

**MP\_0489. Programación  
multimedia y dispositivos móviles**

**UF1. Análisis de tecnologías para  
aplicaciones en dispositivos móviles**

**1.3. Entornos de desarrollo  
y emuladores en Android**

# Índice

---

☰	Objetivos	3
☰	IDE o Entorno de Desarrollo Integrado	4
☰	Android Studio	6
☰	Crear un emulador	23
☰	Resumen	32

## Objetivos

---

**En esta unidad nuestros objetivos principales serán:**

1

Conocer los diferentes IDEs para el desarrollo de apps en dispositivos móviles.

2

Conocer los emuladores que nos van a servir para verificar nuestra aplicación.

3

Aprender a instalar y configurar Android Studio como entorno de desarrollo para aplicaciones Android.

---

¡Ánimo y adelante!

# IDE o Entorno de Desarrollo Integrado

**Desarrollar una app con las herramientas necesarias no solamente es cómodo, sino que evita que perdamos el tiempo en detalles.**

A lo largo de esta apartado nos centraremos en cómo se desarrolla una app dentro de un IDE.

## ¿Qué es un IDE?

Un [Entorno de Desarrollo Integrado](#) (*Integrated Development Environment*) es una aplicación informática que contiene un kit de herramientas para facilitar a los desarrolladores el desarrollo de software.

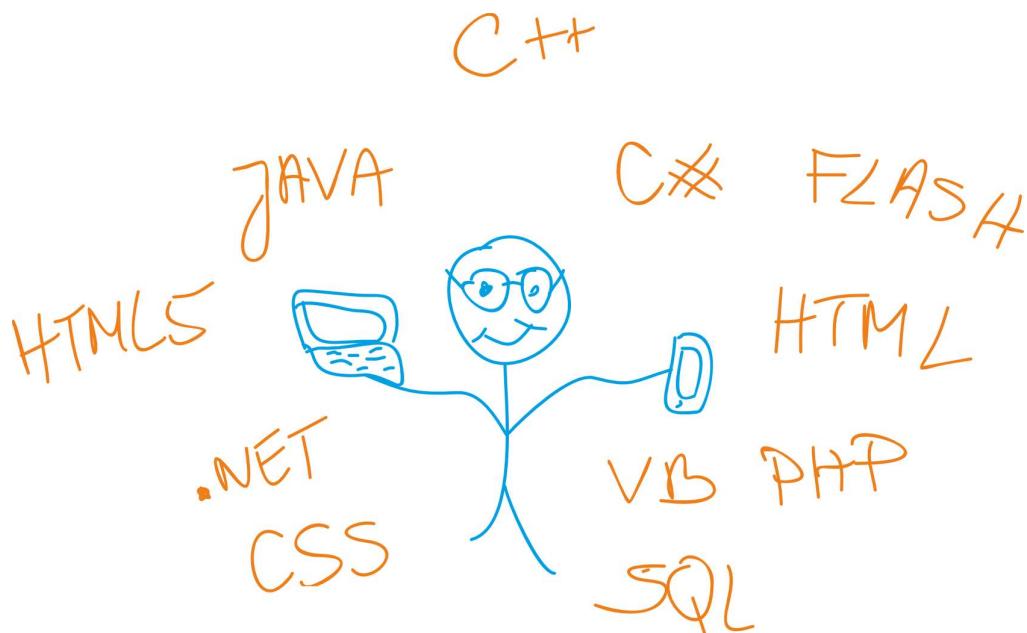
Con un IDE, las tareas comunes como crear paquetes, ficheros, administrar recursos, dependencias, compilar, depurar, etc. se vuelven más fáciles de hacer. Todo nuestro código es más sencillo de entender, los comentarios se encuentran rápidamente y, con el uso de patrones como [MVC](#), podemos saber si estamos en una Vista o en el Controlador.



Recuerda que la vista podría ser, por ejemplo, la parte visual UI (*User Interface*) y el controlador la parte que conecta el Modelo con la Vista.

Nuestro IDE nos proporciona una manera rápida y eficaz para cambiar entre los diferentes ficheros (*Clases, Layouts, Recursos, etc.*) de nuestro proyecto.

Los IDE están diseñados para **maximizar la productividad del programador**, proporcionando **componentes muy unidos con interfaces de usuario similares**. Presentan un único programa, en el que se lleva a cabo todo el desarrollo. Generalmente, este programa suele ofrecer muchas características para la creación, modificación, compilación, implementación y depuración de software.



