Taller de desarrollo de dispositivos móviles

Guía de configuración de entorno para aplicaciones capacitor / android

Docente Marcelo Caiafa – Diciembre 2022

Modificado por : Liliana Pino – Febrero 2024

1-Instalaciones ADICIONALES

1-1 Descargar Android Studio https://developer.android.com/studio

Notas:

- Proceso de instalación dejar opciones por defecto.

1-2-Descargar/Instalar Node.js

https://nodejs.org/en/download/

Notas:

- Opción LTS (Long Term Support).
- Esto también instala npm (instalador de paquetes para Node.js).
- Se agrega Node.js a variable de entorno PATH de windows automáticamente. • "Automáticamente instalar herramientas necesarias..." (opcional)

1-3 Descargar Java – JDK

https://www.oracle.com/java/technologies/javase/jdk21-archive-downloads.html

1-4 Descargar última versión de Gradle

https://gradle.org/install/

Descomprimir el archivo zip de instalación de gradle en un directorio donde tenga permisos por ejemplo C:\Users\Usuario\

- Luego de descomprimir quedará una carpeta C:\Users\Usuario\gradle-XXX y dentro existe un subdirectorio bin, anotar el path hasta este subdirectorio, por ejemplo:

  C:\Users\Usuario\gradle-7.3.1\bin

1-5 Configurar variable de entorno en Windows para Gradle (Este equipo/ propiedades)

Editar la variable **Path** en **Variables de usuario**

Click en **Nuevo**, Agregar path a directorio bin de Gradle, luego **Aceptar**

1-6-Configurar variable de entorno para JAVA_HOME en variables de sistema. Indicar como path , la ruta del JDK
Si la variable no existe crearla con ese nombre

1-7-Configurar variable de entorno para ANDROID_HOME en variable de sistema. Indicar como path, la ruta donde queda el SDK de Android por defecto C:\Users\Usuario\AppData\Local\Android\Sdk

Si la variable no existe crearla con ese nombre

2-Instalaciones

1-4-Verificar **node** instalado

```
C:\Users\Usuario>node --version v14.15.1
```

1-5-Verificar **npm** instalado

```
C:\Users\Usuario>npm --version
6.14.8
```

1-6-Instalar **cli** (command line interface) de ionic utilizando npm

```
C:\Users\Usuario>npm i -g @ionic/cli
```

Notas:

- Opción -g instala ionic cli global, sin -g instala ionic cli en directorio node_modules de directorio actual (ionic cli quedará accesible solo desde este directorio).

1-7-Verificar **ionic cli** instalado

```
C:\Users\Usuario>ionic -v
10.0.0
```

Configuraciones adicionales

3-Crear proyecto capacitor

2.1-Crear directorio **www** y copiar todos los archivos de nuestro proyecto web (index.html, js, css, imágenes)

2.2-Crear archivo package.json (información básica de nuestra aplicación) - Esto se crea el ejecutar el siguiente comando.

```
npm init
```

2.3-Instalar capacitor a partir de proyecto web existente (todo nuestro proyecto web con archivos js, css, index.html debe estar contenido en un directorio con nombre **www**). La opción capacitor/**core** creará un directorio **node_modules** a nivel de nuestro proyecto con todas las dependencias de nuestra aplicación móvil. La opción capacitor/**cli** instala npx (command line interface de capacitor).

```
npm install @capacitor/core


npm install @capacitor/cli --save-dev
```

2-4-Iniciar proyecto capacitor

```
npx cap init
```

2-5-Crear proyecto Android (dejar opciones de preguntas por defecto)

```
npm install @capacitor/android
npx cap add android
```

Configuraciones adicionales

2-6-Generar archivo **capacitor.js** a partir de nuestra aplicación web. Debido a que no utilizamos ningún Module Bundler para importar plugins, es necesario generar un archivo capacitor.js a partir de nuestro proyecto web el cual será inyectado en cada una de las plataformas que trabajemos (android, ios). Para esto es necesario que nuestro archivo index.html contenga dentro de la etiqueta de apertura y cierre de head lo siguiente:

```
<script src="capacitor.js"></script>
```

Ejecutar luego el siguiente comando para sincronizar los archivos.

```
npx cap sync
```

Considerar que los cambios en nuestro proyecto web (html, js, nuevos plugins, etc.) se ven reflejados en nuestra aplicación móvil únicamente si ejecutamos el comando anterior.

