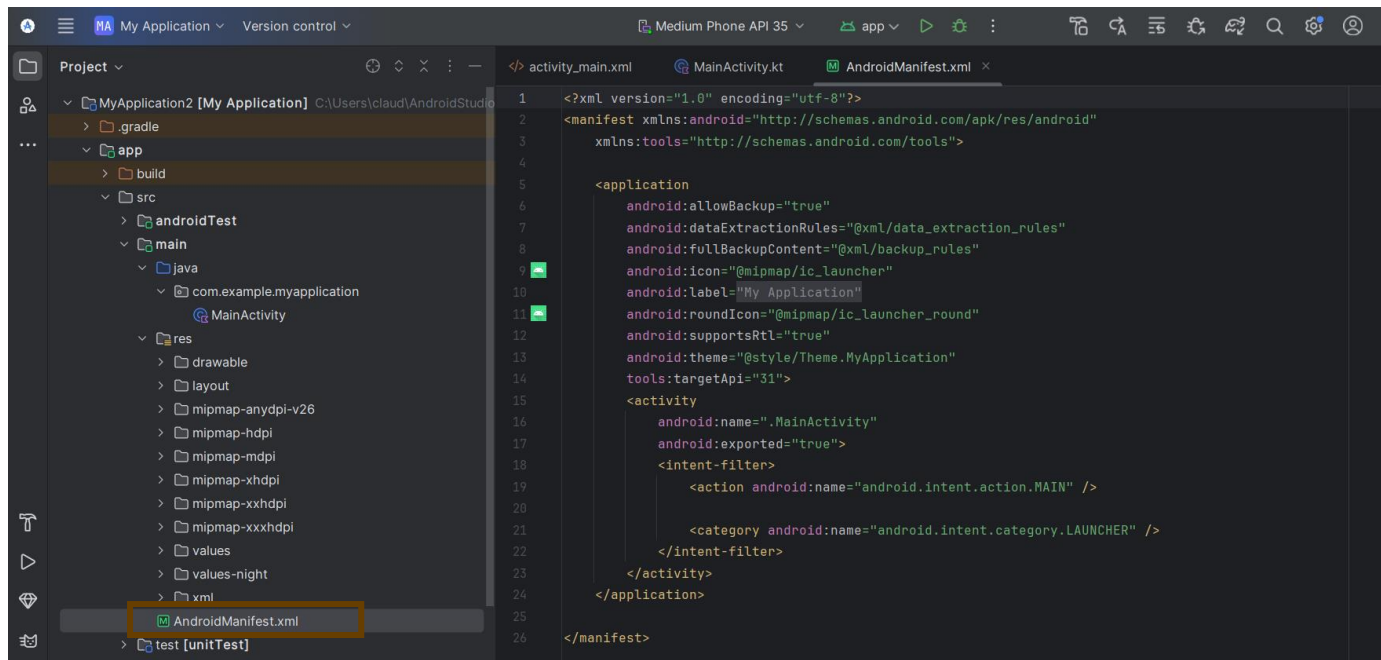
## 3.1. Fix activity\_main.xml



### 3. Estructura proyecto Android

#### Fichero /app/src/main/AndroidManifest.xml

Contiene la definición en XML de muchos de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, icono, ...), sus componentes (pantallas, servicios, ...), o los permisos necesarios para su ejecución. Veremos más adelante más detalles de este fichero.



### 3. Estructura proyecto Android

#### Gradle

Es el sistema de compilación usado por Android Studio, se trata de un plugin a parte del propio Android Studio para poder facilitar su actualización y exportación entre proyectos.

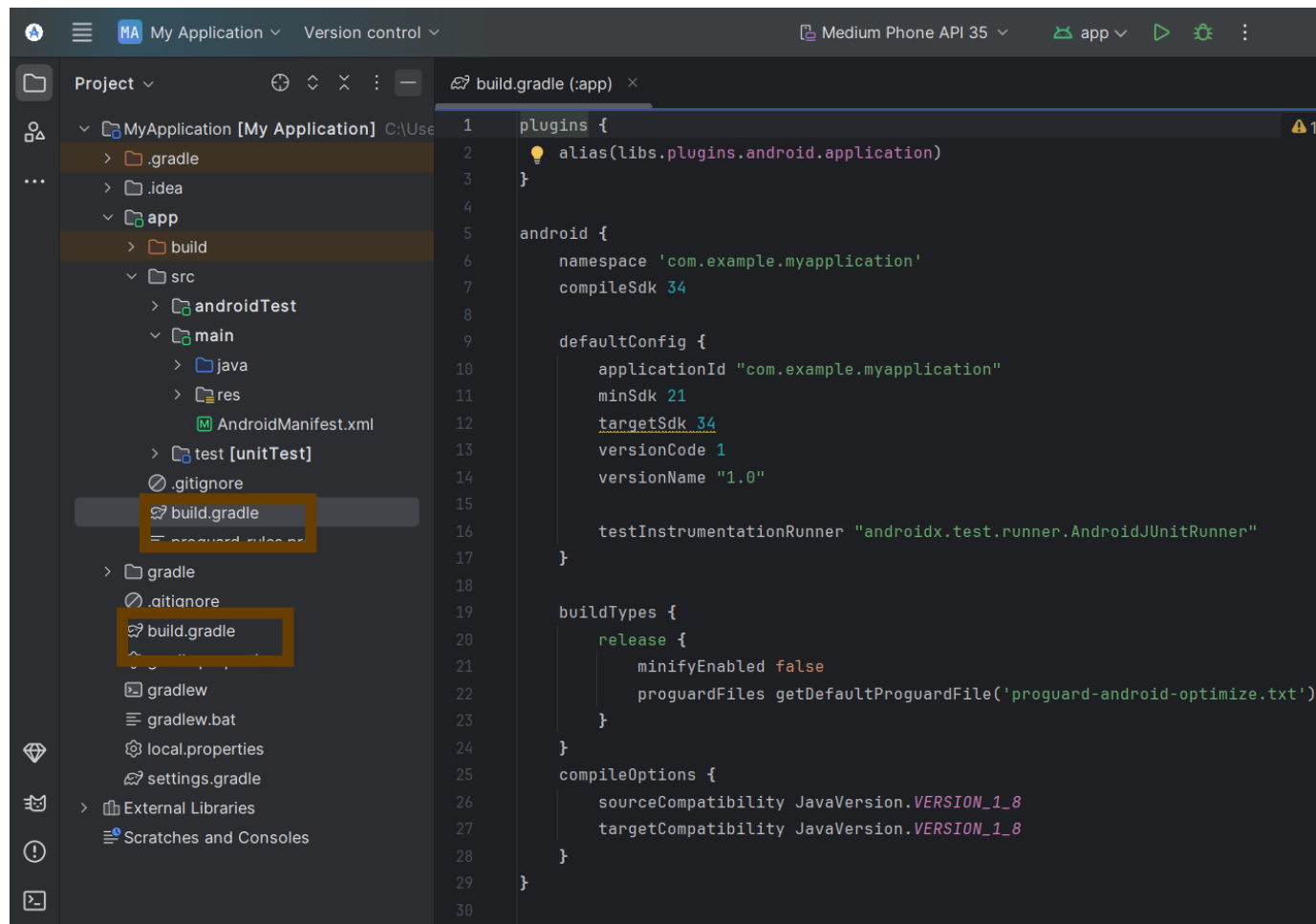
Se dispone de un archivo de proyecto y uno de aplicación.

El de proyecto, contiene información necesaria para la compilación del proyecto, por ejemplo, la versión del SDK de Android utilizada para compilar, la mínima versión de Android que soportará la aplicación, referencias a las librerías externas utilizadas, etc. Más adelante veremos también más detalles de este fichero.

El de aplicación, dentro de la “etiqueta” Android se puede ver toda la información relacionada con la versión objetivo de la aplicación, versión mínima requerida, número de versión e ID de la aplicación. Y dentro de la “etiqueta” dependencias se encuentra, aunque suene redundante, las dependencias de la aplicación es decir todo aquel código que no es propietario (normalmente no pertenece al desarrollador) pero que sin embargo es necesario para ciertas funciones de la app (las llamadas libs).

### 3. Estructura proyecto Android

#### Gradle



### 3. Estructura proyecto Android

#### **Carpeta /app/libs**

Puede contener las librerías java externas (ficheros .jar) que utilice nuestra aplicación. Normalmente no incluiremos directamente aquí ninguna librería, sino que haremos referencia a ellas en el fichero build.gradle.

#### **Carpeta /app/build/**

Contiene una serie de elementos de código generados automáticamente al compilar el proyecto. Dado que estos ficheros se generan automáticamente tras cada compilación del proyecto es importante que no se modifiquen manualmente bajo ninguna circunstancia.