La mayoría de los IDE tienen autocompletado inteligente de código (*IntelliSense*).

Esto contrasta con el desarrollo de software utilizando herramientas no relacionadas, como Vi, GNU Compiler Collection (GCC) o Make.

#### Entorno de desarrollo integrado1

Pulsa el link para saber más sobre el entorno de desarrollo integrado.

[PULSA AQUÍ](#)

# Android Studio

---

Android Studio es el **IDE oficial de Google en el desarrollo de aplicaciones para Android**.

---

Android Studio soporta Java 8 y el lenguaje Kotlin.

---

## Instalar Android Studio

Para instalar Android Studio necesitamos descargarnos el fichero de instalación de la web <https://developer.android.com/studio/> y seguir los pasos que te mostraremos a continuación.

- (i)** ¡Ojo! Para poder ejecutar Android Studio en nuestro ordenador **necesitamos cumplir con unos requisitos mínimos**. La cantidad de memoria RAM recomendado es de 8 GB, nuestro procesador debe tener capacidades para virtualizar (*emulador*) y, además, tener instalado Java 8.

The screenshot shows a Google search results page. The search query is "Requisitos Instalación Android Studio". The results include a snippet titled "Requisitos Android Studio" which lists the minimum requirements for Android Studio. Below this, there is a link to "Android Studio: Requisitos mínimos - Android Studio FAQs" with the URL <https://androidstudiofaqs.com/conceptos/android-studio-requisitos-minimos>. At the bottom of the snippet, there is a "Search for: Requisitos Android Studio" button.

Requisitos Android Studio

**Requisitos mínimos de Android Studio**

- **Windows** 7/8/10 (32 o 64 bits).
- 2 GB de RAM (8 GB de RAM recomendado).
- 2 GB de espacio libre mínimo (4 GB recomendado).
- Resolución mínima de 1.280 x 800.
- Java 8.
- 64 bits y procesador Intel (emulador).

[Android Studio: Requisitos mínimos - Android Studio FAQs](https://androidstudiofaqs.com/conceptos/android-studio-requisitos-minimos)  
<https://androidstudiofaqs.com/conceptos/android-studio-requisitos-minimos>

Search for: Requisitos Android Studio

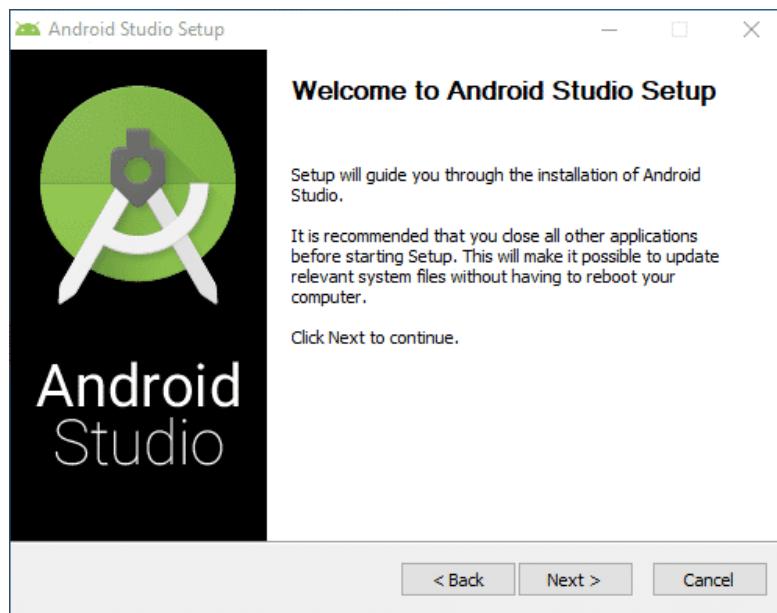
Debemos instalar el **JDK** (*Java Development Kit*) de **Oracle** antes de instalar **Android Studio**.

## Instalación de Android Studio

Vamos a ver paso a paso cómo instalar Android Studio en nuestro ordenador.

