

# Guia de instalacion y ejecucion de las herramientas para la utilización de la aplicación

<b>Servidor</b>	<b>2</b>
Introducción	2
Instalar Docker y correr el contenedor	2
<b>Aplicación móvil</b>	<b>7</b>
Introducción	7
Subíndice	7
Instalación y uso del IDE Android Studio	8
1.1 Descarga Android Studio	8
1.2 Proceso de instalación	8
Instalación de un emulador de Android Studio	9
2.1 Abrir el AVD Manager	9
2.1.1 Abrir desde la pantalla de inicio	10
2.1.2 Abrir el AVD Manager desde la pantalla principal del IDE	11
2.2 Cómo crear un AVD	12
2.3 Ejecutar un dispositivo virtual	15

# Servidor

## Introducción

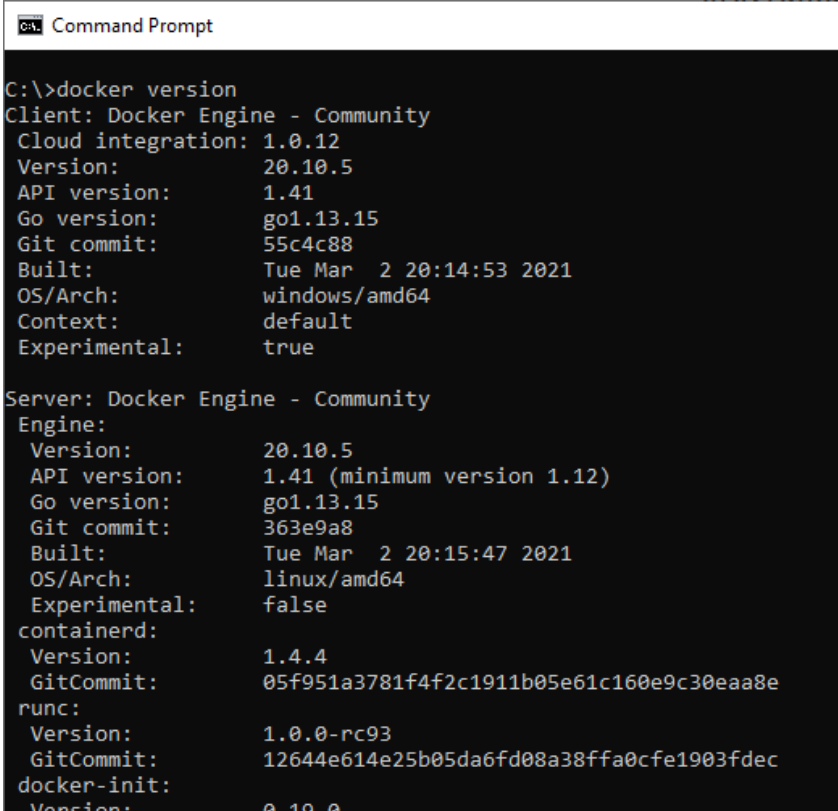
La aplicación desde el lado del servidor incluye un archivo Dockerfile el cuál es un documento de texto que contiene un conjunto de comandos o instrucciones que se ejecutan sucesivamente para generar una imagen en particular con todas las dependencias necesarias.

## Instalar Docker y correr el contenedor

El primer paso consiste en descargar e instalar Docker Desktop, disponible en la página oficial ([Docker Desktop](#)), seleccionando el sistema operativo que se dispone.

Para verificar que Docker se esté ejecutando correctamente en el equipo se puede correr el siguiente comando:

- `docker version`



```

C:\>docker version
Client: Docker Engine - Community
 Cloud integration: 1.0.12
 Version:           20.10.5
 API version:       1.41
 Go version:        go1.13.15
 Git commit:        55c4c88
 Built:             Tue Mar  2 20:14:53 2021
 OS/Arch:           windows/amd64
 Context:           default
 Experimental:      true

Server: Docker Engine - Community
 Engine:
  Version:           20.10.5
  API version:       1.41 (minimum version 1.12)
  Go version:        go1.13.15
  Git commit:        363e9a8
  Built:             Tue Mar  2 20:15:47 2021
  OS/Arch:           linux/amd64
  Experimental:      false
 containerd:
  Version:           1.4.4
  GitCommit:        05f951a3781f4f2c1911b05e61c160e9c30eaa8e
 runc:
  Version:           1.0.0-rc93
  GitCommit:        12644e614e25b05da6fd08a38ffa0cfe1903fdec
 docker-init:
  Version:           0.19.0
```

Una vez instalado Docker Desktop se puede proceder con la generación de la imagen, para ello es necesario ejecutar el comando en la carpeta donde se encuentra el Dockerfile (Dentro de la carpeta Servidor).