# ACTIVIDAD

Mejorar rendimiento emulador Android



# ÍNDICE

## Tema 2

---

1. Aplicaciones Android.
2. Nuevo proyecto Android.
3. Estructura proyecto Android.
4. ¿Cómo probar un proyecto Android?

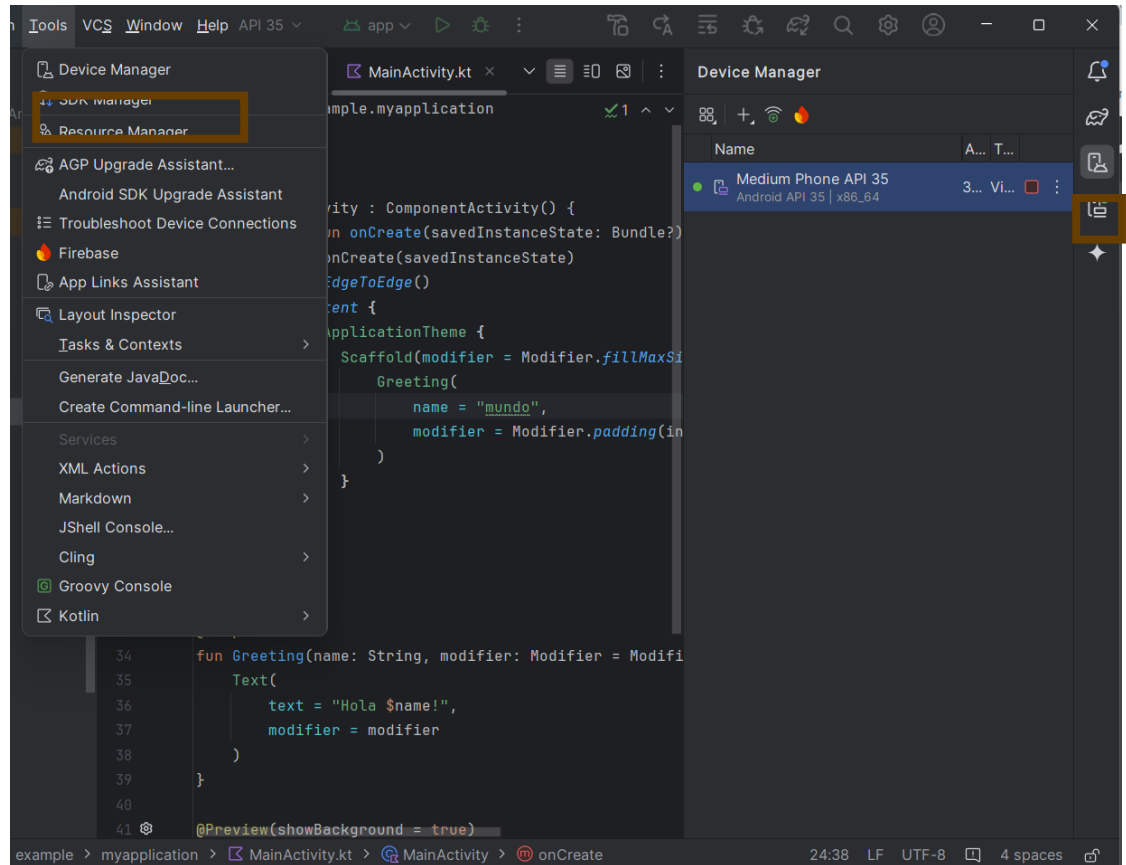
## 4. ¿Cómo probar una aplicación Android?

Para probar nuestras aplicaciones disponemos de dos formas básicas:

- A través de un terminal móvil Android que deberemos conectar a nuestro Pc.
- **A través de un emulador dentro del propio Android Studio muy fácil de manejar desde su gestor (AVD Manager) que dispone de dispositivos preconfigurados además de permitirnos realizar configuraciones a medida.**

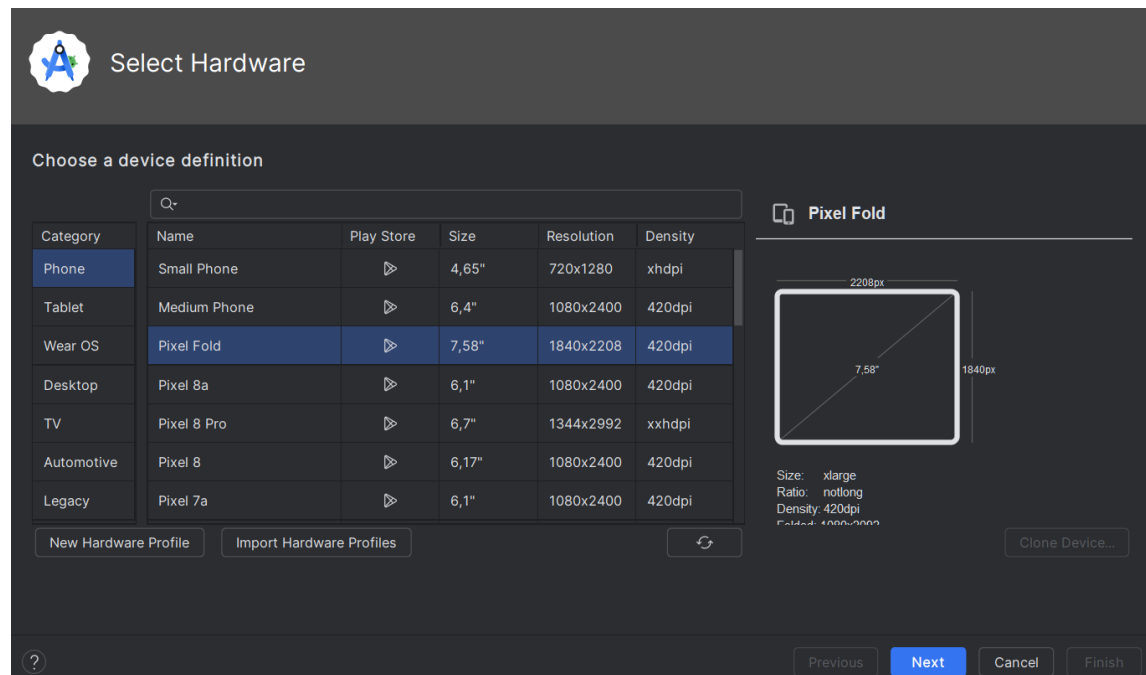
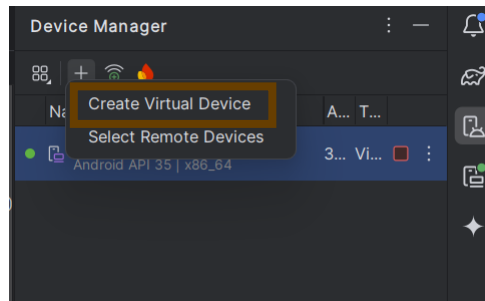
## 4.1. Emulador Android Studio

