


| | | |
|---|--|-------------------------------------|
|  | UNIVERSIDAD DON BOSCO FACULTAD DE INGENIERÍA ESCUELA DE COMPUTACIÓN | |
| CICLO: I/2021 | GUIA DE LABORATORIO #02 | |
| | Nombre de la Práctica: | Introducción Android |
| | MATERIA: | Desarrollo de Software para Móviles |

I. OBJETIVOS

1. Conocer los fundamentos del entorno de desarrollo Android.
2. Preparar el entorno de desarrollo con Android Studio.
3. Aprender a configurar un emulador para probar aplicaciones.
4. Crear una aplicación “Hola mundo” y ejecutarla en un emulador

II. INTRODUCCION TEORICA

Android

Sistema operativo basado en Linux, diseñado principalmente para móviles con pantalla táctil como teléfonos inteligentes o tabletas inicialmente desarrollados por Android, Inc., que Google respaldó financieramente y más tarde compró en 2005.

Versionamientos.

| ANDROID PLATFORM VERSION | API LEVEL | CUMULATIVE DISTRIBUTION |
|--------------------------|-----------|-------------------------|
| 4.0 Ice Cream Sandwich | 15 | |
| 4.1 Jelly Bean | 16 | 99.8% |
| 4.2 Jelly Bean | 17 | 99.2% |
| 4.3 Jelly Bean | 18 | 98.4% |
| 4.4 KitKat | 19 | 98.1% |
| 5.0 Lollipop | 21 | 94.1% |
| 5.1 Lollipop | 22 | 92.3% |
| 6.0 Marshmallow | 23 | 84.9% |
| 7.0 Nougat | 24 | 73.7% |
| 7.1 Nougat | 25 | 66.2% |
| 8.0 Oreo | 26 | 60.8% |
| 8.1 Oreo | 27 | 53.5% |
| 9.0 Pie | 28 | 39.5% |
| 10. Android 10 | 29 | 8.2% |

Fundamentos.

- Las herramientas de Android SDK generan un archivo **.apk**.
- Un APK tiene archivos de recursos y datos (paquete de Android)

Una vez instalada en el dispositivo, cada aplicación de Android se aloja en su propia zona de pruebas de seguridad:

- El sistema operativo Android es un sistema Linux multiusuario en el que cada aplicación es un usuario diferente.
- De forma predeterminada, el sistema le asigna a cada aplicación una ID de usuario de Linux única (solo el sistema utiliza la ID y la aplicación la desconoce). El sistema establece permisos para todos los archivos en una aplicación de modo que solo el ID de usuario asignado a esa aplicación pueda acceder a ellos.
- Cada proceso tiene su propio equipo virtual (EV), por lo que el código de una aplicación se ejecuta de forma independiente de otras aplicaciones.
- De forma predeterminada, cada aplicación ejecuta su proceso de Linux propio.

Componentes.

| Componente | Descripción | Clase y etiqueta para declararlo en archivo de manifiesto. |
|------------------------|--|--|
| Actividades (Activity) | Una <i>actividad</i> representa una pantalla con interfaz de usuario. Ej. App de correo, App de contactos, etc. | Activity <activity> |
| Servicios (Service) | Un <i>servicio</i> es un componente que se ejecuta en segundo plano para realizar operaciones prolongadas o tareas para procesos remotos. Un servicio no proporciona una interfaz de usuario. Por ejemplo, un servicio podría reproducir música en segundo plano mientras el usuario se encuentra en otra aplicación | Service <service> |

Componentes.

| Componente | Descripción | Clase y etiqueta para declararlo en archivo de manifiesto. |
|--|--|--|
| Proveedores de contenido (ContentProvider) | Un <i>proveedor de contenido</i> administra un conjunto compartido de datos de la app. Puedes almacenar los datos en el sistema de archivos, en una base de datos SQLite, en la Web o en cualquier otra ubicación de almacenamiento persistente a la que tu aplicación pueda acceder. A través del proveedor de contenido, otras aplicaciones pueden consultar o incluso modificar los datos (si el proveedor de contenido lo permite) | ContentProvider <provider> |
| Receptores de mensajes (BroadcastReceiver) | Un <i>receptor de mensajes</i> es un componente que responde a los anuncios de mensajes en todo el sistema. Muchos mensajes son originados por el sistema; por ejemplo, un mensaje que anuncie que se apagó la pantalla, que la batería tiene poca carga o que se tomó una foto. Las aplicaciones también pueden iniciar mensajes; por ejemplo, para permitir que otras aplicaciones sepan que se descargaron datos al dispositivo y están disponibles para usarlos. | BroadcastReceiver <receiver> |

III. PROCEDIMIENTO

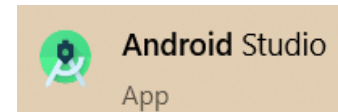
Para realizar la práctica debe tener instalado:

- **Java JDK** (Según el sistema operativo que posea en su computador)
 - Enlace de descarga: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
- **Android Studio.**
 - Enlace de descarga: <https://developer.android.com/studio>