- `docker build -t colectivos_api_image .`

Este comando toma un archivo Dockerfile y lo transforma en una imagen, creando las capas necesarias y guardandolas en el equipo local con la etiqueta definida con el parámetro `-t`.

```
Command Prompt
C:\ProyectoFinal\Servidor\PredictorTiempoColectivo>docker build -t colectivos_api_image .
[+] Building 0.5s (24/24) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 32B
=> [internal] load .dockerignore
=> => transferring context: 35B
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:5.0-buster-slim
=> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:5.0-buster-slim
=> [build 1/13] FROM mcr.microsoft.com/dotnet/sdk:5.0-buster-slim
=> [internal] load build context
=> => transferring context: 8.36kB
=> [base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:5.0-buster-slim
=> CACHED [base 2/2] WORKDIR /app
=> CACHED [final 1/2] WORKDIR /app
=> CACHED [build 2/13] WORKDIR /src
=> CACHED [build 3/13] COPY [API/Api.csproj, API/]
=> CACHED [build 4/13] COPY [Application/Services/Services.csproj, Application/Services/]
=> CACHED [build 5/13] COPY [Infrastructure/SQLRepository/SQLRepository.csproj, Infrastructure/SQLRepository/]
=> CACHED [build 6/13] COPY [Domain/Domain.csproj, Domain/]
=> CACHED [build 7/13] COPY [Common/Helper.csproj, Common/]
=> CACHED [build 8/13] COPY [Application/Regresion/Regresion.csproj, Application/Regresion/]
=> CACHED [build 9/13] COPY [Infrastructure/FileRepository/FileRepository.csproj, Infrastructure/FileRepository/]
=> CACHED [build 10/13] RUN dotnet restore "API/Api.csproj"
=> CACHED [build 11/13] COPY . .
=> CACHED [build 12/13] WORKDIR /src/API/
=> CACHED [build 13/13] RUN dotnet build "Api.csproj" -c Release -o /app/build
=> CACHED [publish 1/1] RUN dotnet publish "Api.csproj" -c Release -o /app/publish
=> CACHED [final 2/2] COPY --from=publish /app/publish .
=> exporting to image
=> => exporting layers
=> => writing image sha256:5e2aed46decf52d19c20ded4ce7c207c4ba14e1b96152ea5c344c963746648c9
=> => naming to docker.io/library/colectivos_api_image

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

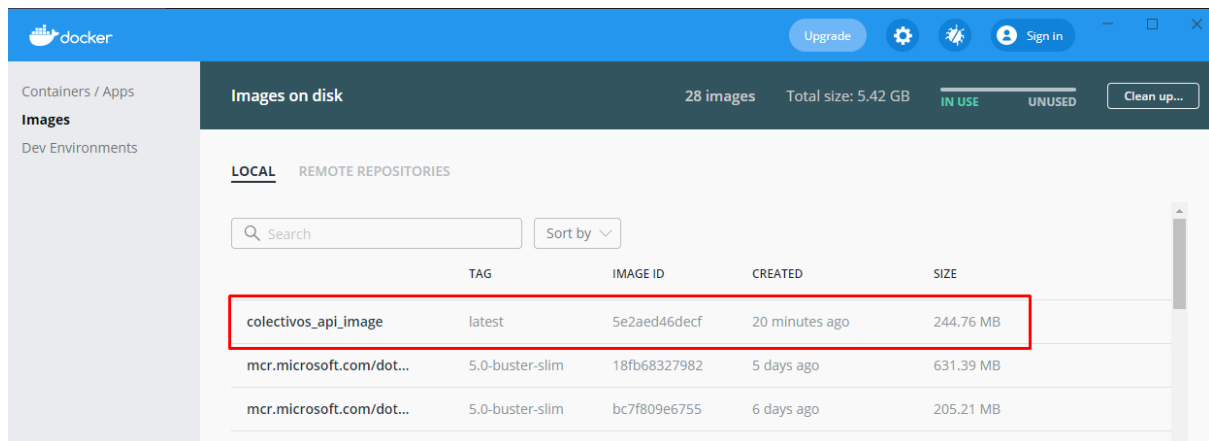
C:\ProyectoFinal\Servidor\PredictorTiempoColectivo>
```

Para validar si la imagen fue creada correctamente ejecutar el siguiente comando, qué lista las imágenes construidas y disponibles en el sistema:

- `docker images`

```
Command Prompt
C:\ProyectoFinal\Servidor\PredictorTiempoColectivo>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
colectivos_api_image latest              5e2aed46decf        20 minutes ago     245MB
mcr.microsoft.com/dotnet/sdk 5.0-buster-slim    18fb68327982        5 days ago         631MB
mcr.microsoft.com/dotnet/aspnet 5.0-buster-slim    bc7f809e6755        5 days ago         205MB
```

O sino desde la aplicación de Docker Desktop:



En caso de que la imagen se haya generado correctamente, se ejecuta el comando

- `docker run -d -p 8080:80 -d --name colectivo_api_container colectivos_api_image`

Dicho comando crea un contenedor a partir de una imagen existente en el sistema. El nombre del contenedor es `colectivo_api_container` (el cuál puede ser elegido a gusto, pero en caso que no se defina un nombre Docker asignará un nombre aleatorio al contenedor), se redirigirá todo el tráfico del puerto 8080 del anfitrión al puerto 80 del contenedor ( `-p <puerto-host>:<puerto-contenedor>`). El flag `-d` (detached) es para que la terminal no quede capturada por la ejecución del contenedor. Si no se especifica `-d`, la terminal queda inhabilitada hasta que se envíe un comando de terminar como `Ctrl-C`.

```

C:\ProyectoFinal\Servidor\PredictorTiempoColectivo>docker run -d -p 8080:80 -d --name colectivo_api_container colectivos_api_image
377ffd53b1a98525a0f2ac02128f75a58b659abc6920c8c0fa36072e5d72e76c
C:\ProyectoFinal\Servidor\PredictorTiempoColectivo>
  