Tools -> Device Manager

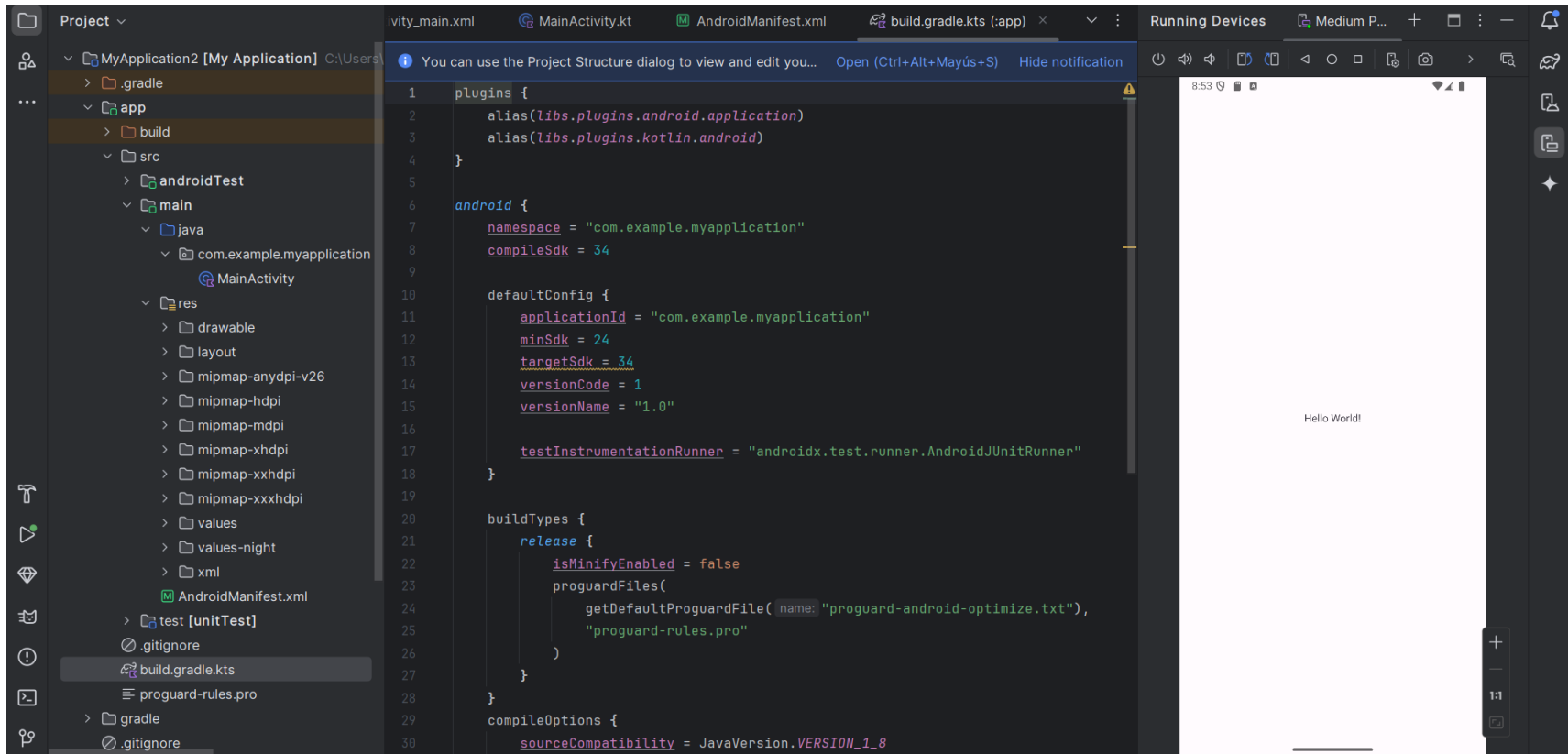


## 4.1. Emulador Android Studio

Device Manager -> Create virtual device

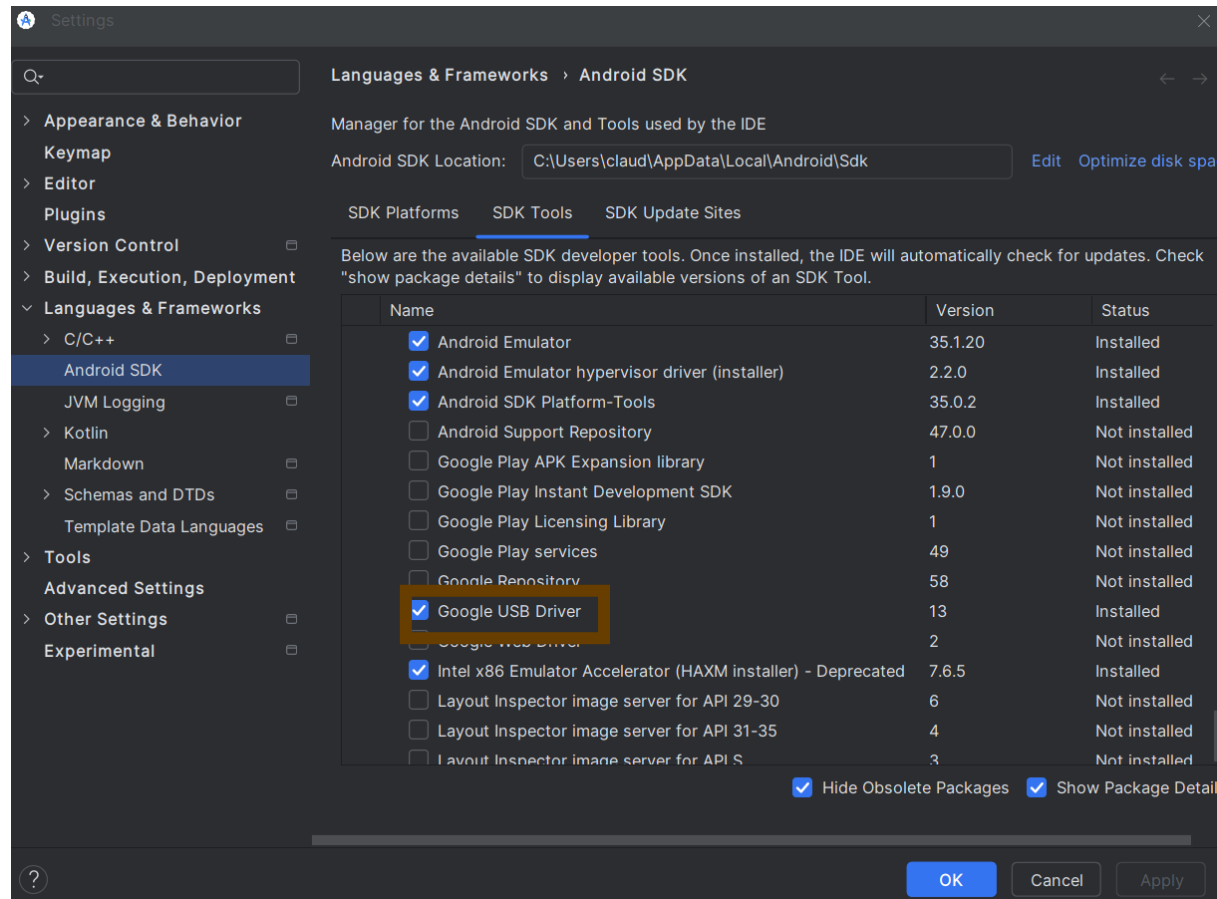


## 4.1. Emulador Android Studio



## 4.2. Conectar dispositivo Android

Tools -> SDK Manager -> SDK Tools



## 4.2. Conectar dispositivo Android

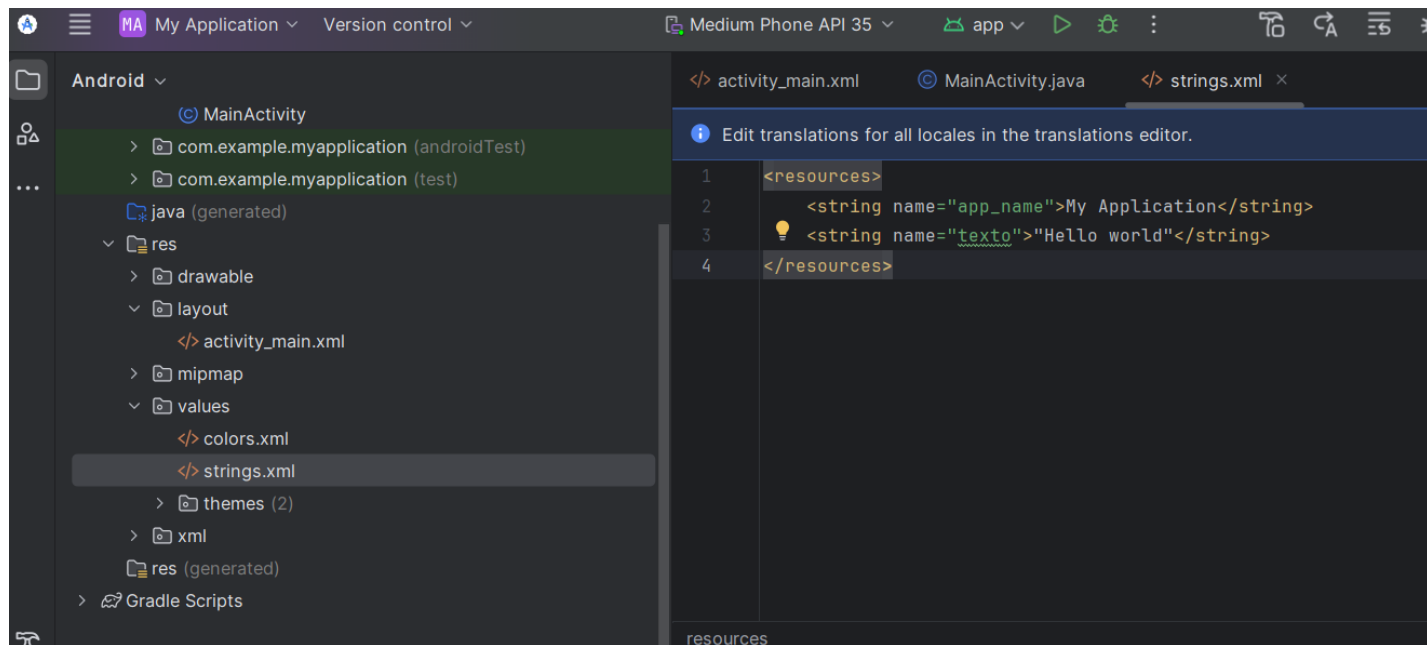
Activar la Depuración USB en nuestro teléfono Android de prueba.