2-7-Modificar permisos en el archivo **android/app/src/main/ AndroidManifest.xml**

```
<!-- Permissions -->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-feature android:name="android.hardware.location.gps" />
```

2-8-Crear y ejecutar APK (Se asume Android Studio Instalado)

```
npx cap run android
```

2-9-Esta última sentencia genera el archivo de nuestra aplicación móvil en:

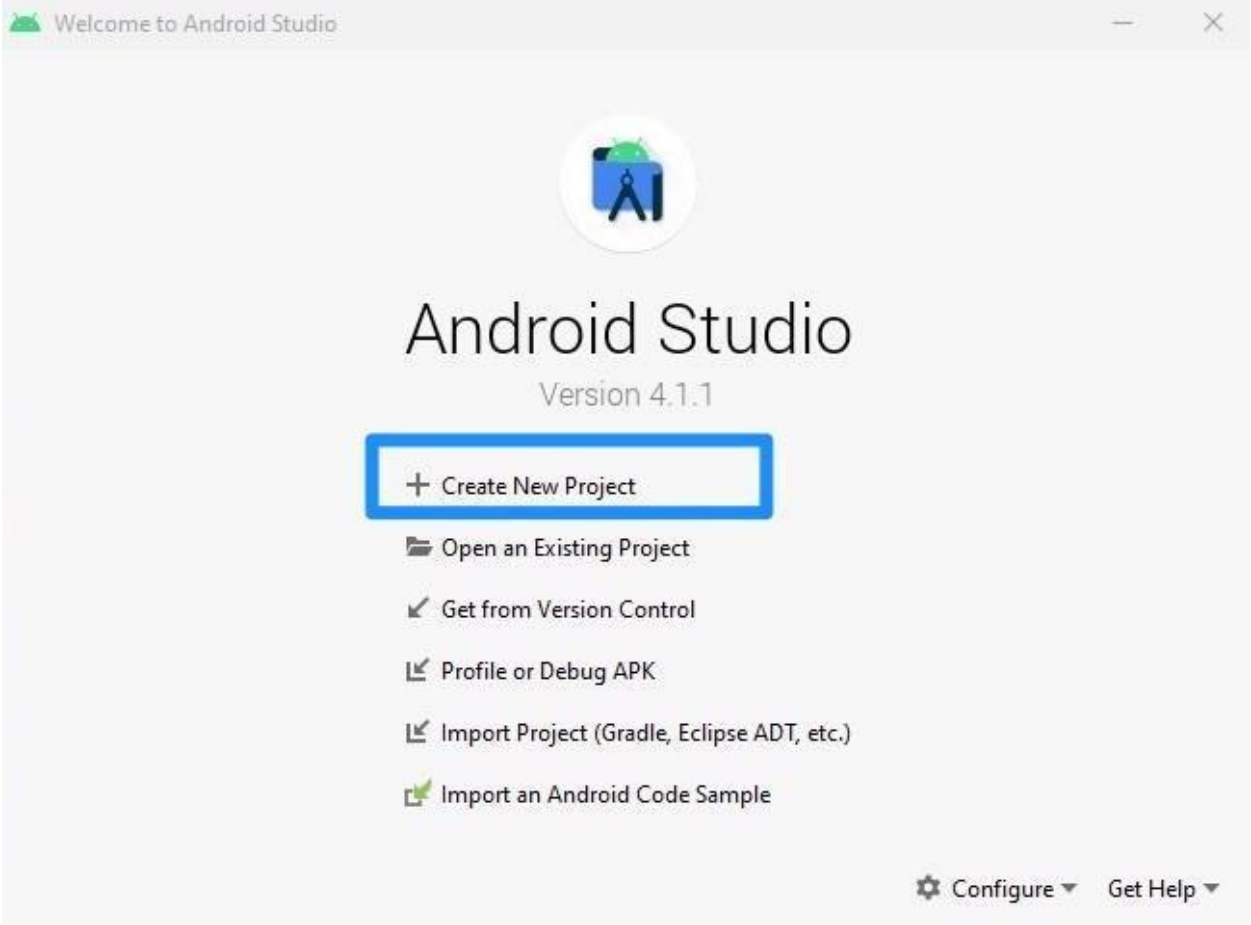
```
.../android/app/build/outputs/apk/debug/app-debug.apk
```