```

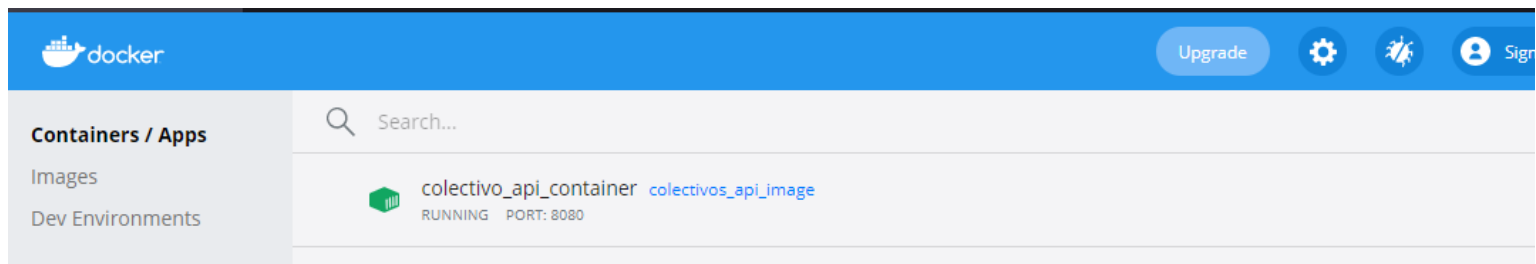
Para visualizar el contenedor funcionando se puede correr el siguiente comando donde se muestra los contenedores ejecutándose o detenidos en el sistema:

- `docker ps`

```

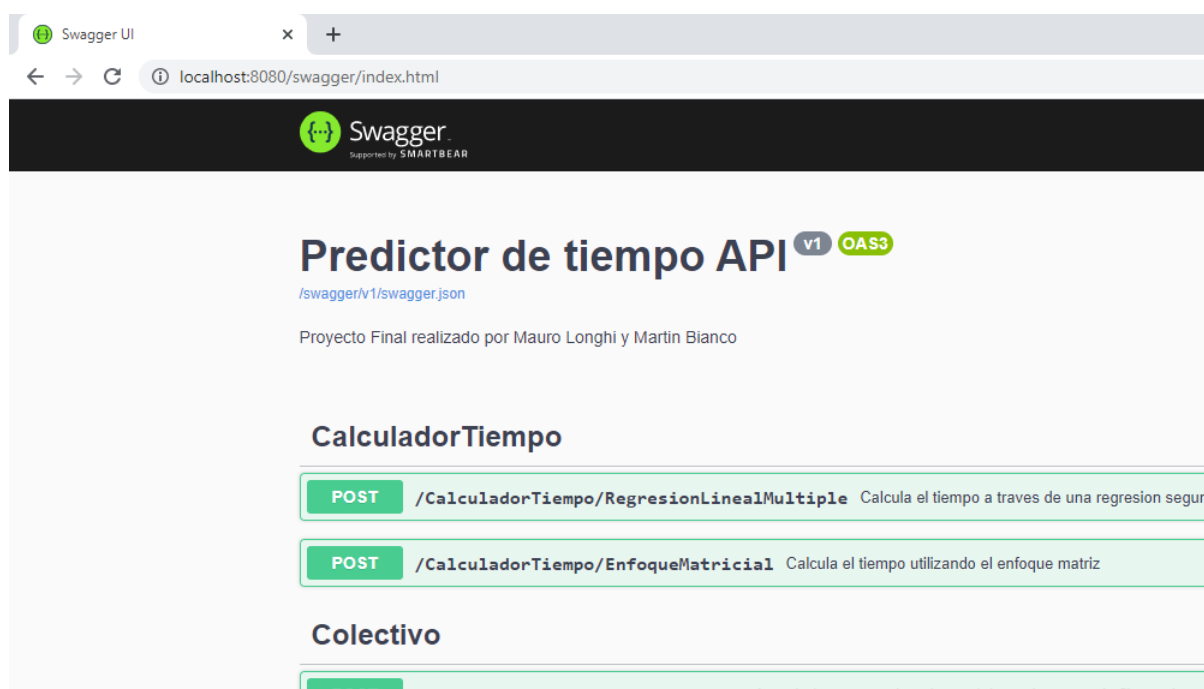
C:\ProyectoFinal\Servidor\PredictorTiempoColectivo>docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                               NAMES
377ffd53b1a9   colectivos_api_image "dotnet Api.dll"        59 seconds ago Up 55 seconds  443/tcp, 0.0.0.0:8080->80/tcp      colectivo_api_container
C:\ProyectoFinal\Servidor\PredictorTiempoColectivo>
  
```

O desde la aplicación de Docker Desktop



Para probar que nuestro servicio está activo, abrimos en un navegador la dirección:

- <http://localhost:8080/swagger/index.html>



# Aplicación móvil

## Introducción

El siguiente documento incluye la descripción de los pasos para la configuración del entorno de desarrollo y la creación y ejecución de la aplicación móvil desarrollada para el proyecto final.

El objetivo de este apartado es facilitar el uso de las diferentes herramientas desarrolladas y así permitir la prueba y depuración de dichas herramientas sin necesidad de un dispositivo físico, accesible directamente desde el editor de Android Studio.