1. Activar las opciones de desarrollador:
  1. Ajustes -> Acerca del teléfono -> Número de compilación (Build number)
  2. Pulsar 7 veces sobre este número
2. Cerramos Ajustes y volvemos a abrirlo
  1. Ajustes -> Opciones avanzadas -> Opciones de desarrollador
  2. Activamos la opción Depuración USB
3. Conectamos el teléfono Android al ordenador vía USB y aparecerá el siguiente diálogo en el dispositivo



## 5. Internacionalizar aplicación Android

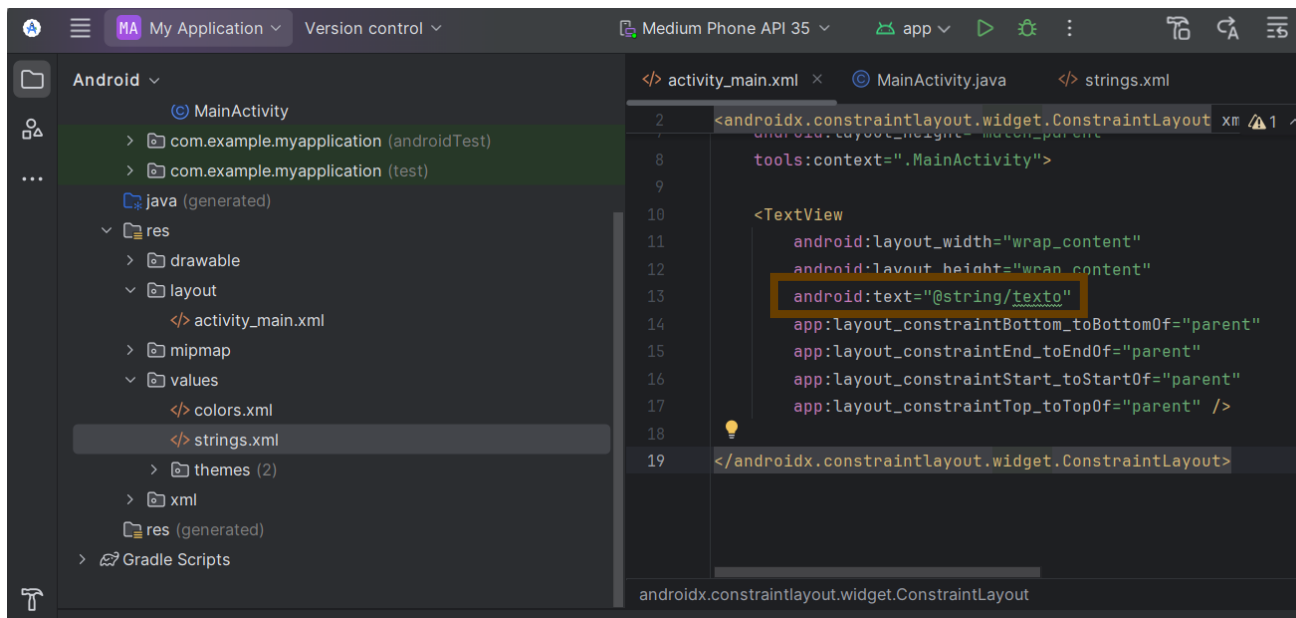
Definir una nueva cadena de texto en el fichero de recursos  
**/src/main/res/values/strings.xml**:





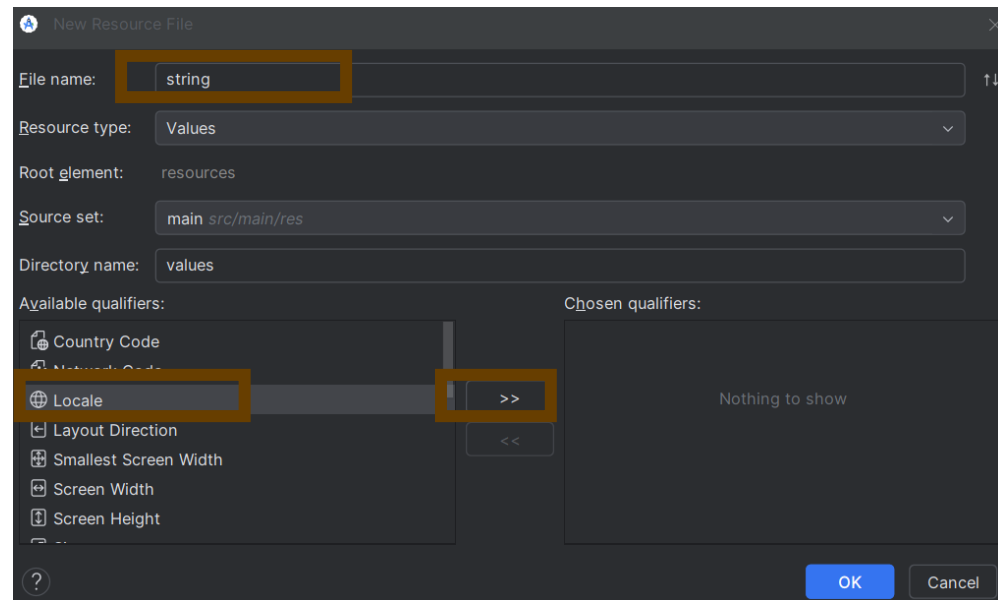
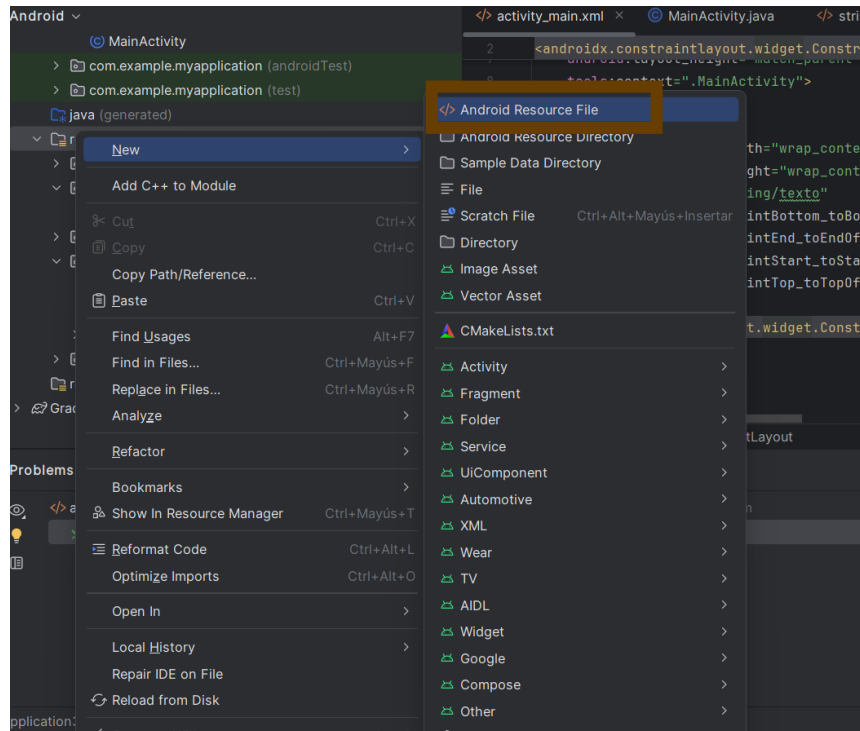
## 5. Internacionalizar aplicación Android

En el archivo `activity_main.xml`, indicar el identificador de la cadena como valor de la propiedad **`android:text`**, siempre precedido del prefijo **`"@string/"`**, de la siguiente forma