Notas:

4-

Configuraciones adicionales

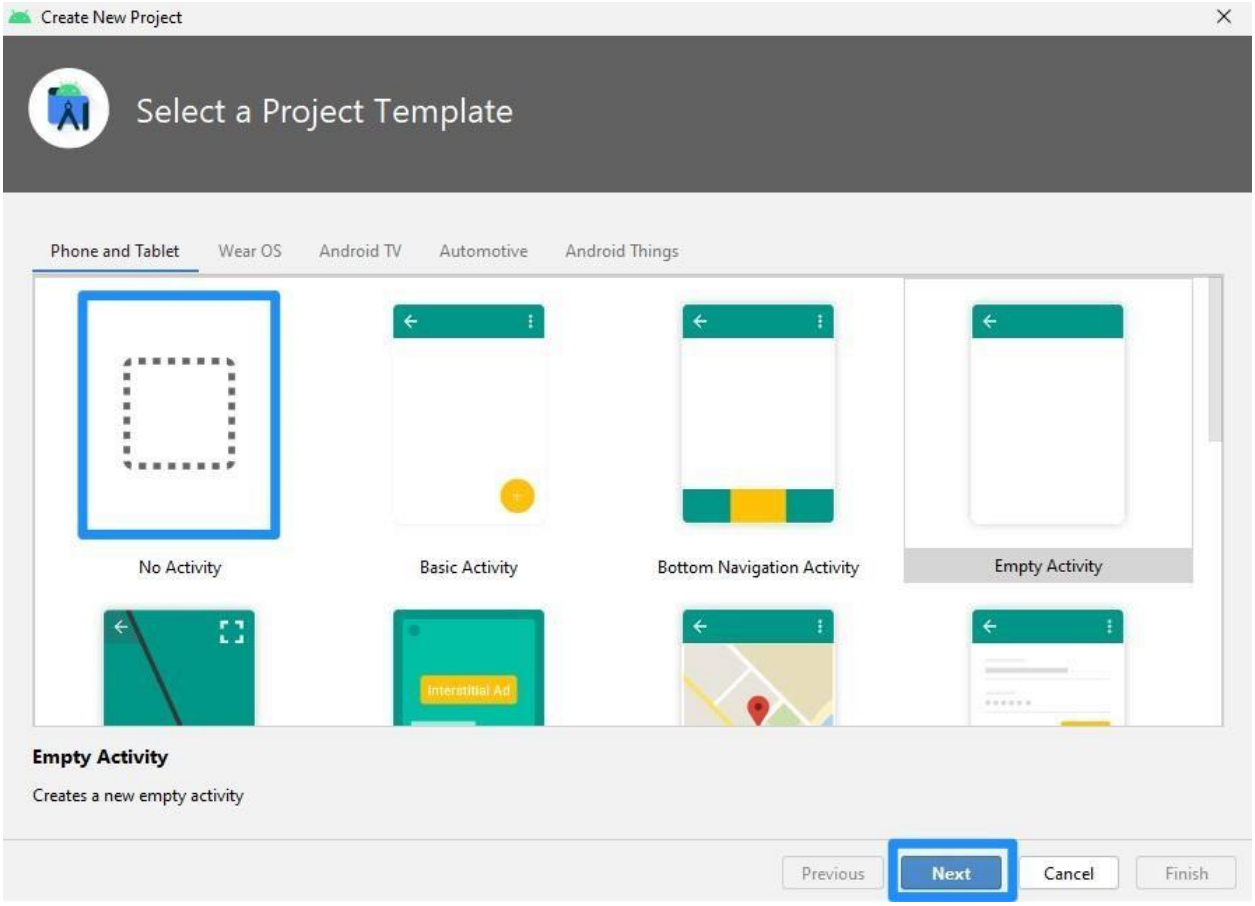
- Este archivo (apk) es nuestra aplicación la cual puede ser instalada en cualquier dispositivo Android.
- A partir de aquí podemos compartir el archivo por ejemplo vía google drive o testairy.com para descargarlo en nuestro dispositivo.