## Subíndice

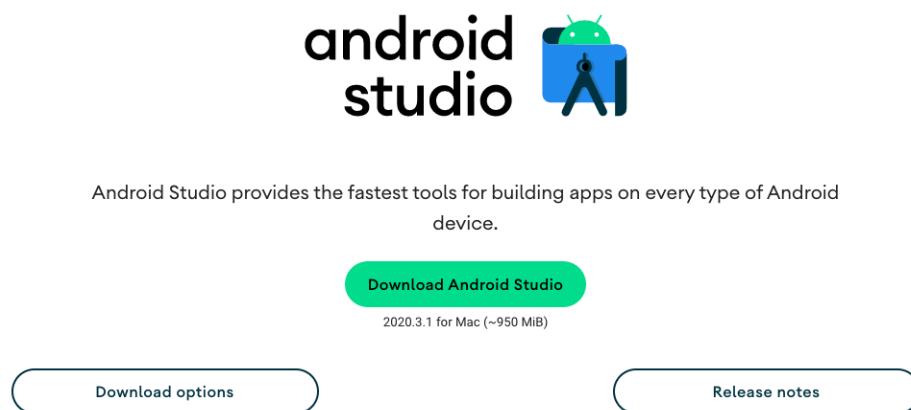
- 1 - Guía de instalación y uso del IDE Android Studio.
- 2 - Guía de instalación de un emulador de Android Studio.

# 1. Instalación y uso del IDE Android Studio

A continuación se describirán los pasos para la instalación del entorno de desarrollo integrado oficial para el desarrollo de apps para Android, Android Studio ([introducción a Android Studio](#))

## 1.1 Descarga Android Studio

El primer paso es descargar el instalador, disponible oficialmente para Windows, Mac y Linux, mediante su enlace de descarga en su [web oficial](#), de acuerdo con la versión del sistema operativo que contenga.



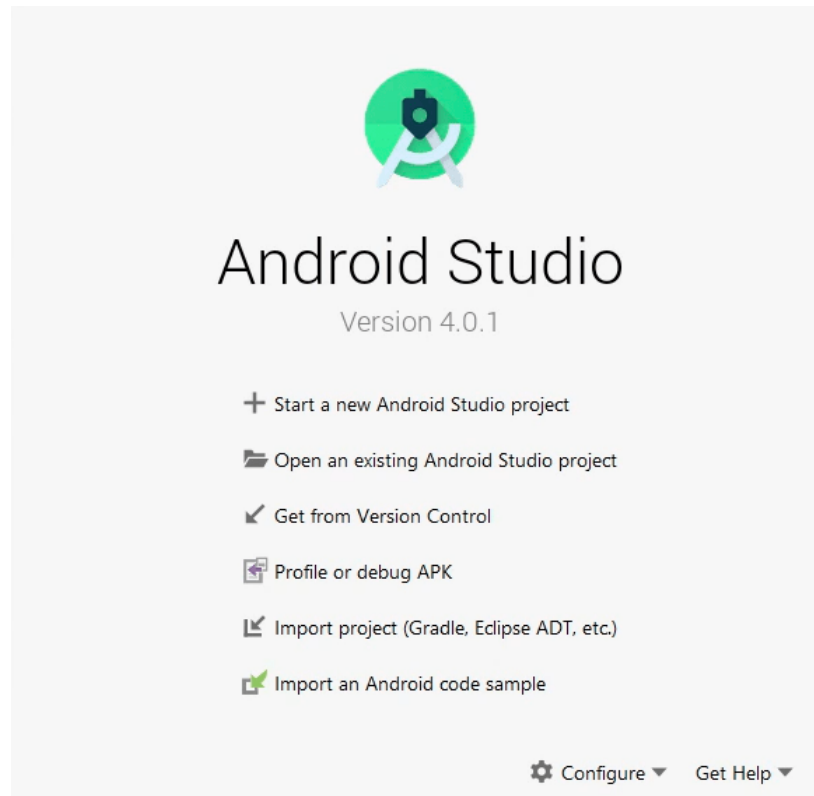
Una vez descargado, comenzaremos el proceso de instalación, ejecutando el archivo ejecutable descargado.

## 1.2 Proceso de instalación

Una vez ejecutado el archivo descargado, simplemente seguir los pasos del asistente de configuración en Android Studio, asegurando la instalación de los paquetes del SDK que recomienda.

[Proceso de instalación para Windows, Mac, Linux](#)

Luego de haberse instalado correctamente todos los paquetes, podremos seleccionar por defecto iniciar el entorno de desarrollo y así ir a la pantalla inicial para abrir el proyecto desarrollado y realizar las pruebas correspondientes.



## 2. Instalación de un emulador de Android Studio

Android Studio permite la prueba y depuración de nuestras aplicaciones sin necesidad de un dispositivo físico. Esta característica será accesible directamente desde el editor y así comprobar el funcionamiento de las aplicaciones desarrolladas, a través de la [creación de un dispositivo virtual](#) (AVD o Android Virtual Device).

Un emulador o dispositivo, es un software que permite ejecutar otras arquitecturas en una plataforma diferente a su original, ya sea con otra arquitectura hardware o sistema operativo. Permite reproducir de una manera muy precisa el comportamiento de un programa, al igual que se realizaría en el dispositivo para el que fue diseñado y así poder probar una aplicación en diferentes dispositivos y niveles de API de Android.

### 2.1 Abrir el AVD Manager

Para abrir el Administrador de dispositivos virtuales de Android (AVD), se puede acceder desde dos puntos dentro del entorno de desarrollo (IDE) Android Studio.