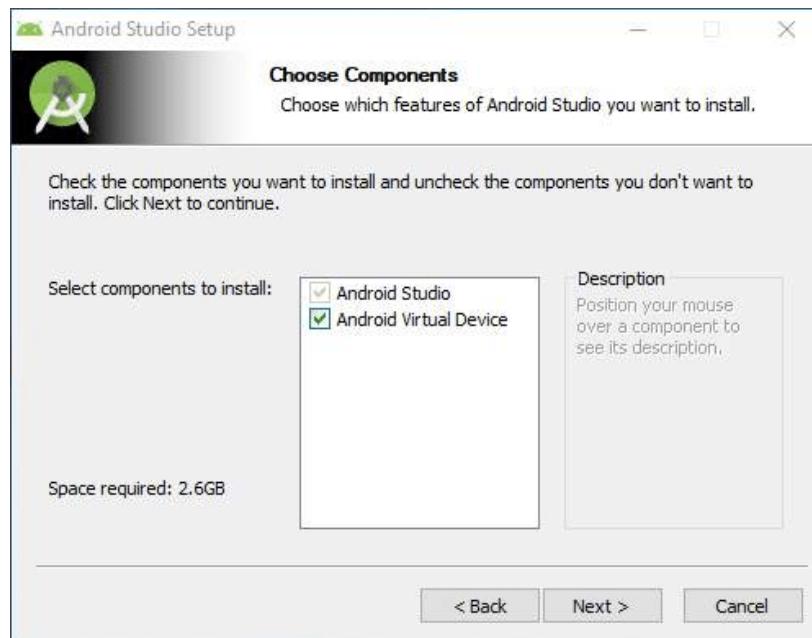
Podemos descargarlo en la página oficial: <https://developer.android.com/studio/>

## Paso 1



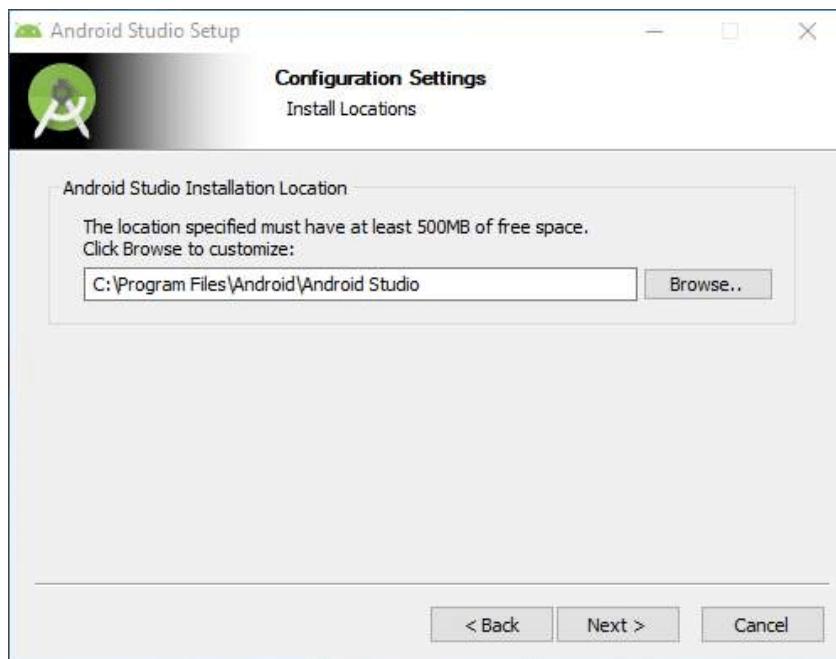
Una vez descargado, empezamos con la instalación.