3-2-2-Seleccionar template **No Activity** y click botón **Next**

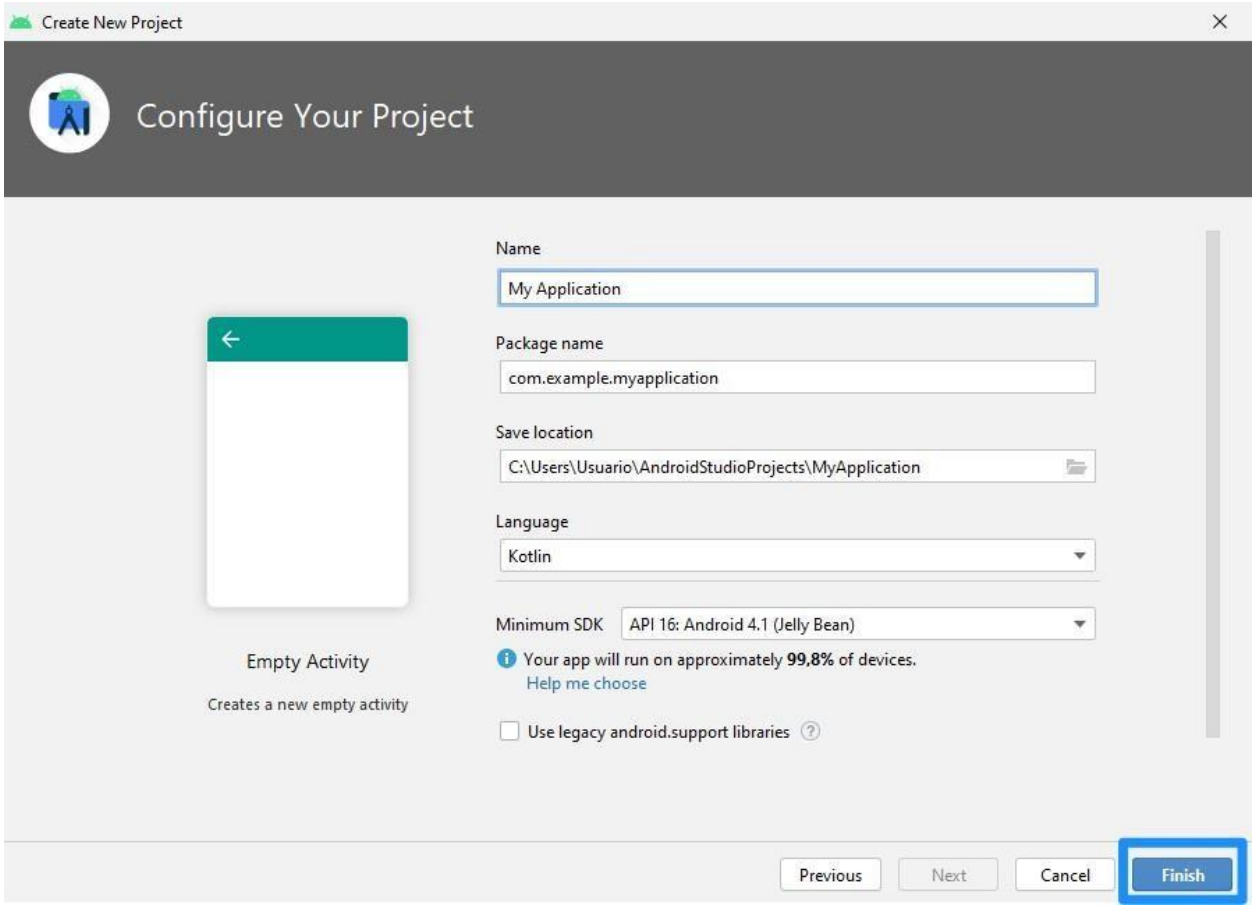
4-

Configuraciones adicionales



3-2-3-En configurar proyecto dejar