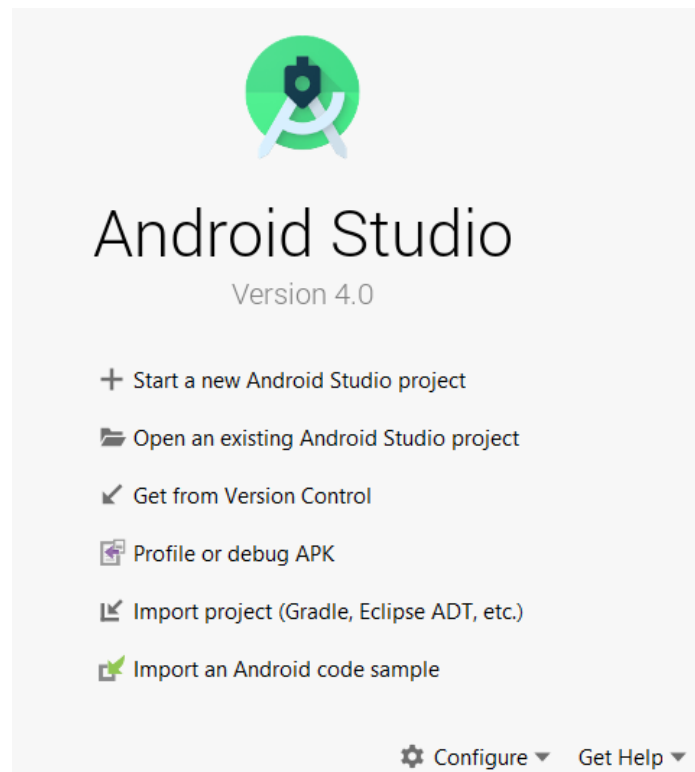
Creación de un nuevo proyecto con Android Studio.

Teniendo instalado Java JDK y Android Studio siga los siguientes pasos:

1. Usando el acceso directo que tiene en su escritorio ingrese a Android Studio.

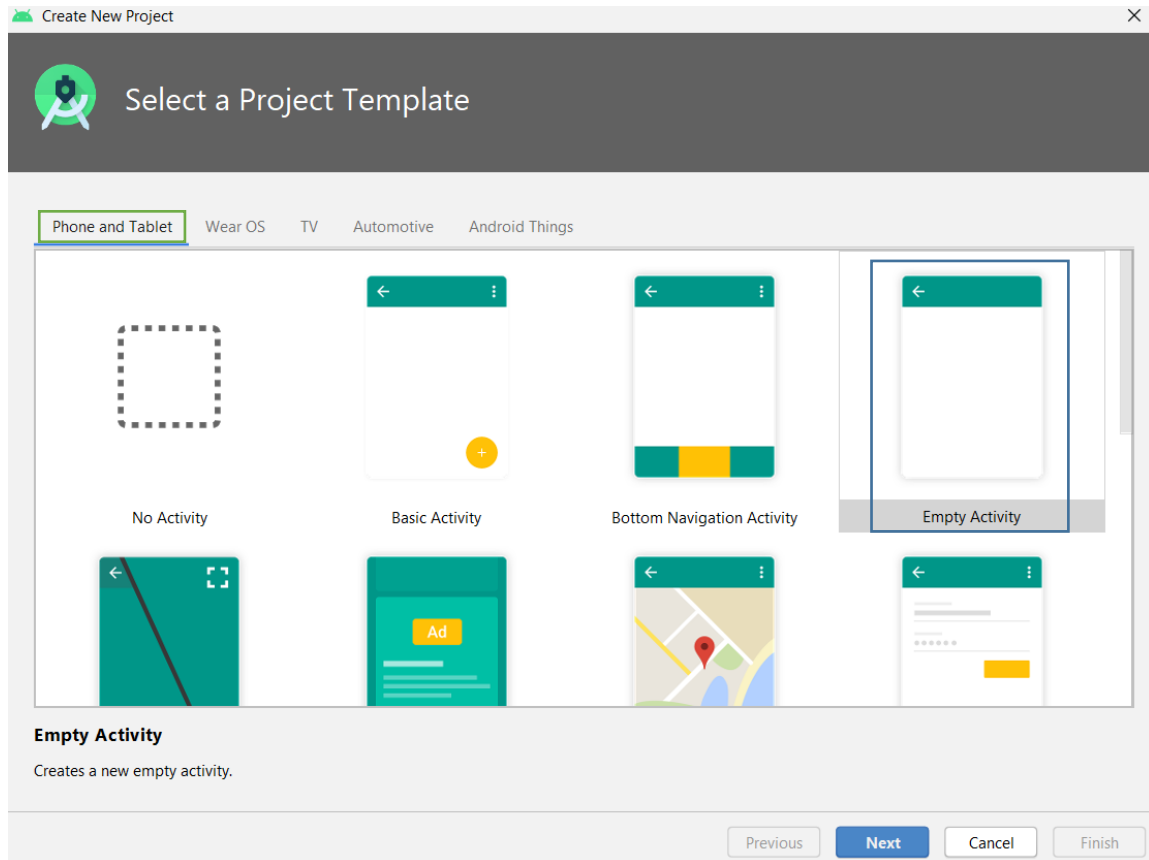


Esto permite ingresar a la pantalla de inicio de Android Studio como la siguiente:



2. Seleccione la opción **“Start a new Android Studio Project”** o su equivalente en español, para proceder a crear nuestro primer proyecto.