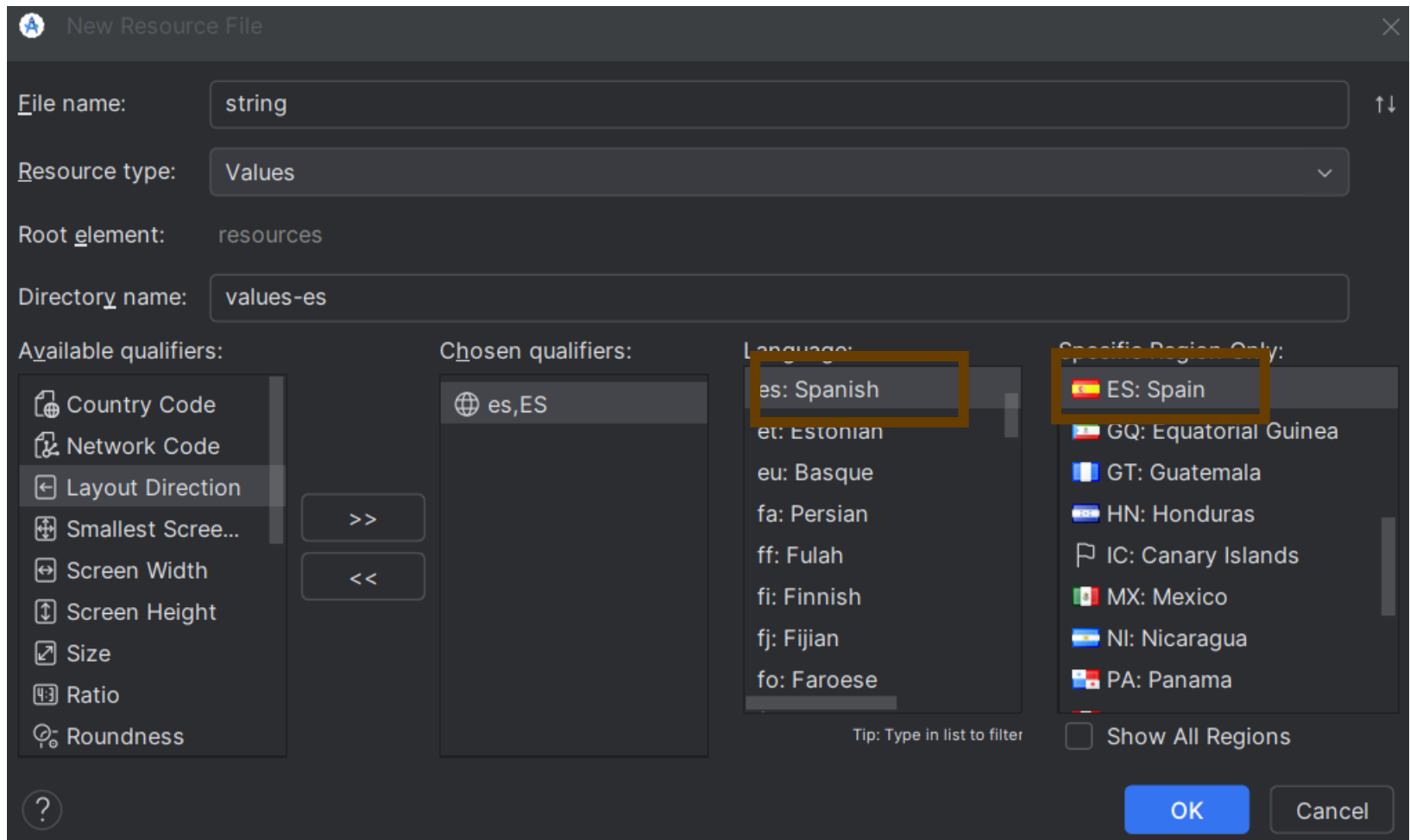


## 5. Internacionalizar aplicación Android

Añadimos un nuevo archivo de recurso. Sobre la carpeta res del proyecto, hacemos clic en botón derecho **New -> Android Resource File**

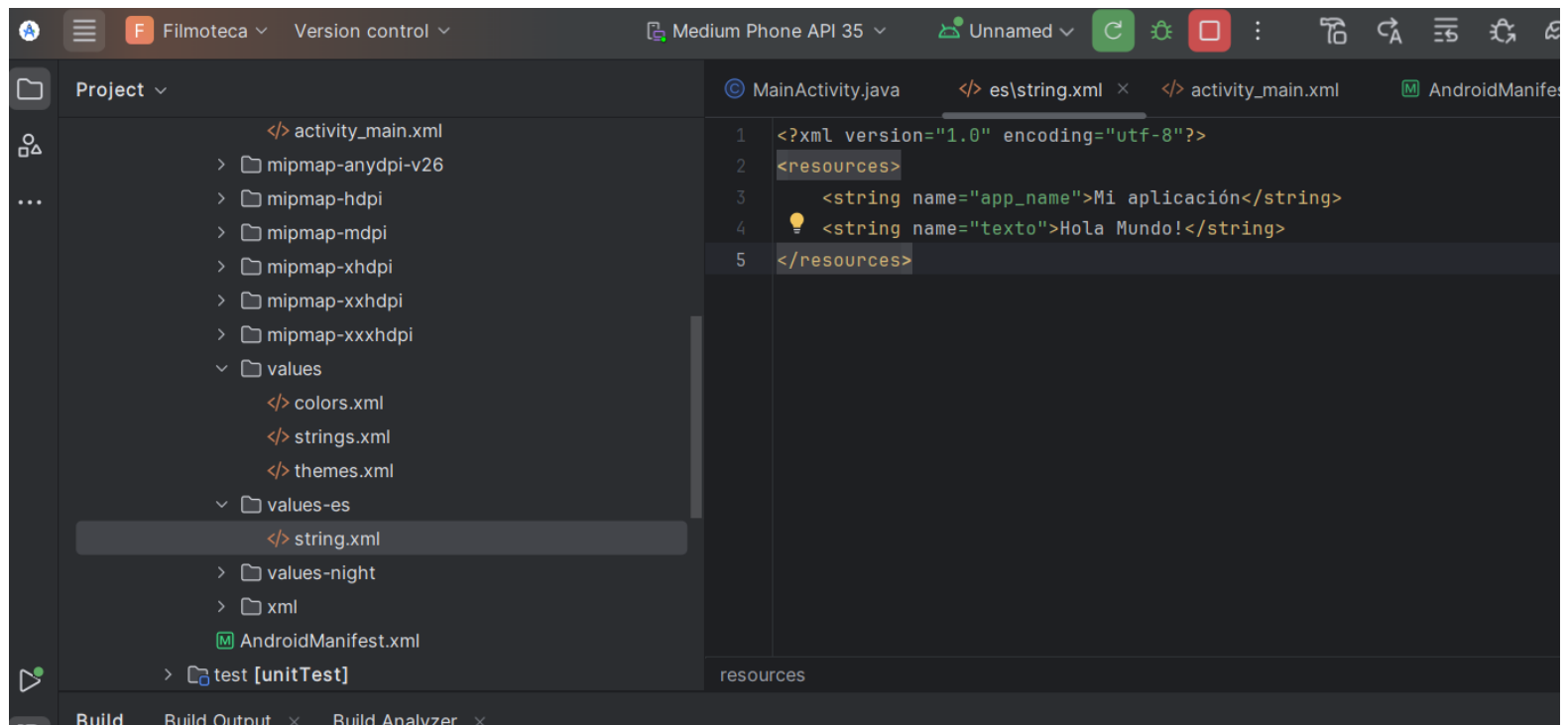


## 5. Internacionalizar aplicación Android



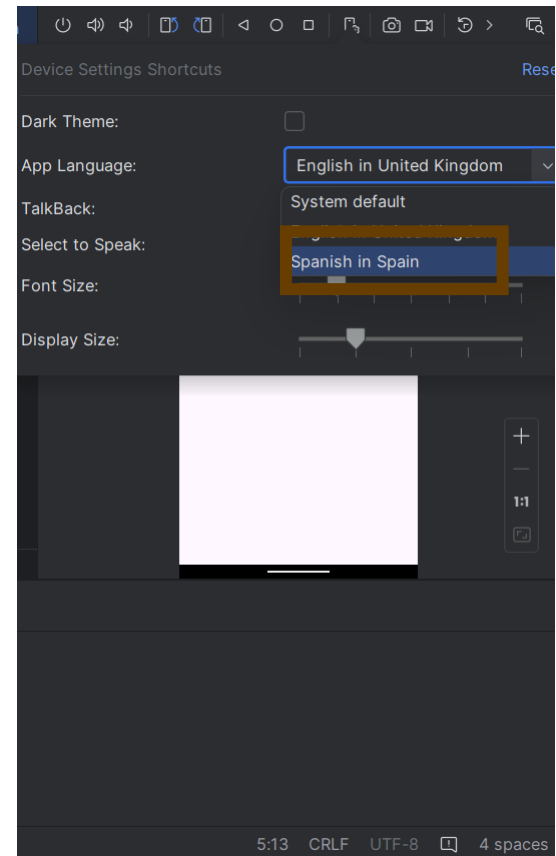
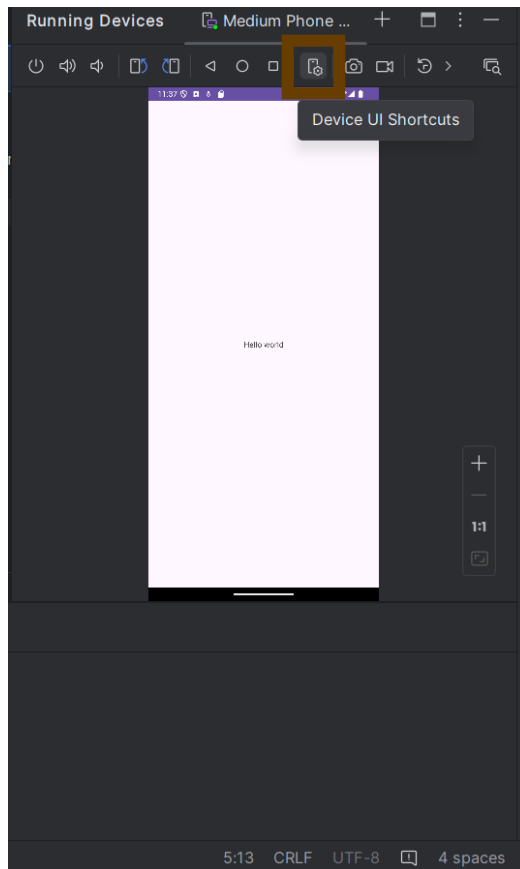
## 5. Internacionalizar aplicación Android

Copiamos el contenido del archivo string.xml por defecto al nuevo archivo string.xml (ES-rES) y cambiamos el contenido de los string al español