opciones por defecto y click en **Finalizar**

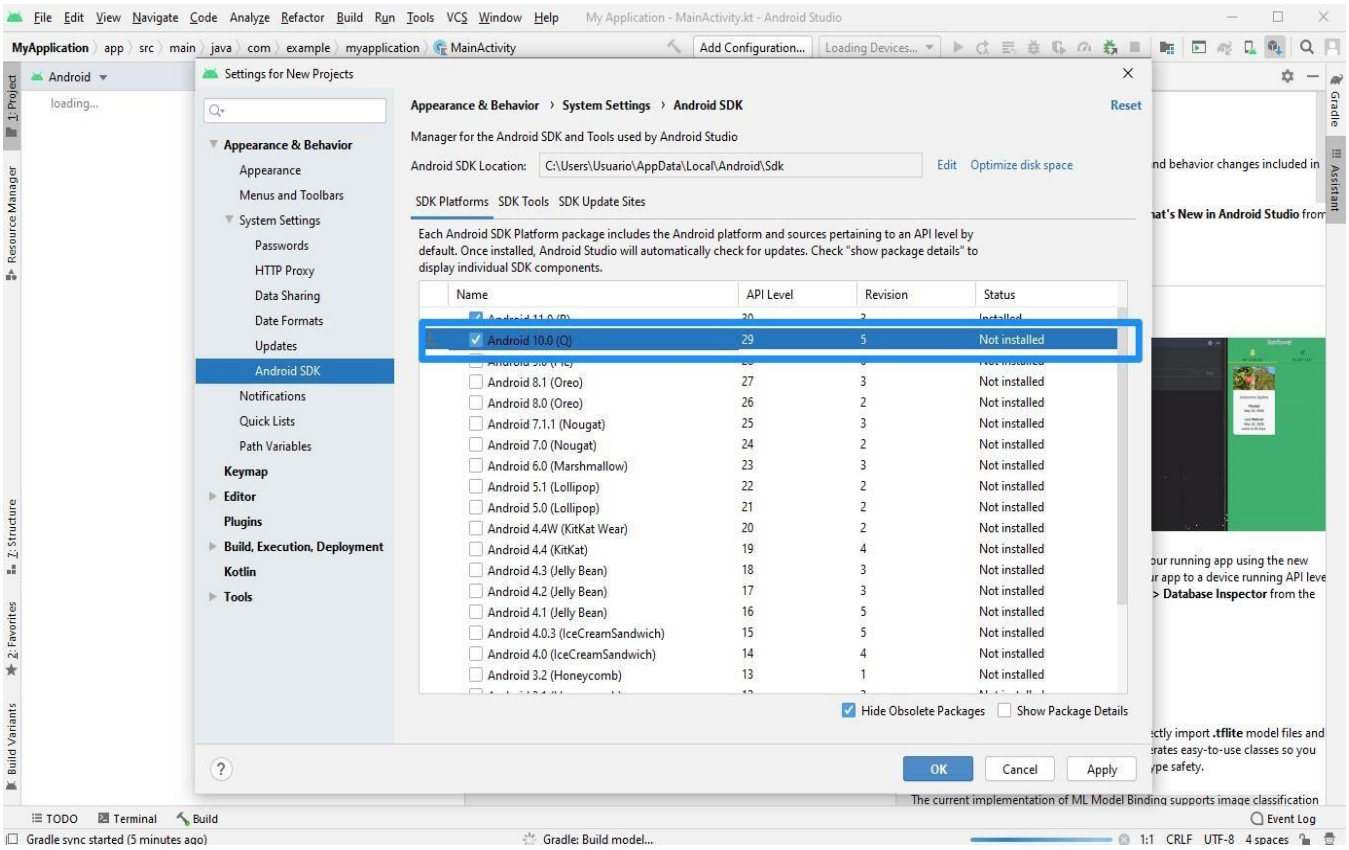
Configuraciones adicionales



3-2-4-En el menú seleccionar barra de herramientas, SDK Manager