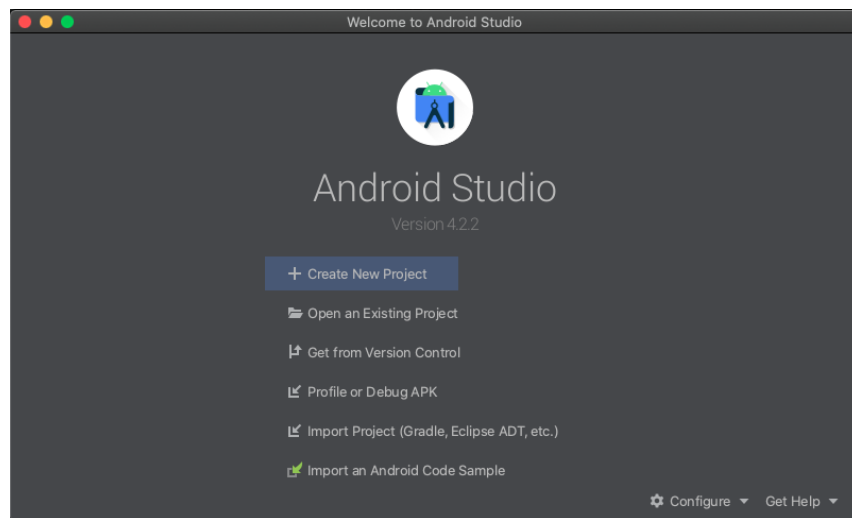



- 1) desde la pantalla de inicio del IDE, accediendo al AVD Manager.
- 2) dentro de la pantalla principal, en la barra de herramientas, accediendo al AVD Manager.

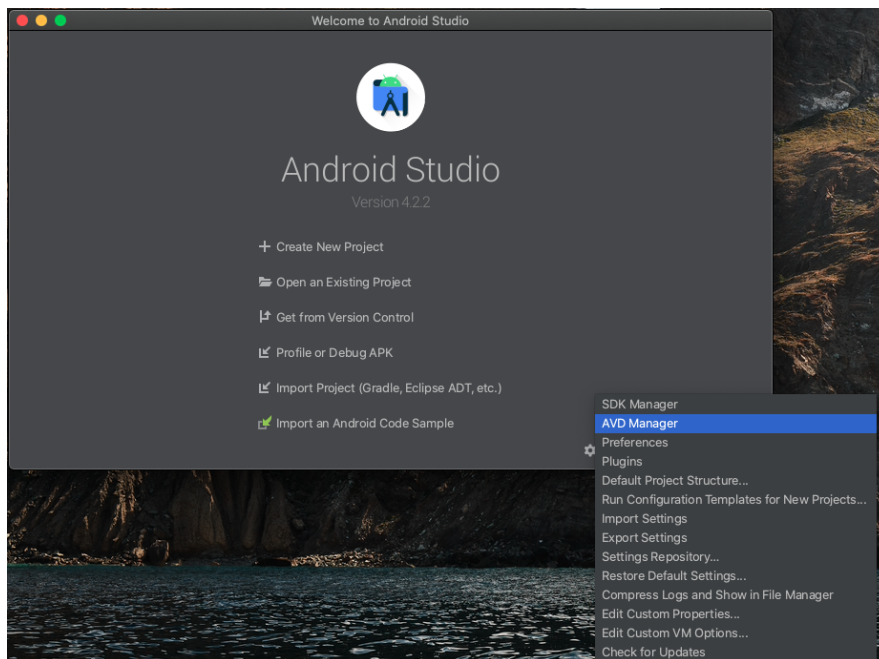
## 2.1.1 Abrir desde la pantalla de inicio

Para abrir el Administrador de AVD desde la pantalla de inicio, realiza una de las siguientes acciones:

- abrir el IDE para ir a la pantalla inicial.



- Selecciona la opción de Configuración  **Configure** , ubicado en la parte inferior derecha de la ventana.
- Selecciona la opción AVD Manager.

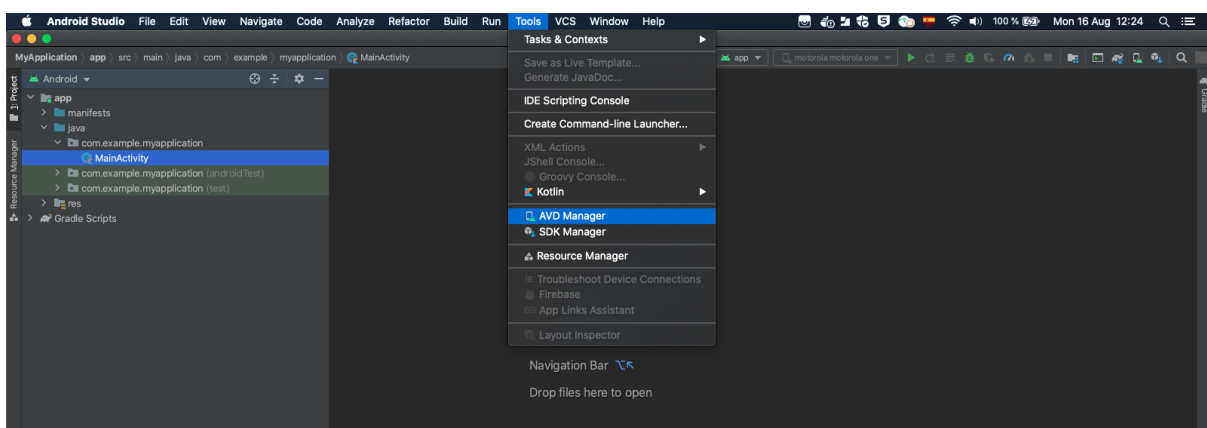


## 2.1.2 Abrir el AVD Manager desde la pantalla principal del IDE

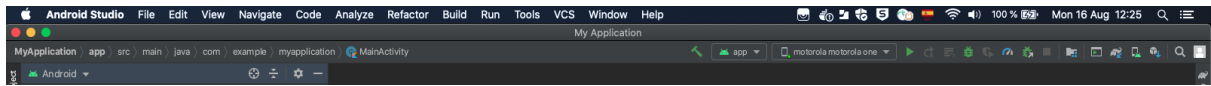
- Selecciona **Tools > AVD Manager**.