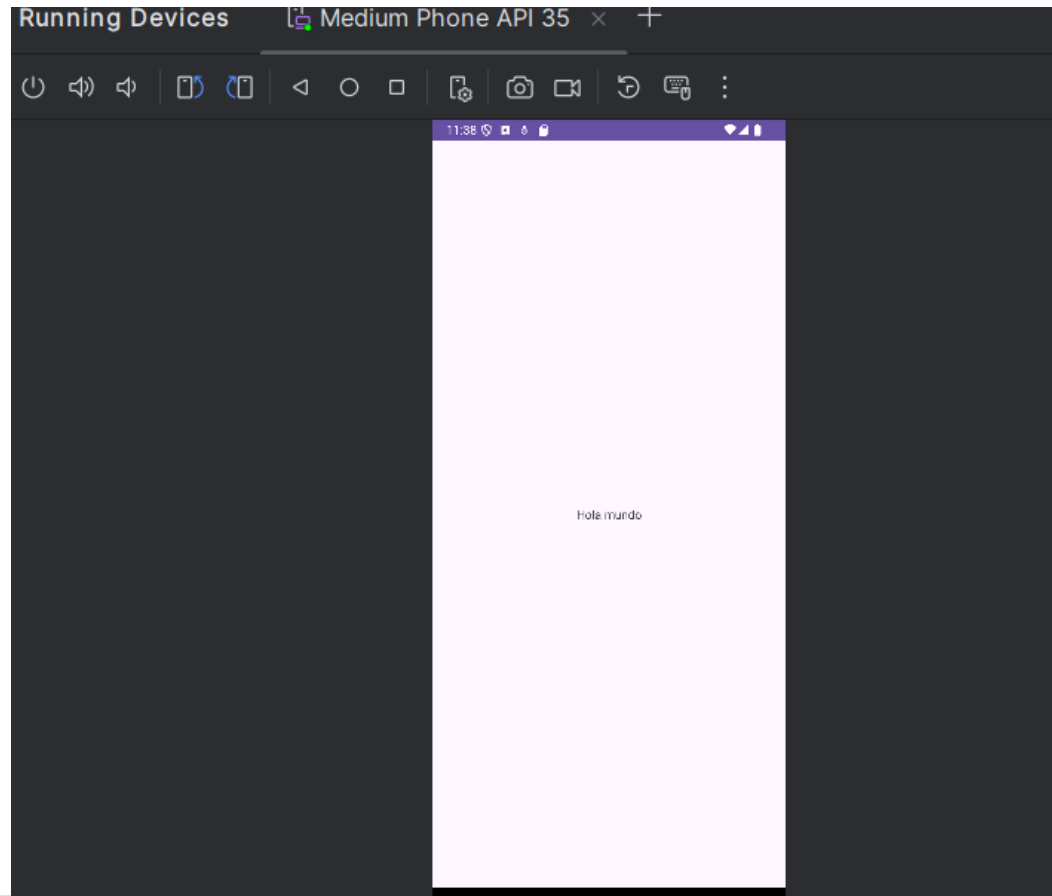
## 5. Internacionalizar aplicación Android

Ejecutamos la app en el emulador y cambiamos la configuración de idioma del dispositivo



## 5. Internacionalizar aplicación Android

Comprobamos que el contenido de la aplicación cambia a español.



# ACTIVIDAD

Añadir un nuevo archivo para idioma inglés

# PROYECTO

## CREACIÓN DEL PROYECTO

Nuevo proyecto Android cuyo nombre será **Filmoteca** y deberá ser compatible con dispositivos Android a partir de Android 5.0+ (lollipop)

Deberá incluir una primera **Actividad** a partir de una plantilla vacía y se llamará **MainActivity**.

Haremos que se genere automáticamente su **layout activity\_main.xml**

Añadiremos los siguientes **componentes gráficos** en el layout:

- Un componente de tipo **TextView** con el texto "**Creado por**" seguido de nuestro nombre.
- Un componente de tipo **ImageView**. Añadir una imagen (formato PNG) como recurso de tipo drawable, al directorio **res/drawable**. Haz que el componente ImageView muestre la imagen.



# PROYECTO

## CREACIÓN DEL PROYECTO

Añadiremos los siguientes **componentes gráficos** en el layout:

- Un botón **Button** con el texto “**Ir al sitio web**”
- Un botón **Button** con el texto “**Obtener soporte**”
- Un botón **Button** con el texto “**Volver**”

# PROYECTO

## Paso 1: Añadir la imagen en formato PNG al directorio res/drawable

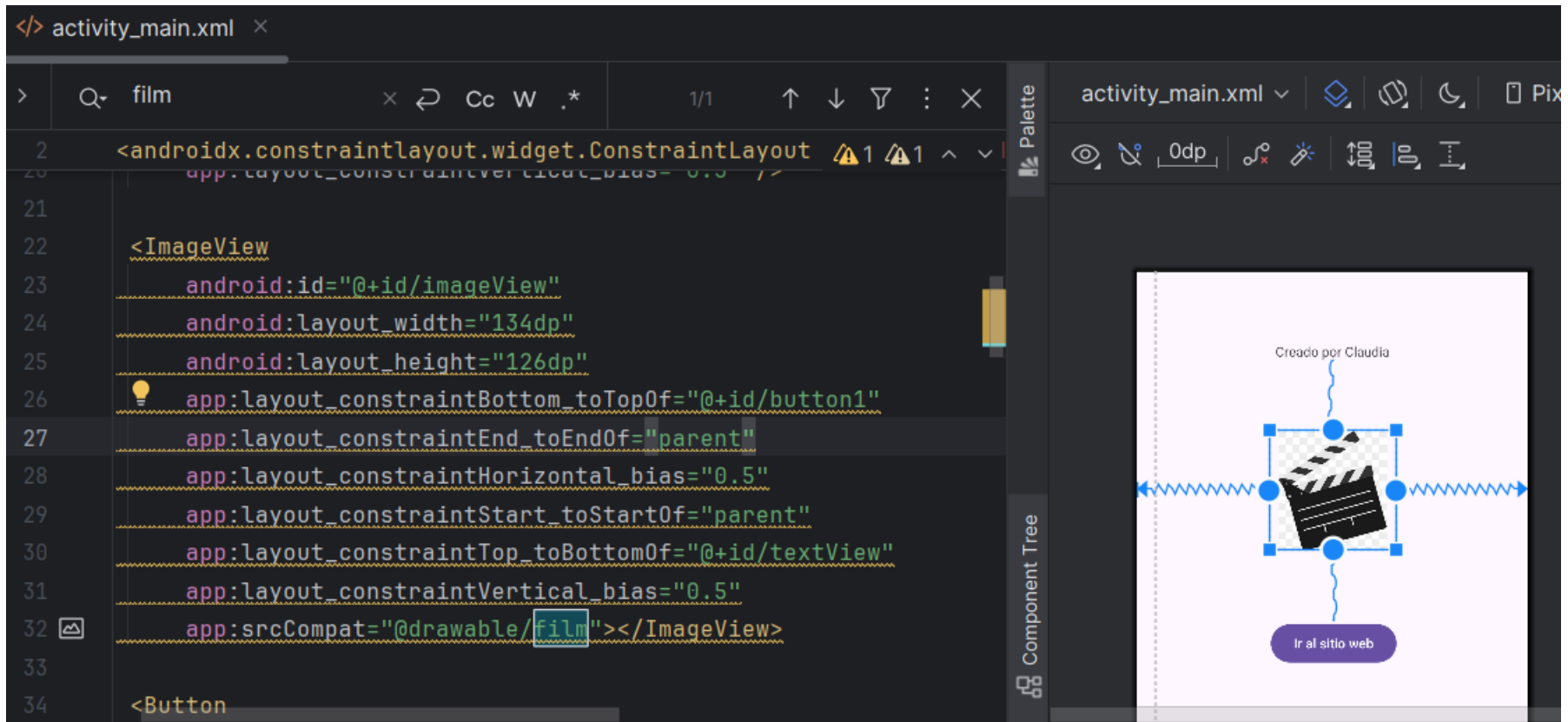
1. **Descarga o localiza la imagen en formato PNG** que desees usar en tu proyecto.

2. **Agrega la imagen al proyecto:**

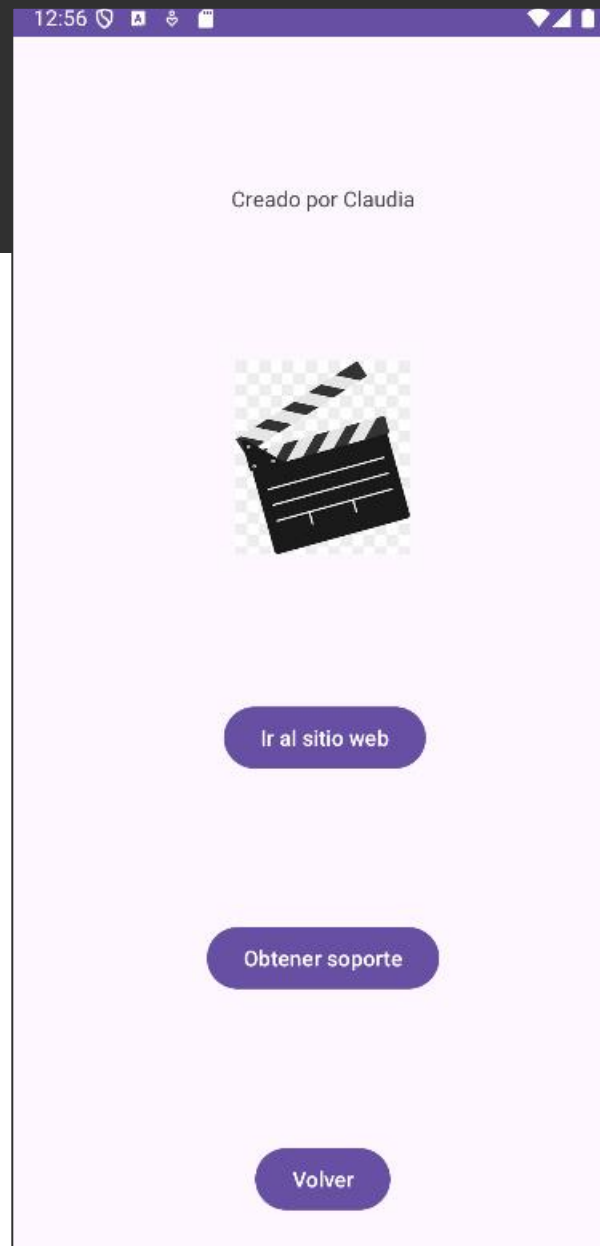
- **1º forma:** Arrastra y suelta tu archivo PNG dentro de la carpeta drawable. Asegúrate de que el nombre del archivo no contenga letras mayúsculas, espacios ni caracteres especiales (usa guiones bajos si es necesario).