3-2-5-En la nueva ventana, seleccionar checkbox para API Level 29 y 30 luego click OK

Configuraciones adicionales



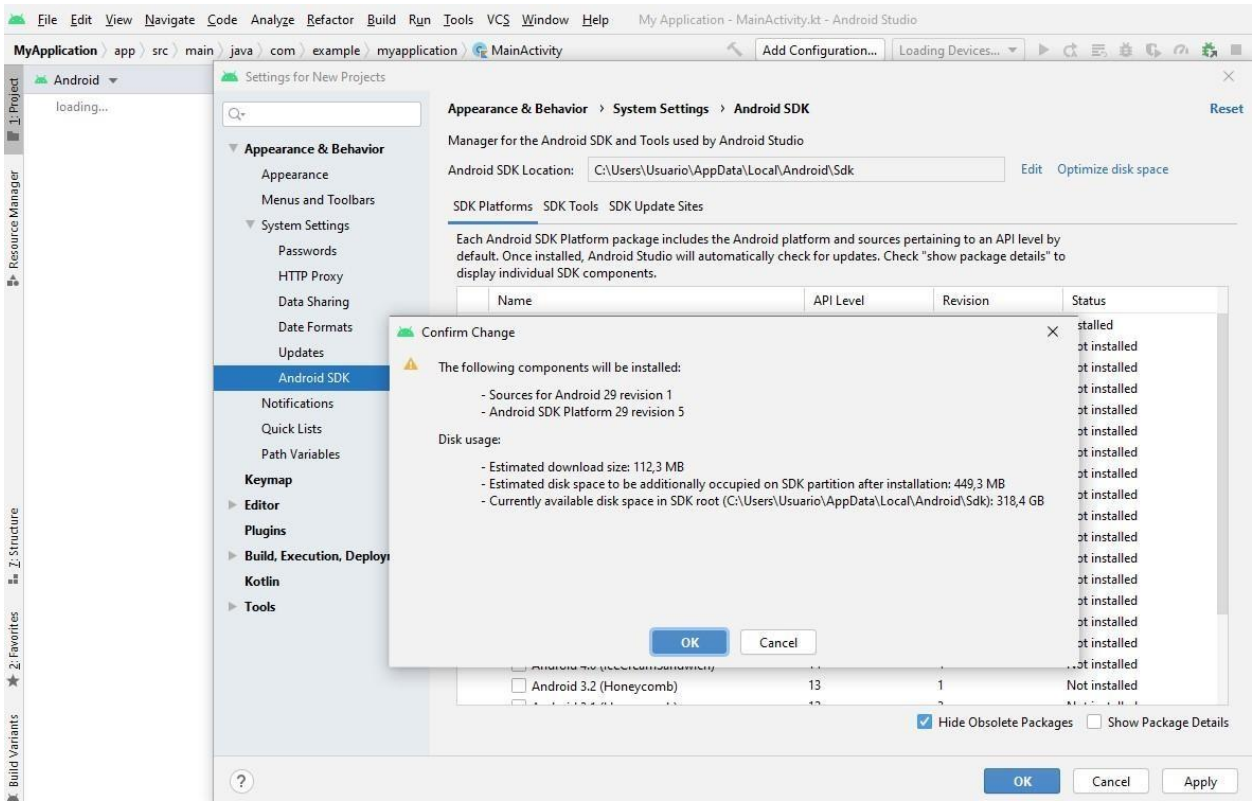
Notas:

- Estas apis están relacionadas con las distintas versiones de android y dispositivos virtuales en los cuales podemos ejecutar nuestra aplicación.

3-2-6-Confirmar cambios click **OK**

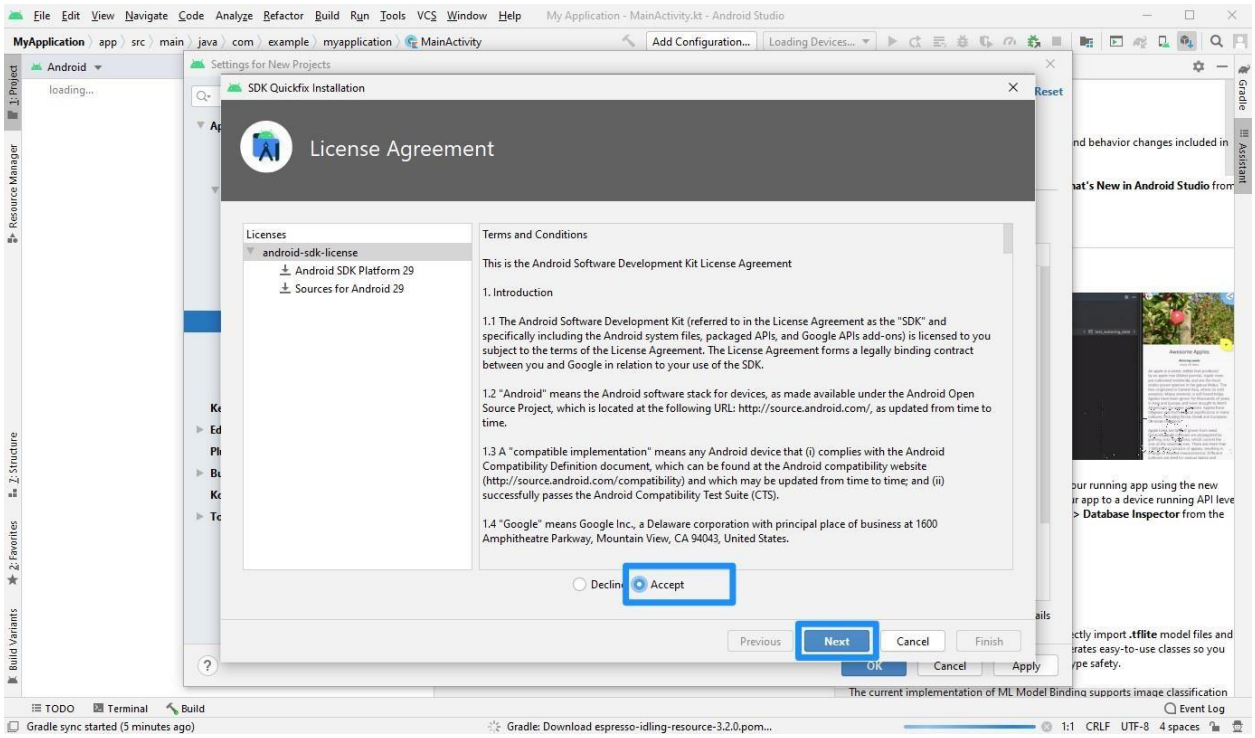
4-

Configuraciones adicionales



3-2-7-Acuerdo de licencia click radio **Accept** y **Next**

Configuraciones adicionales



3-2-8-Cuando finaliza la descarga click en **Finalizar**