Se mostrará la siguiente pantalla para poder: seleccionar el tipo de dispositivo donde vamos a poder ejecutar nuestra aplicación Android, alguna plantilla de proyecto predeterminada, etc.



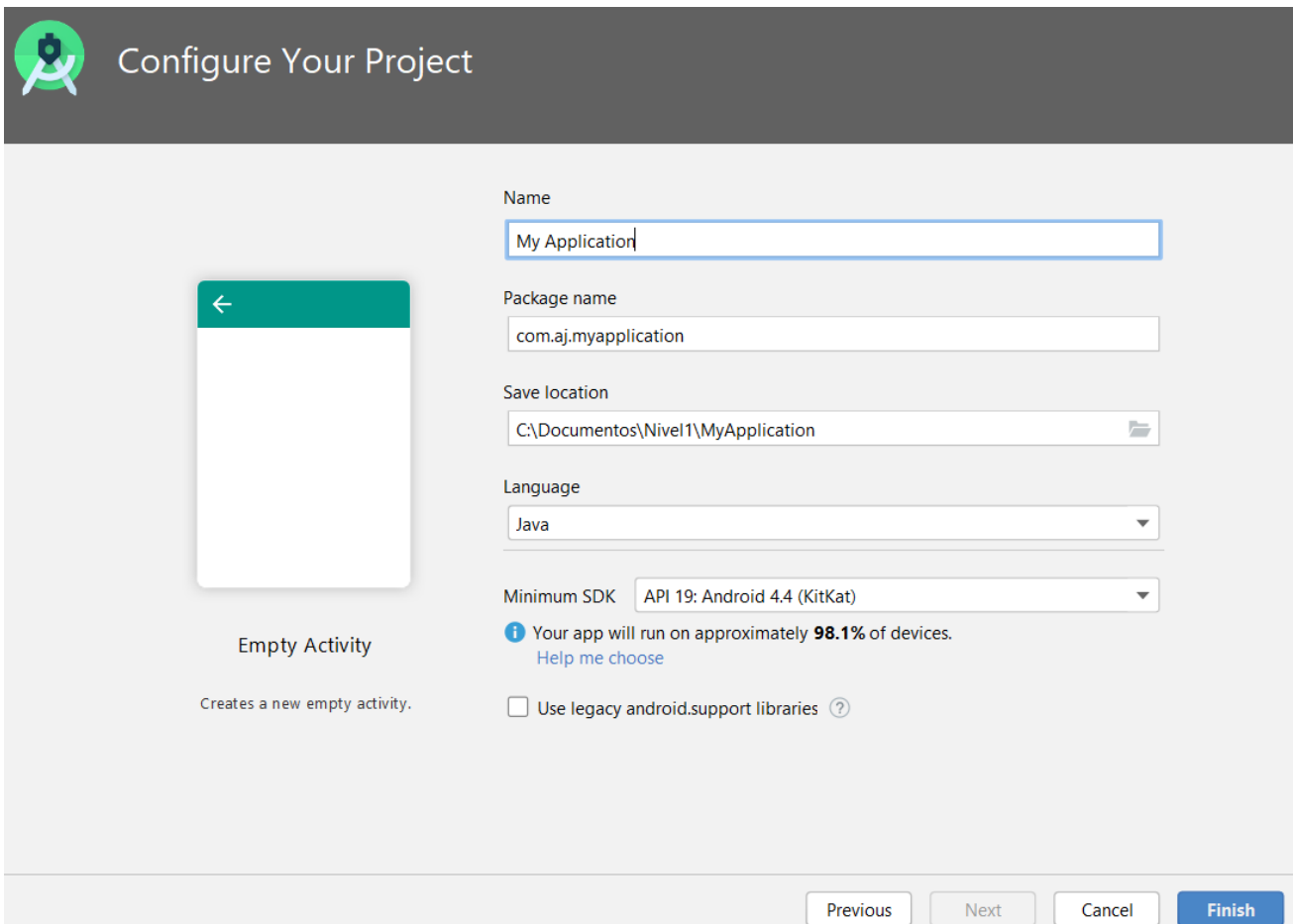
3. Vamos a crear nuestro ejemplo para que pueda ejecutarse **en teléfonos y tabletas** y además vamos a seleccionar una de las plantillas que vienen por denominada **“Empty Activity”** (Actividad vacía)
4. Selecciona el botón **Next** (Siguiente)

Deberá aparecer una pantalla como la siguiente:

Esta pantalla nos permite configurar:

- a- **El nombre de nuestra aplicación.** (Será el nombre que el usuario verá en su dispositivo cuando instale la aplicación)
- b- **El nombre del paquete.** Representa el identificador único de la aplicación y debes asegurarte de que no se repita con ninguna otra aplicación Android en el mundo. Este nombre de paquete será el nombre de la carpeta donde se instalará la aplicación en el dispositivo (dentro de \data\data) por lo tanto es imprescindible que selecciones un nombre que representa a la aplicación como única. También cuando quieras subir tu aplicación a la tienda de Google Play para distribuirla, Google realizará una validación evitando que este nombre de paquete ya exista previamente asignado a otra aplicación.
- c- **Ubicación del proyecto (Save location).** Es la carpeta donde se guardará tu proyecto fuente
- d- **Lenguaje.** Es el lenguaje que utilizarás para desarrollar tu aplicación (Java o Kotlin)
- e- **SDK Mínimo.** Es la versión mínima de Android con la cuál tu aplicación será compatible. Es decir si un cliente quiere instalar tu aplicación en su dispositivo deberá tener instalado en su dispositivo ésta versión de Android como mínimo.