## Paso 2



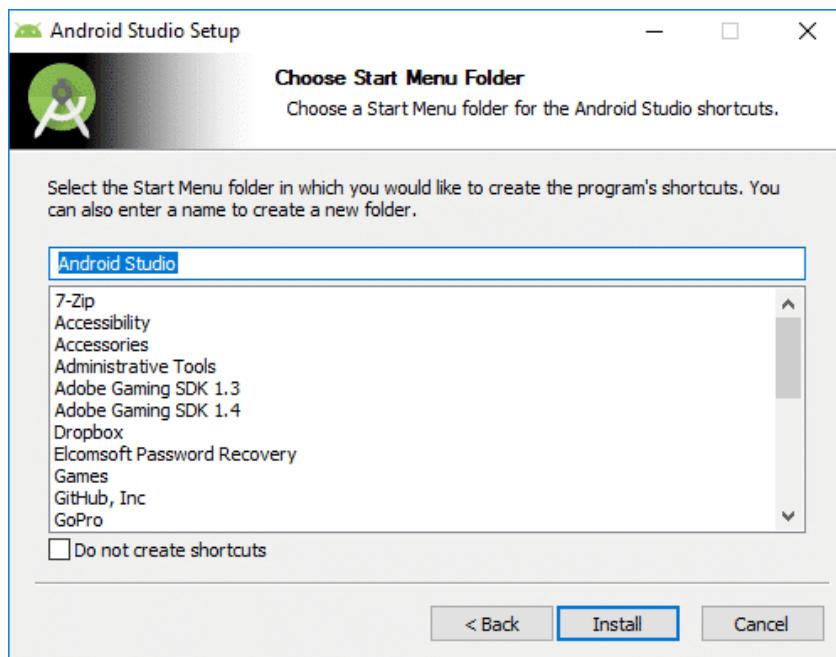
En este paso debemos elegir qué componentes queremos. Se nos ofrece como opción el Android Virtual Device, que instalaremos, dado que es el emulador para comprobar nuestro desarrollo.

## Paso 3



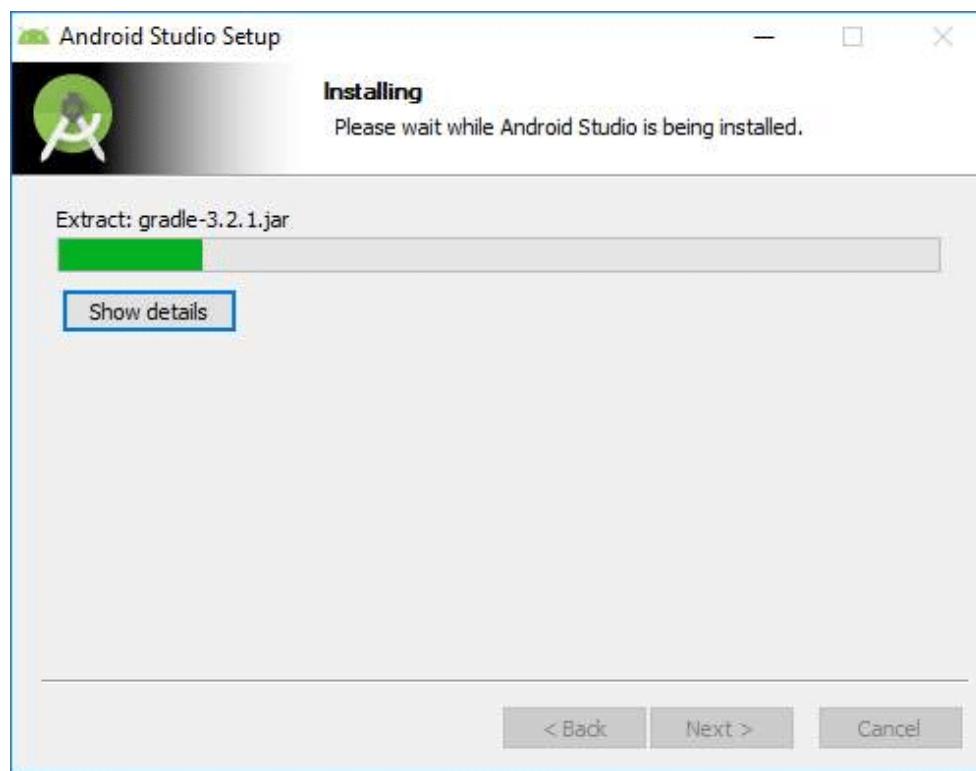
Elegimos el lugar donde se instalará Android Studio.