# PROYECTO

Una vez importada la imagen, ve al archivo activity\_main.xml y configura la nueva imagen en el componente ImageView

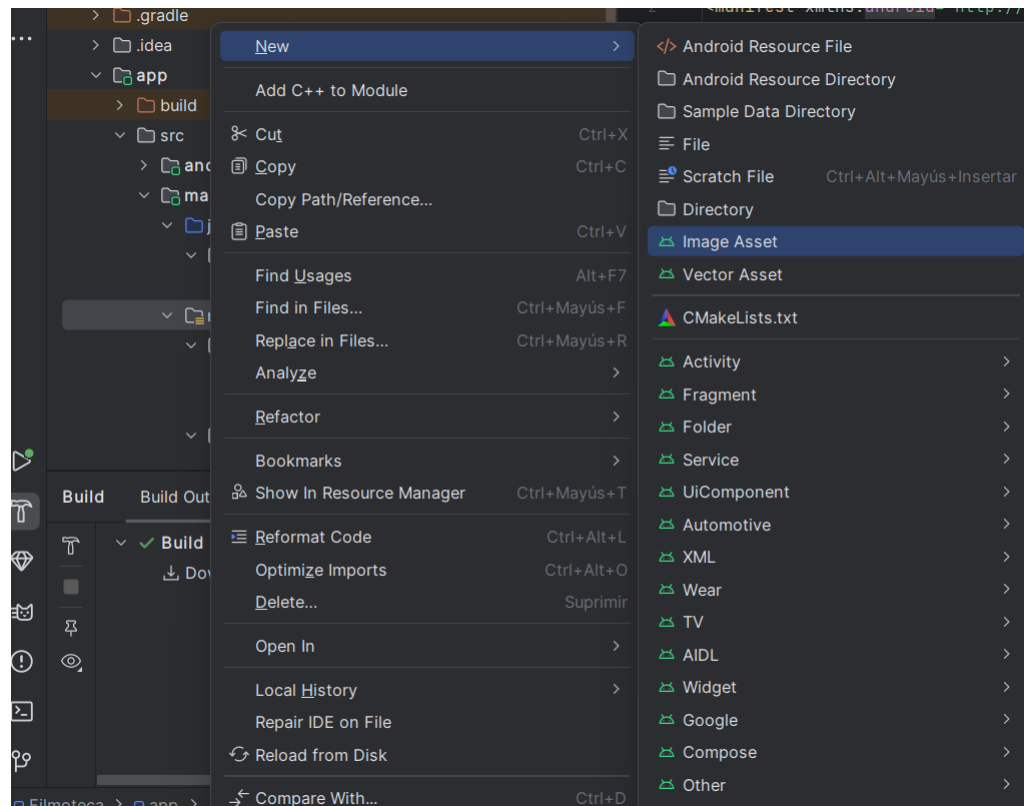


# PROYECTO

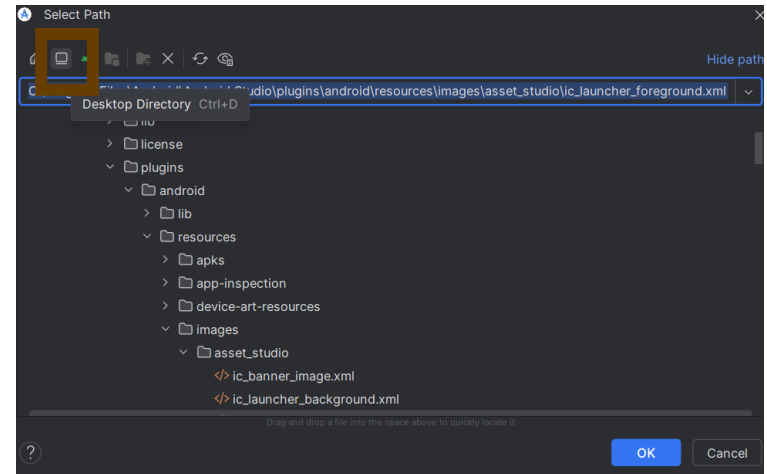
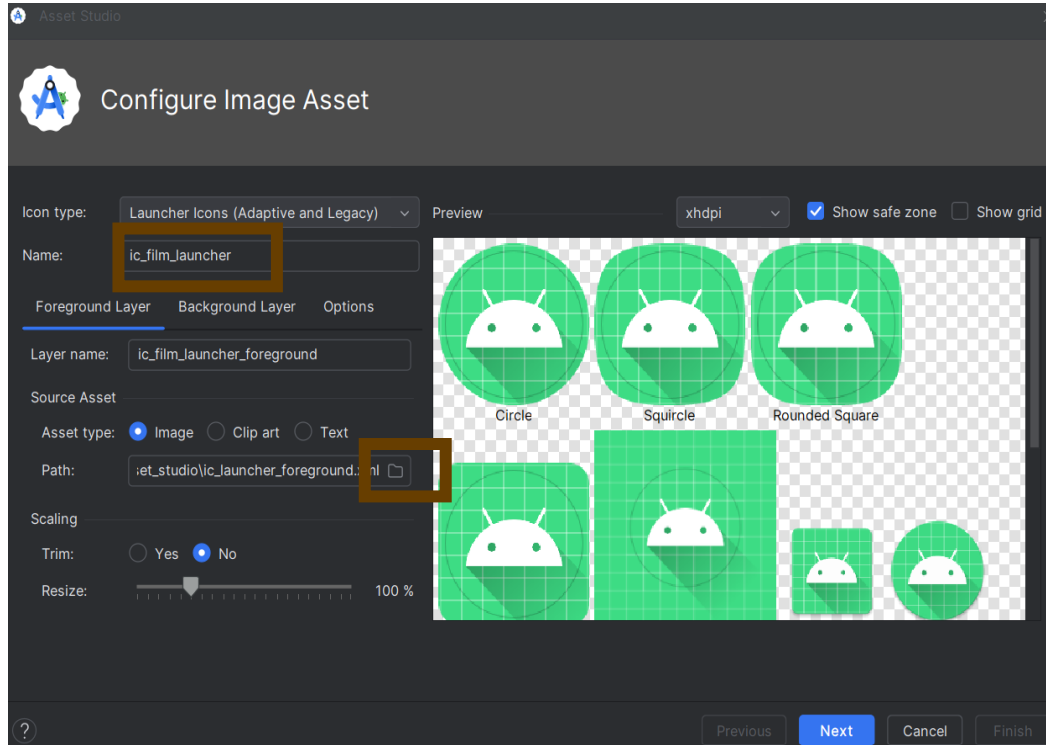