5- Selecciona **Finish** (Finalizar)



Configure Your Project

Name
My Application

Package name
com.aj.myapplication

Save location
C:\Documentos\Nivel1\MyApplication

Language
Java

Minimum SDK
API 19: Android 4.4 (KitKat)

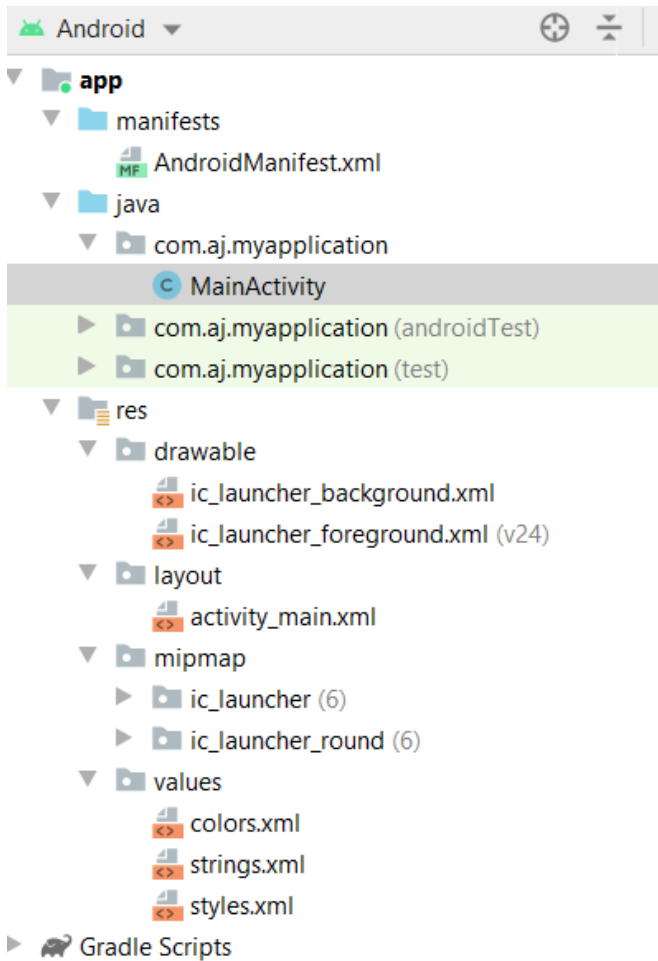
Empty Activity
Creates a new empty activity.

Information: Your app will run on approximately **98.1%** of devices.
[Help me choose](#)

☐ Use legacy android.support libraries ?

Buttons: Previous, Next, Cancel, **Finish**

Espera unos segundos mientras se configura el proyecto y al final deberás observar lo siguiente:



descripción breve.

AndroidManifest.xml: Archivo de configuración de la aplicación para definir algunos elementos como: permisos, actividades, servicios, u otros componentes que formaran parte de la aplicación.

\java\<nombre de paquete>

Paquete principal de la aplicación donde podremos encontrar la actividad principal de nuestro proyecto y donde podremos agregar otros paquetes y otras clases.

res\drawable

Carpeta que puede contener recursos de tipo imágenes.

res\layout

Carpeta que puede contener recursos que representan los diseños de pantalla de nuestra aplicación.

res\mipmap

Carpeta que contiene recursos que contienen imágenes que representan a nuestra aplicación.

res\values

Carpeta que puede contener recursos como listas de valores que podrán ser leídos por

nuestra aplicación (listas de colores, estilos, cadenas, menús, etc.)

Configuración de emulador.

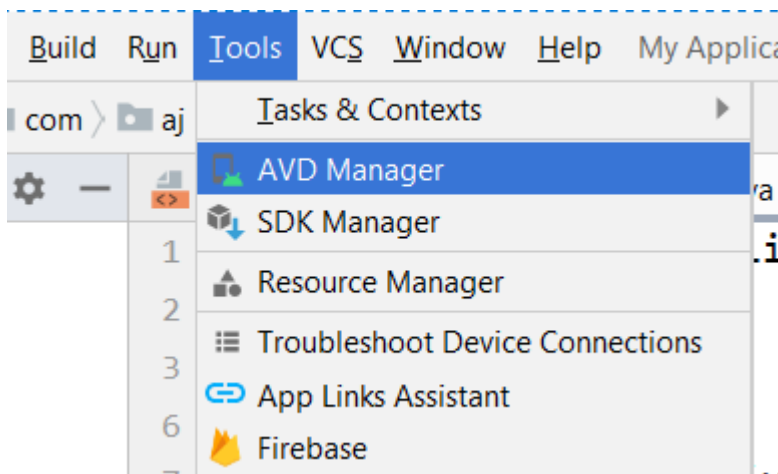
Un emulador es una máquina virtual que representa un dispositivo Android con una versión del sistema operativo y un hardware determinado.