- En la barra de herramientas, haz clic en **AVD Manager**.



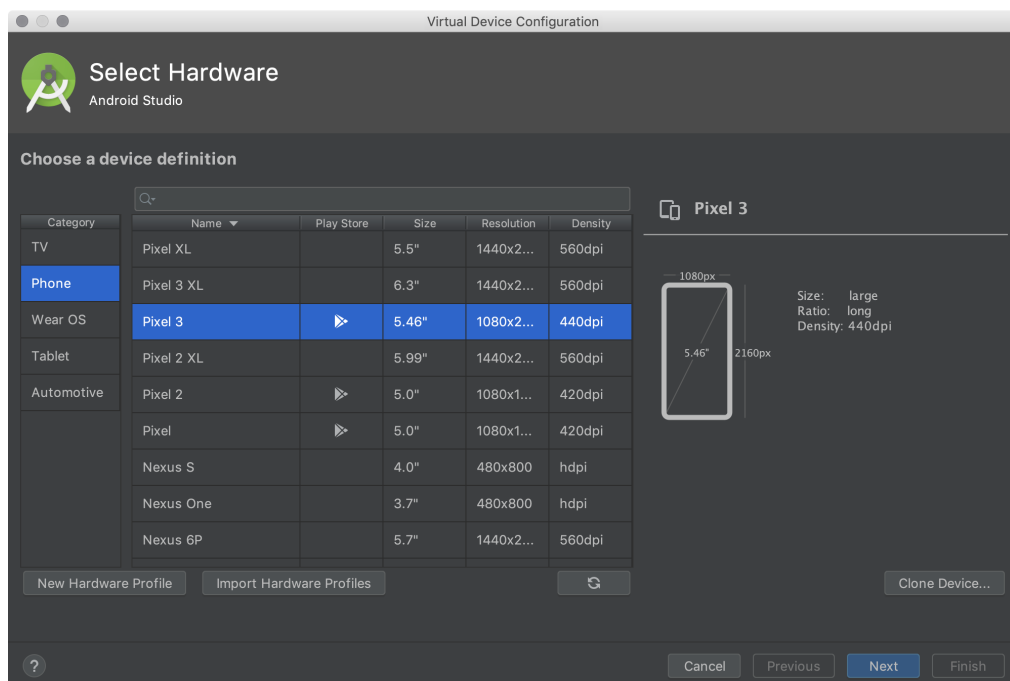
- Alternativamente, se encuentra el icono del AVD Manager dentro de la barra de herramientas visible en la parte superior derecha del IDE.




## 2.2 Cómo crear un AVD

Desde el AVD Manager, vamos a seguir los siguientes pasos:

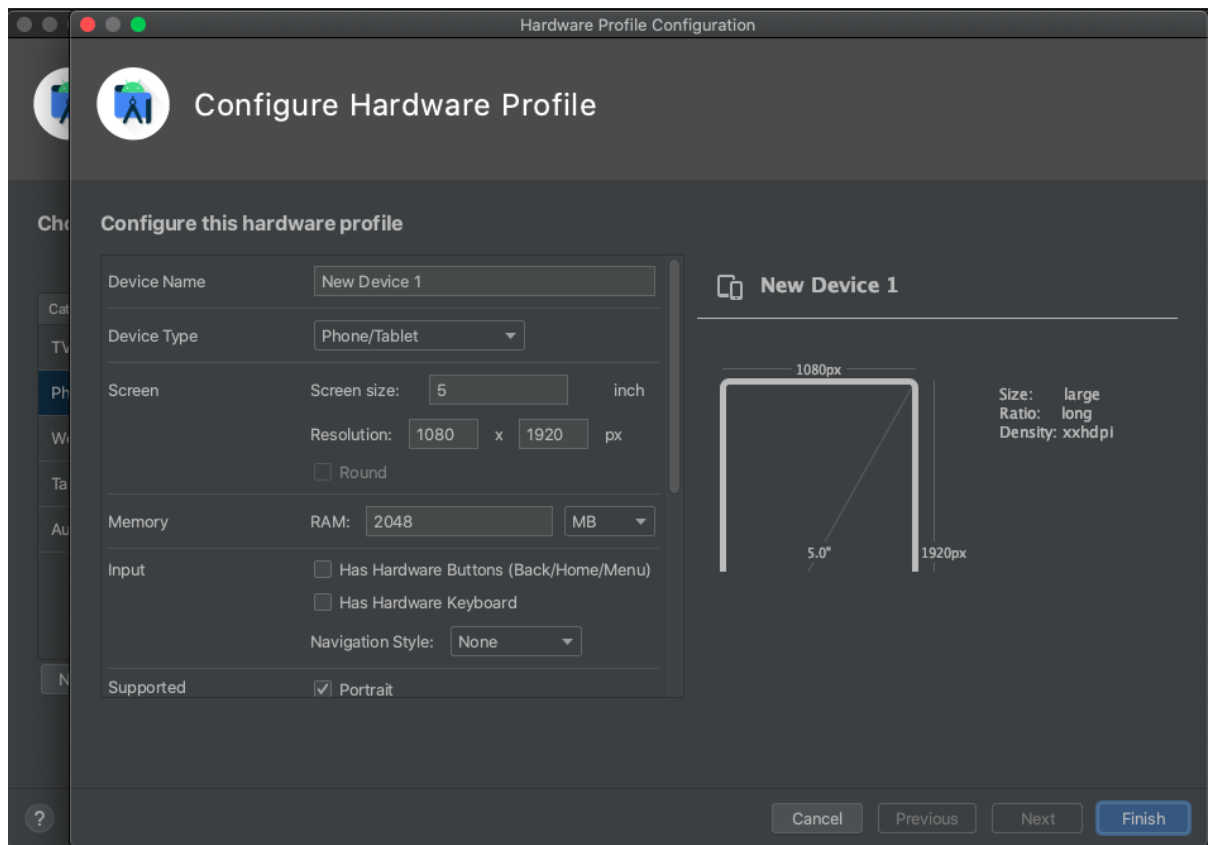
- 1) Seleccionar **Create Virtual Device**, en la parte inferior del diálogo del Administrador de AVD, donde aparecerá la página **Select Hardware**