# PROYECTO

**2º forma:** Sobre la carpeta res, hacer clic botón derecho **New-> Image Asset**

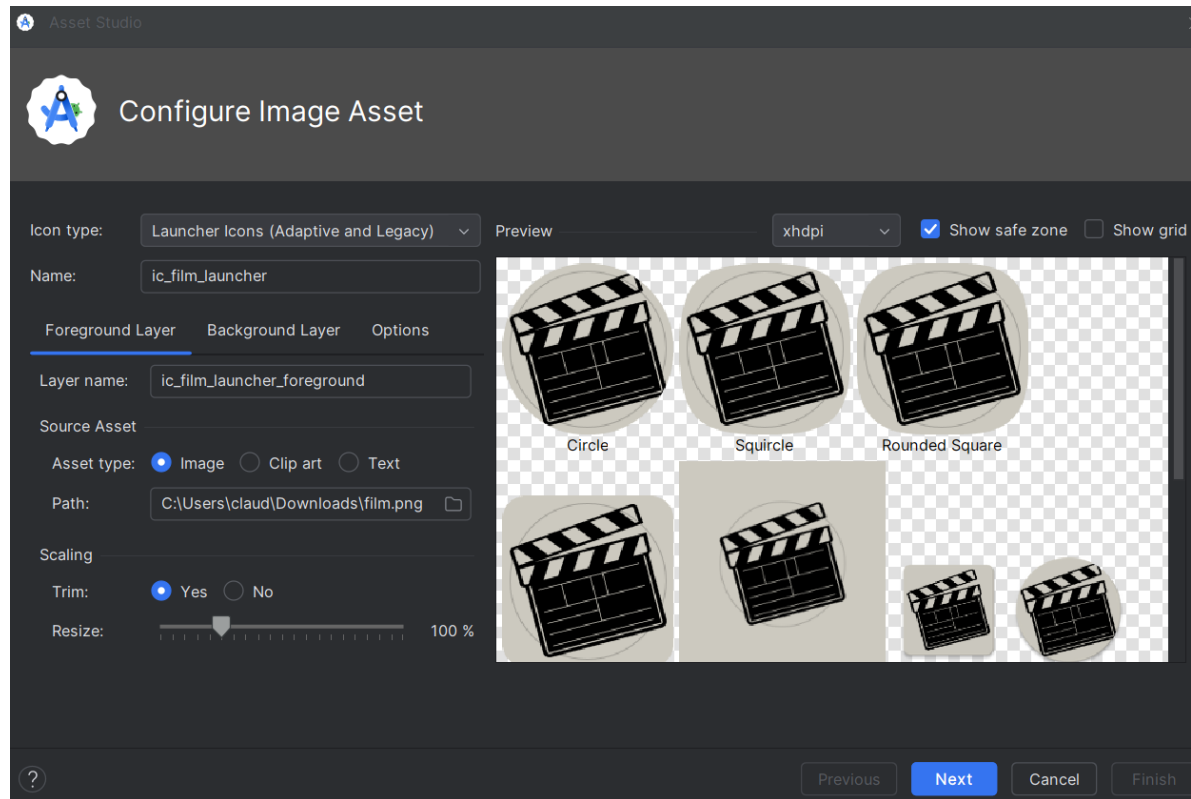


# PROYECTO



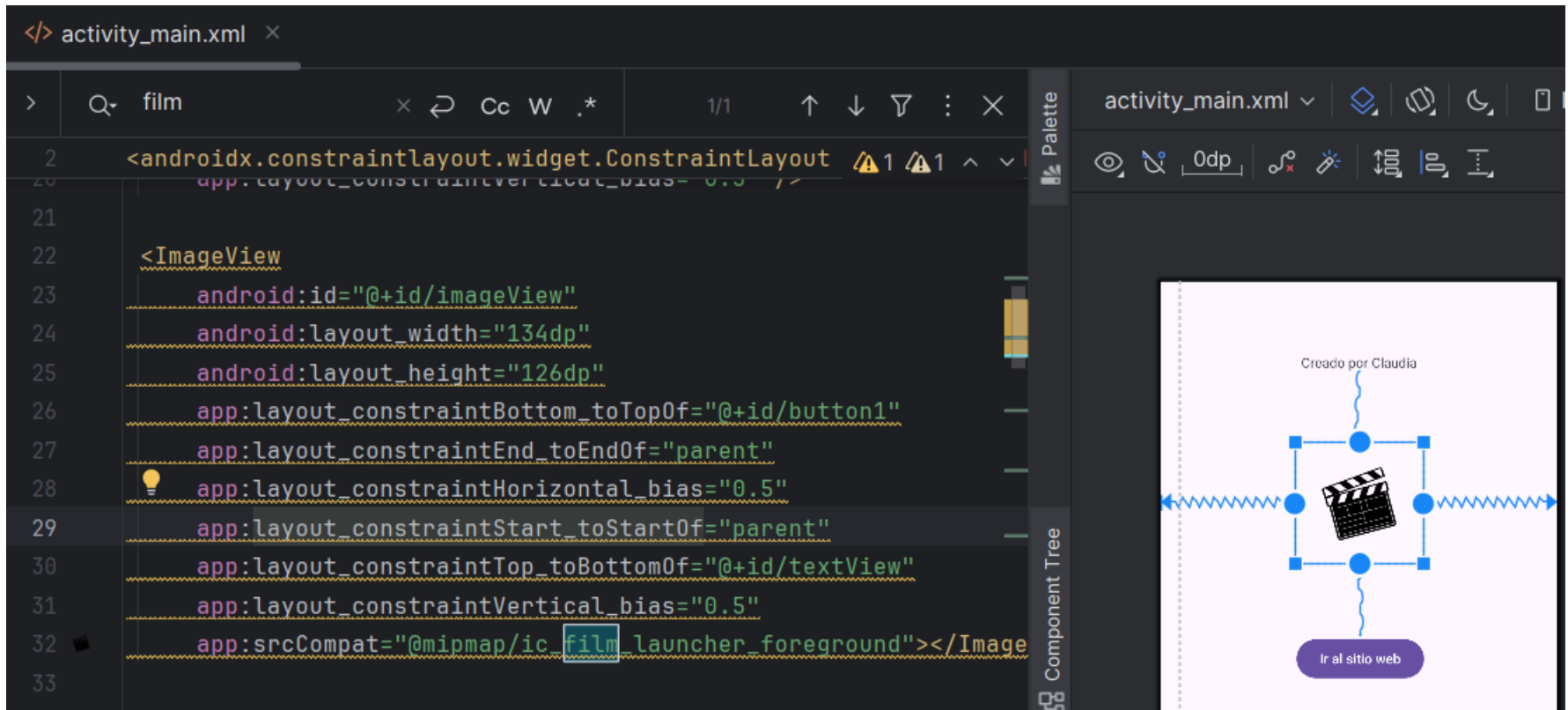
# PROYECTO

Configura la imagen a tu gusto y cuando termines haz clic en Next y luego en Finish



# PROYECTO

Una vez importada la imagen, ve al archivo activity\_main.xml y configura la nueva imagen en el componente ImageView





# PROYECTO

## INTERNACIONALIZACIÓN DEL PROYECTO

Por último, crearemos una nueva carpeta de recursos values con el clasificador en para que se aplique en dispositivos configurados en idioma inglés, por lo que la carpeta será res/values-en

Repetir el mismo proceso para que se aplique en dispositivos configurados en idioma español, por lo que la carpeta será res/values-es

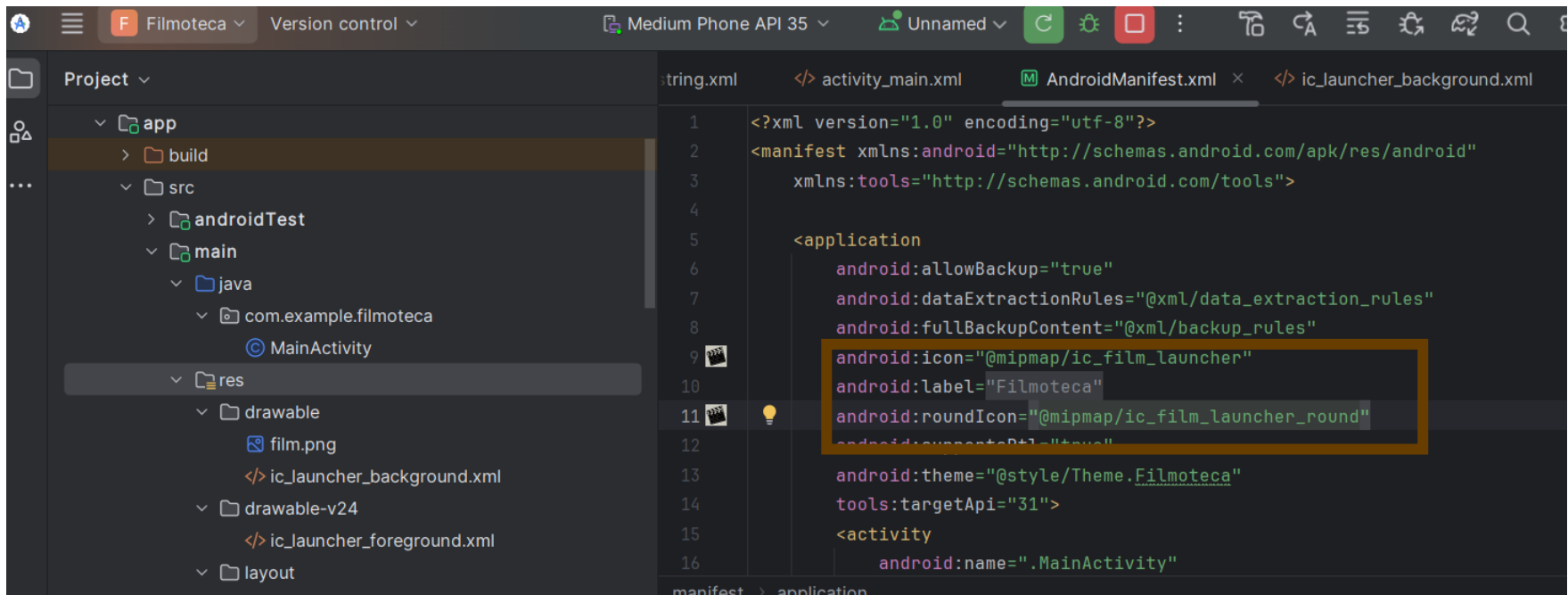
# PROYECTO

## ICONO DEL PROYECTO

Cambiaremos el icono del proyecto. Sigue los pasos que se muestran en las siguientes diapositivas

# PROYECTO

Ve al archivo AndroidManifest.xml y cambia los siguientes valores:



# PROYECTO