Con el emulador podremos hacer pruebas de nuestra aplicación antes de ser puesta en un entorno de producción y sin la necesidad (si así se requiere) de contar con un dispositivo físico.

Antes de poder ejecutar la aplicación que acabamos de crear con los pasos anteriores vamos a crear un emulador donde será ejecutada dicha aplicación.

Creación de emulador.

1. Selecciona las opciones **Tools->AVD Manager** (Herramientas) en el menú principal de Android studio.
Esto permite ingresar a la ventana de administrador de Dispositivos Virtuales de Android (AVD)




Esto te permitirá acceder a la siguiente pantalla:

Esta consola te permite administrar todos tus dispositivos virtuales que tengas configurado en tu equipo.





































Esto quiere decir que puedes : Agregar, eliminar, modificar características, activar dichos dispositivos virtuales.



Podrás contar con varios dispositivos (Ten cuidado con esto, pues cada dispositivo ocupa una carpeta con archivos en tu disco duro y el espacio utilizado será proporcional a las características que hayas definido para cada dispositivo virtual.)



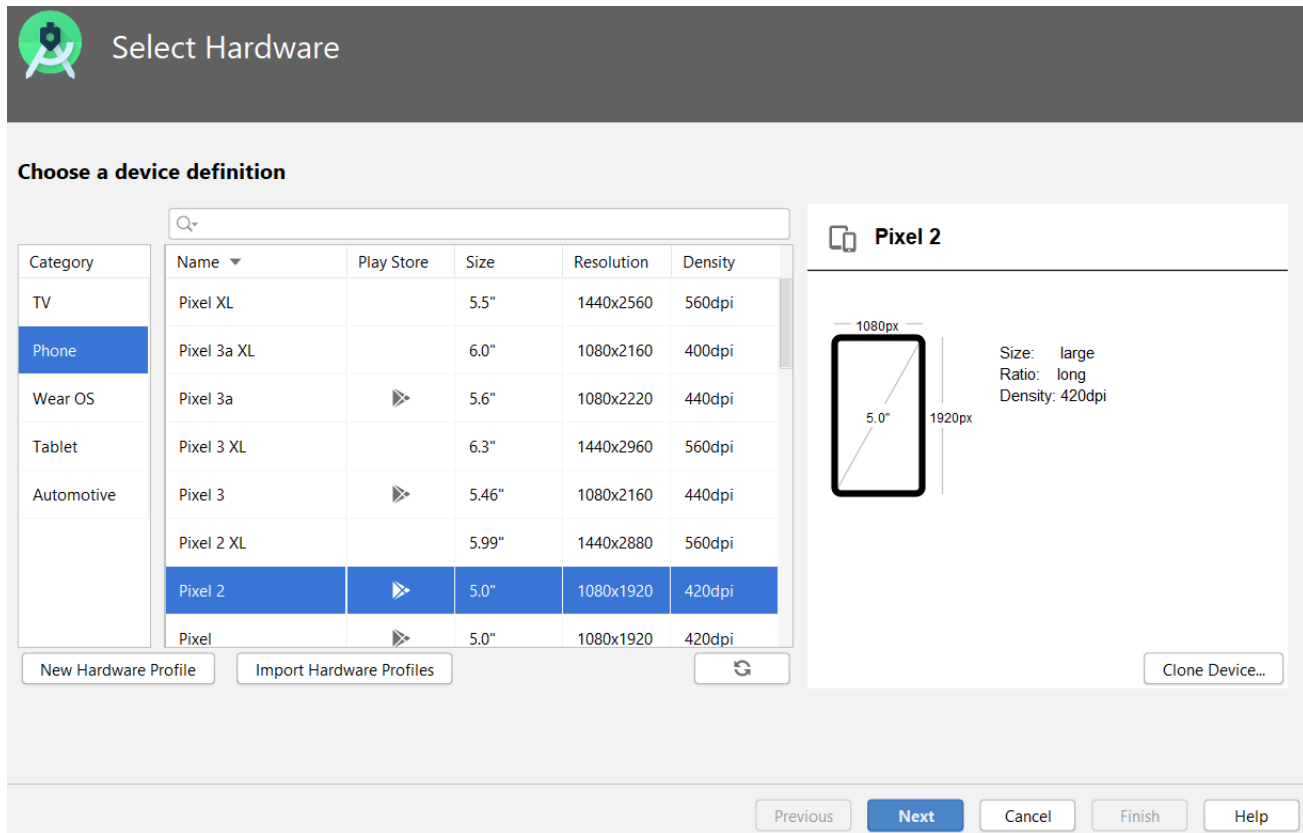
Your Virtual Devices

Android Studio

| Type | Name | Play Store | Resolution | API | Target | CPU/ABI | Size on ... | Actions |
|---|--------------------------------|------------|---------------|-----|-------------------------|---------|-------------|---|
|  | 10.1 WXGA (Tablet) API 22 | | 800 × 1280... | 22 | Android 5.1 | x86 | 2.5 GB |    |
|  | 10.1 WXGA (Tablet) (TABLET ... | | 800 × 1280... | 23 | Android 6.0 | x86 | 2.8 GB |    |
|  | Emulador Gigerbread | | 1080 × 192... | 10 | Android 2.3.3 | x86 | 5.7 GB |    |
|  | Emulador Marshmallow-2 | | 1080 × 192... | 23 | Android 6.0 (Google ... | x86_64 | 3.5 GB |    |
|  | Nexus 5X API 19 | | 1080 × 192... | 19 | Android 4.4 (Google ... | x86 | 5.2 GB |    |
|  | Nexus 5X API 22 | | 1080 × 192... | 22 | Android 5.1 (Google ... | x86 | 3.0 GB |    |
|  | Nexus 5X API 25 | | 1080 × 192... | 25 | Android 7.1.1 (Googl... | x86_64 | 3.7 GB |    |
|  | Nexus 5X API 25 2 | | 1080 × 192... | 25 | Android 7.1.1 (Googl... | x86_64 | 3.6 GB |    |
|  | Nexus 5X API 28 | | 1080 × 192... | 28 | Android 9.0 (Google ... | x86 | 3.5 GB |    |

+ Create Virtual Device...



2- Selecciona el botón “**Create Virtual Device...**” (Crear dispositivo virtual)




Esta ventana permite definir las características de hardware que simulará tu dispositivo virtual :

- (categoría de dispositivo: teléfono, Tablet, etc)
- Tamaño de la pantalla

3- Selecciona en Category : **Phone** y un modelo como el mostrado en pantalla : **pixel 2** (5.0")