Cabe destacar que los diferentes modelos y propiedades que nos ofrece el IDE, corresponden a los provistos por Google. Además es necesario que el tipo de

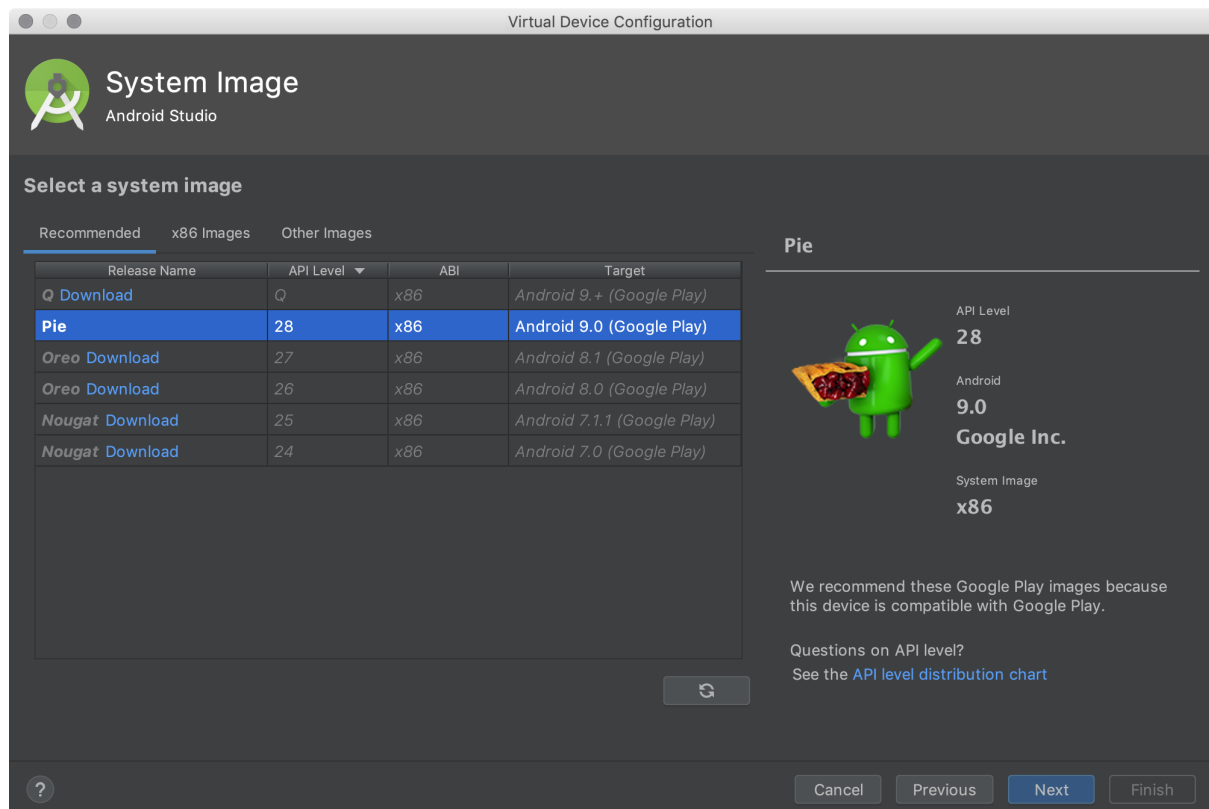
hardware elegido tenga la opción de Play Store  en la columna propiamente dicha, debido a que consumiremos servicios propios de google maps.

Por otro lado, si quisiéramos realizar alguna modificación en el hardware base que nos proveen cada una de las opciones anteriores, podemos configurarlas seleccionando la opción **New Hardware Profile**, brindándonos mayor capacidad de RAM, diferentes resoluciones, entre otras, manteniendo como base el modelo elegido en el paso previo.