4- Selecciona el botón “**Next**” (Siguiente)

Ahora procederemos a seleccionar la imagen o versión del sistema operativo Android que deseamos emular en nuestro dispositivo.




System Image

Select a system image

Recommended x86 Images Other Images

| Release Name | API Level ▼ | ABI | Target |
|--|-------------|-----|-----------------------------|
| R Download | 30 | x86 | Android 10.0+ (Google Play) |
| Q Download | 29 | x86 | Android 10.0 (Google Play) |
| Pie Download | 28 | x86 | Android 9.0 (Google Play) |
| Oreo Download | 27 | x86 | Android 8.1 (Google Play) |
| Oreo Download | 26 | x86 | Android 8.0 (Google Play) |
| Nougat Download | 25 | x86 | Android 7.1.1 (Google Play) |
| Nougat Download | 24 | x86 | Android 7.0 (Google Play) |

R



API Level
30

Android
10.0+

Google Inc.

System Image
x86

We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?
[See the API level distribution chart](#)

i A system image must be selected to continue.

Previous Next Cancel Finish Help

En el tab **“Recommended”** podrás encontrar versiones de sistema operativo recomendado por el asistente.

Si tienes instalado/descargado en tu SDK estas versiones podrías seleccionar una de estas, de lo contrario observarás que a la par de cada versión de sistema operativo está una opción **“Download”** (No la selecciones).


La descarga de una versión del sistema operativo de Android puede tardarse varios minutos dependiendo de tu enlace de Internet.

Puedes también revisar en la pestaña **x86-images** para ver que opciones tienes disponibles además y poder seleccionar una versión de Android.

Busca una de las imágenes que ya tengas disponibles en tu computador (una que no tenga “download”), de lo contrario tendrás que descargar una.

5- En el ejemplo se selecciona **Marshmallow** como ejemplo. (Pero tu puedes seleccionar otra imagen que ya tengas disponible)

6- Seleccionar el botón “**Next**” (Siguiente)




System Image

Select a system image

Recommended x86 Images Other Images

| Release Name | API Level | ABI | Target |
|---|-----------|--------|---------------------------|
| <i>Nougat</i> Download | 25 | x86_64 | Android 7.1.1 |
| Nougat | 24 | x86_64 | Android 7.0 (Google APIs) |
| <i>Nougat</i> Download | 24 | x86 | Android 7.0 (Google APIs) |
| <i>Nougat</i> Download | 24 | x86_64 | Android 7.0 |
| <i>Nougat</i> Download | 24 | x86 | Android 7.0 |
| Marshmallow | 23 | x86_64 | Android 6.0 (Google APIs) |
| <i>Marshmallow</i> Download | 23 | x86 | Android 6.0 (Google APIs) |
| Marshmallow | 23 | x86 | Android 6.0 |
| <i>Marshmallow</i> Download | 23 | x86_64 | Android 6.0 |
| Lollipop | 22 | x86 | Android 5.1 (Google APIs) |

Marshmallow



API Level
23

Android
6.0

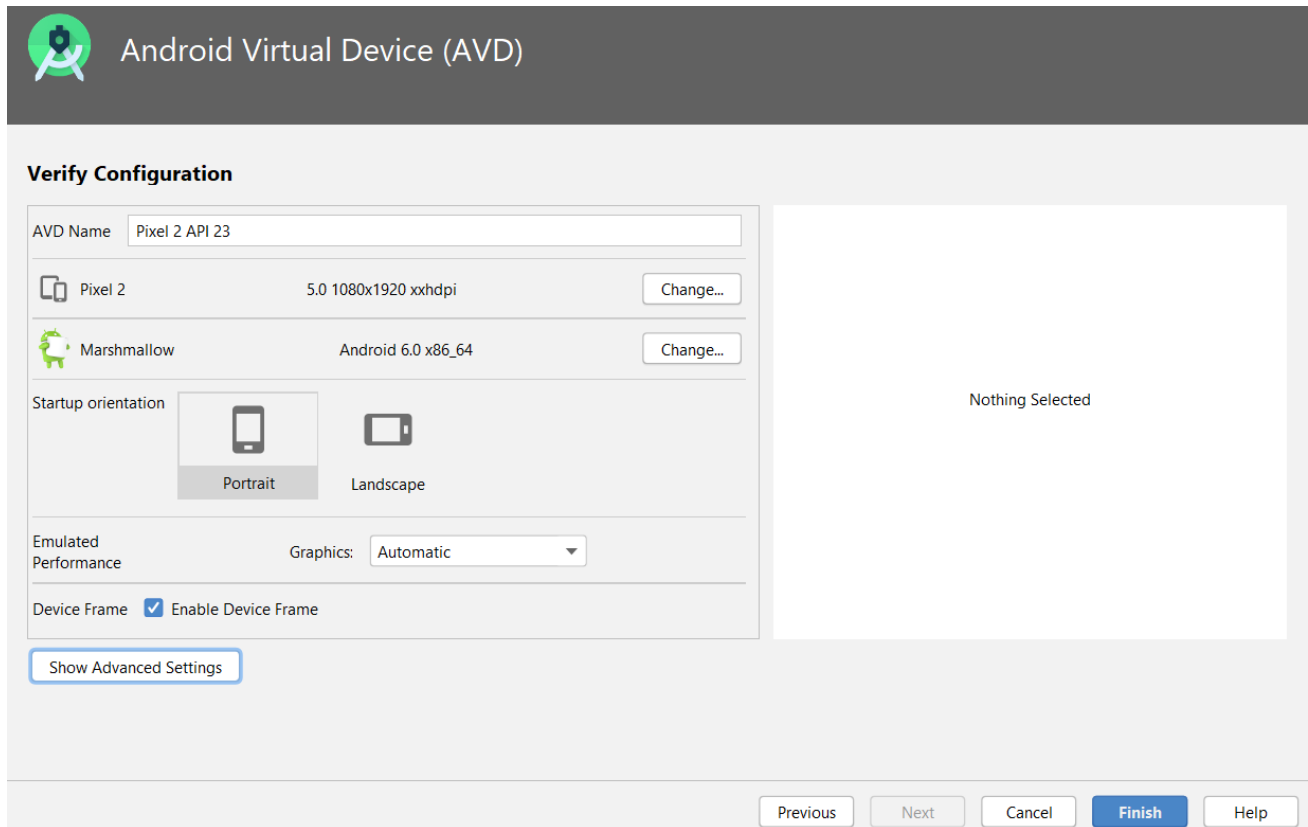
Google Inc.

System Image
x86_64

Questions on API level?
[See the API level distribution chart](#)

Previous **Next** Cancel Finish Help

Ahora puedes darle un nombre a tu dispositivo virtual y configurar algunas características adicionales.



7- Haz un clic en “**Show advanced settings**” (Mostrar opciones avanzadas)
Para configurar algunas opciones como:

Memoria Ram

Memoria interna

Memoria SD

Considera NO colocar mucha memoria pues será consumida de tu computadora cada vez que inicies el dispositivo virtual.

| | | |
|---|--|------------------------------------|
| RAM: | <input type="text" value="1536"/> | <input type="text" value="MB"/> |
| VM heap: | <input type="text" value="256"/> | <input type="text" value="MB"/> |
| Internal Storage: | <input type="text" value="800"/> | <input type="text" value="MB"/> |
| SD card: | <input checked="" type="radio"/> Studio-managed <input type="text" value="512"/> <input type="text" value="MB"/> | |
| | <input type="radio"/> External file <input type="text" value=""/> | |
| | <input type="radio"/> No SDCard | |
| <input checked="" type="checkbox"/> Enable Device Frame | | |
| Custom skin definition | <input type="text" value="pixel_2"/> | <input type="button" value="..."/> |

8- Selecciona el botón “**Finish**” (Finalizar) , espera unos segundos y regresaras a la ventana principal del AVD Manager.