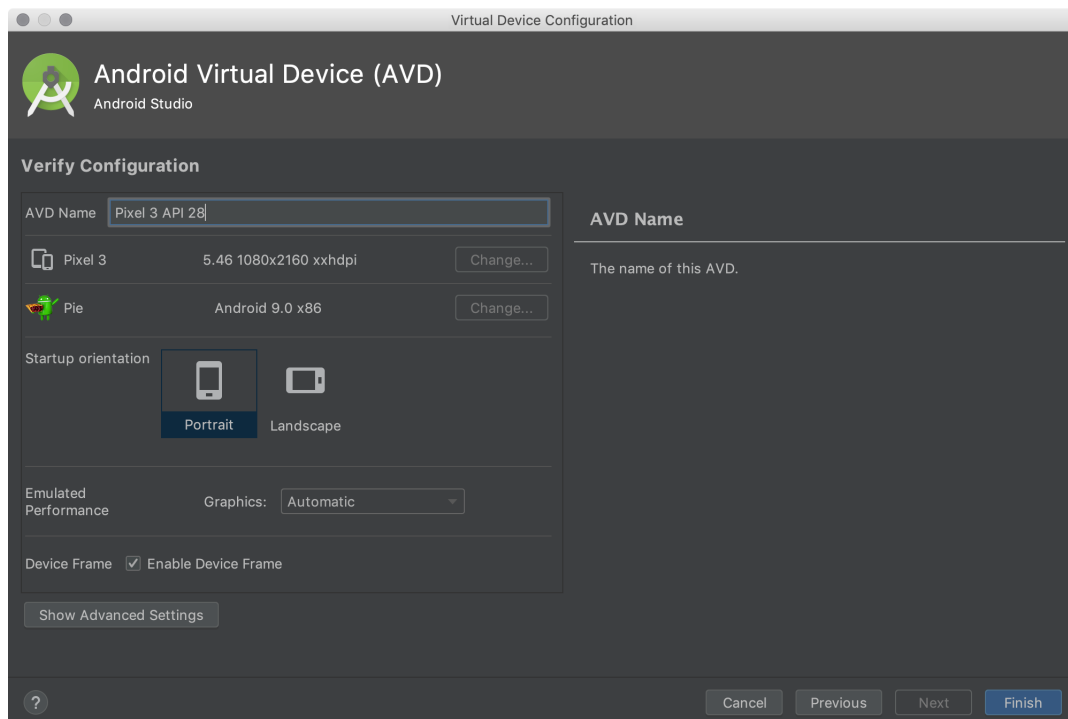
2) Selecciona un perfil de hardware y haz clic en **Next**.

En este punto, podemos seleccionar el archivo de imagen para el sistema operativo, desde las versiones más modernas a las más antiguas. Al seleccionar la imagen, el nivel de API del dispositivo de destino es importante, ya que no podrá ejecutarse la app en una imagen del sistema con un nivel de API inferior al que se necesita, según se especifica en el atributo [minSdkVersion](#) del archivo de manifiesto de la app (minSdkVersion = 26).

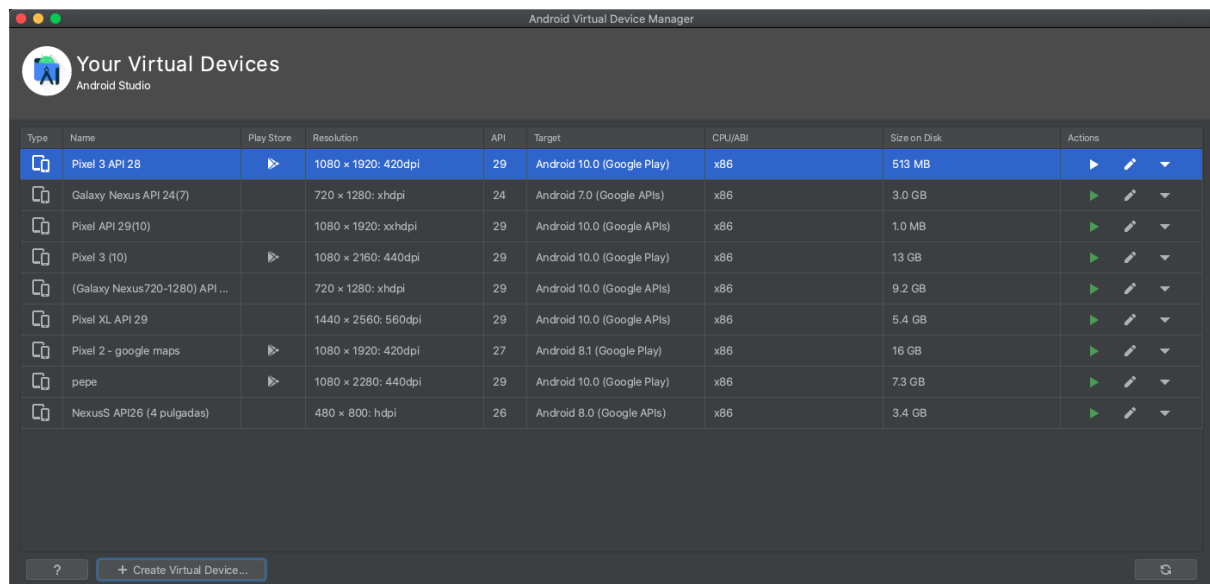


4) Selecciona la imagen del sistema para un nivel de API determinado y haz clic en **Next**.

Por último, aparecerá la página **Verify Configuration**, donde se encuentran los detalles como el nombre del dispositivo virtual creado y otras características que no tendremos en cuenta.



5) Al hacer clic en Finish, el nuevo AVD aparecerá en la página de **Your Virtual Devices** o en el diálogo **Select Deployment Target**.



## 2.3 Ejecutar un dispositivo virtual

Por último para poder correr el dispositivo virtual, solo tenemos que seleccionar el dispositivo virtual creado en el AVD Manager, y en la columna de

**Actions** seleccionar la opción de **Run** , para que el IDE ejecute y muestre el emulador o dispositivo creado