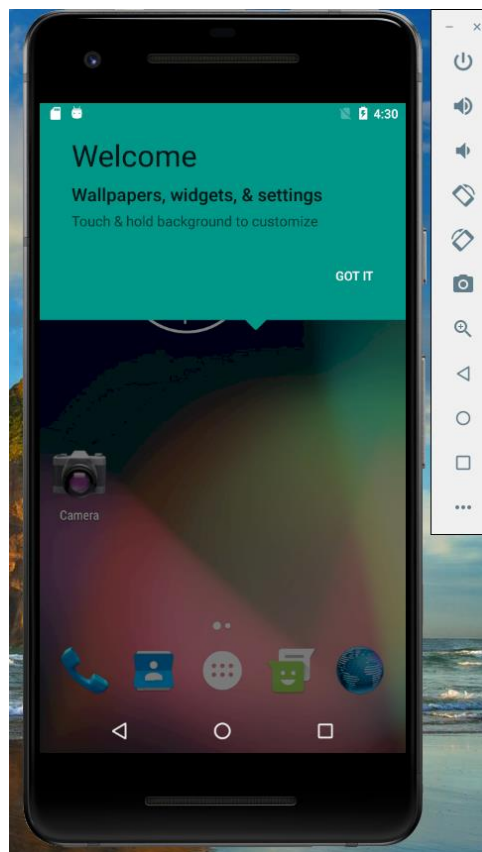
Ahora tu dispositivo virtual está listo.
selecciónalo y haz clic en el botón **play** para ejecutarlo.



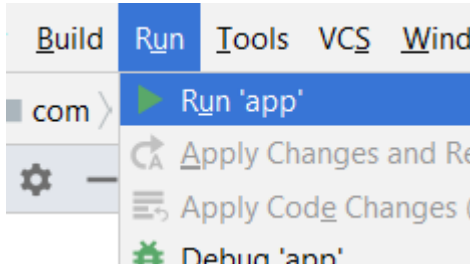
| | | | | | | | | |
|--|------------------------------|--|----------------|----|-------------------------|--------|--------|--|
| | NEXUS 5X API 20 | | 1080 x 1920... | 20 | Android 5.0 (Google ... | x86 | 5.0 GB | |
| | Pixel 2 API 23 | | 1080 x 1920... | 23 | Android 6.0 (Google ... | x86_64 | 2.5 GB | |
| | Pixel 2 API 23 - Marshmallow | | 1080 x 1920... | 23 | Android 6.0 (Google ... | x86_64 | 8.0 GB | |

Espera unos segundos y el dispositivo se mostrará en pantalla.
La primera vez que se ejecuta el dispositivo virtual puede tardarse un poco más, ten paciencia.

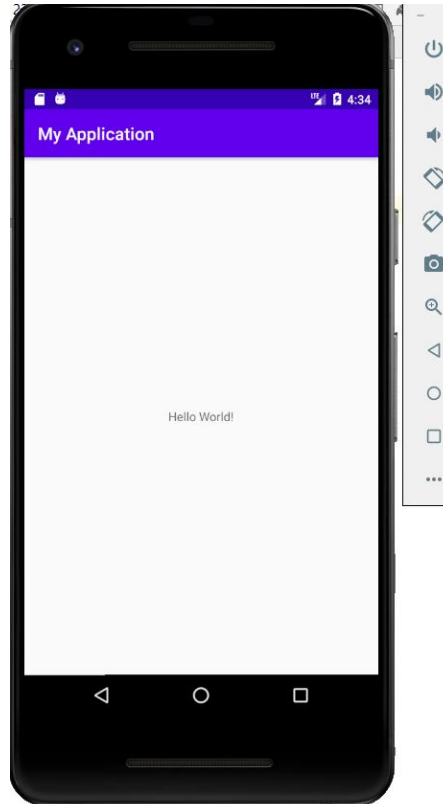
Puedes hacer clic en la opción “GOT IT” (para ocultar la pantalla de bienvenida)



Ahora puedes ejecutar tu aplicación en este dispositivo virtual.
Puede usar las opciones Run->Run 'app' del menú principal de Android.



La aplicación se ejecutará en el emulador como se muestra en la pantalla:



No olvides cerrar tu emulador antes de cerrar apagar tu equipo y cerrar Android Studio.

Importante:

Cuando quieras ejecutar la aplicación directamente en tu dispositivo físico, puedes usar el cable USB para conectarlo a tu computador (Considera que tu computador tenga los drivers adecuados para que reconozca a tu dispositivo físico)

Cuando el dispositivo este conectado puedes validar que Android Studio lo ha reconocido si se observa esto en la barra de herramientas:



Entonces cuando lo ejecutes, la aplicación se instalará y ejecutara en tu dispositivo físico.

IV. DISCUSION DE RESULTADOS

1. Crear un emulador con las siguientes características:
 - a. Hardware: pixel 3, tamaño de pantalla: mayor o igual a 5", memoria RAM: 500 MB, memoria interna: 7500 MB
 - b. Sistema operativo: Android Lollipop.
2. Crea otro proyecto llamado: Segunda App
 - a. SDK mínimo: API 20
 - b. Plantilla: **Empty Activity**
 - c. Cambia el mensaje "**Hello World**" por "<Tu nombre completo>"
 - d. Ejecuta la aplicación en el emulador creado en el paso 1

V. BIBLIOGRAFIA

- Desarrollo de aplicaciones para Android 2016 / Android application development, Anaya Multimedia
- El gran libro de Android: 7ª Edición (Jesús Tomás Gironés)