## Paso 4



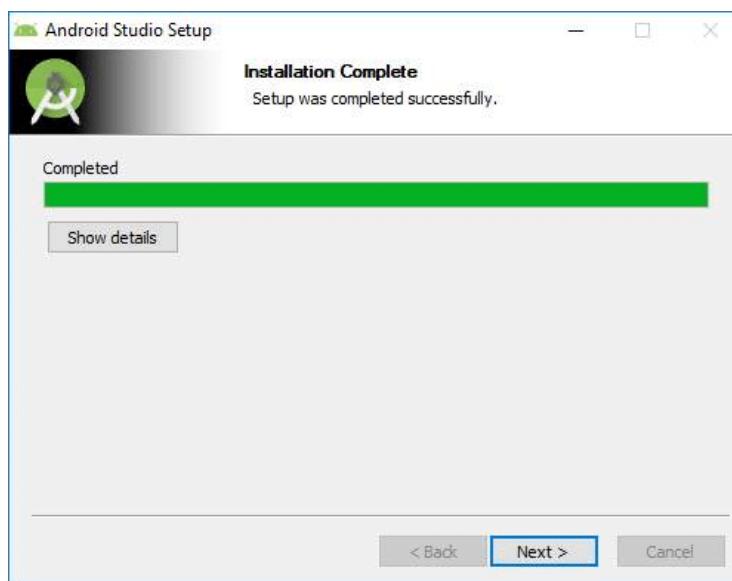
Continuamos.

## Paso 5



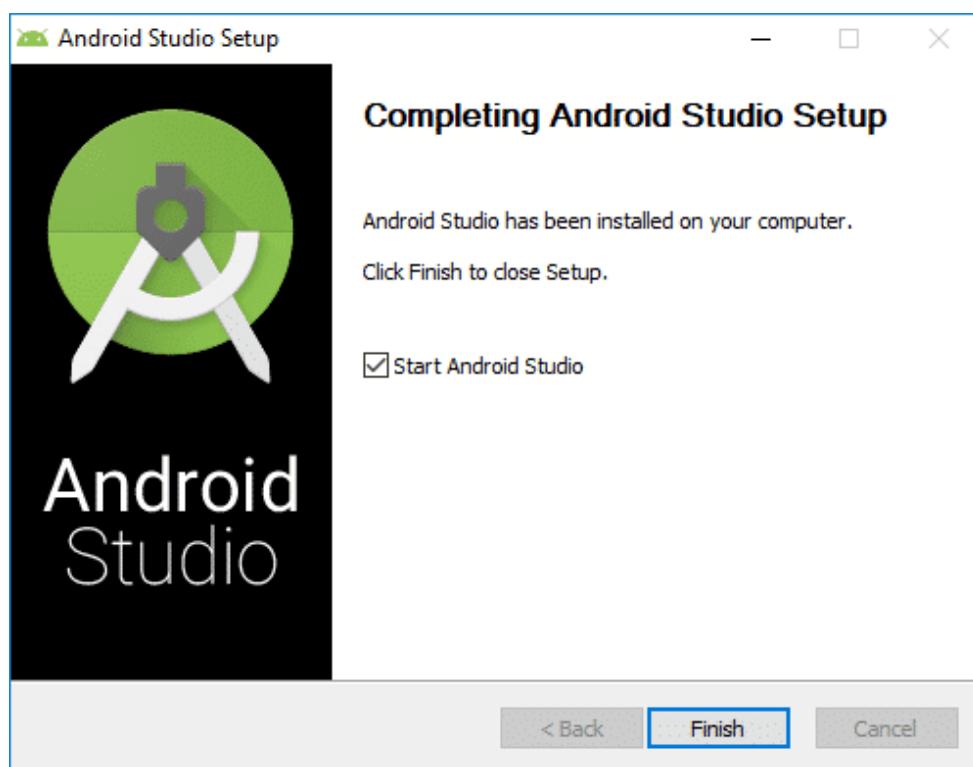
Esperamos mientras se instala.

## Paso 6



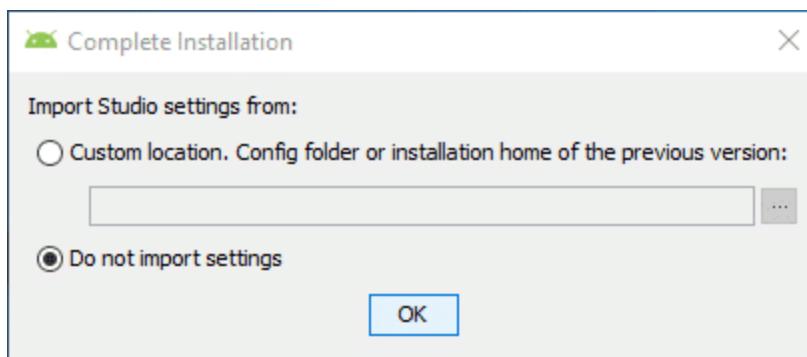
Una vez completado, continuamos.

## Paso 7



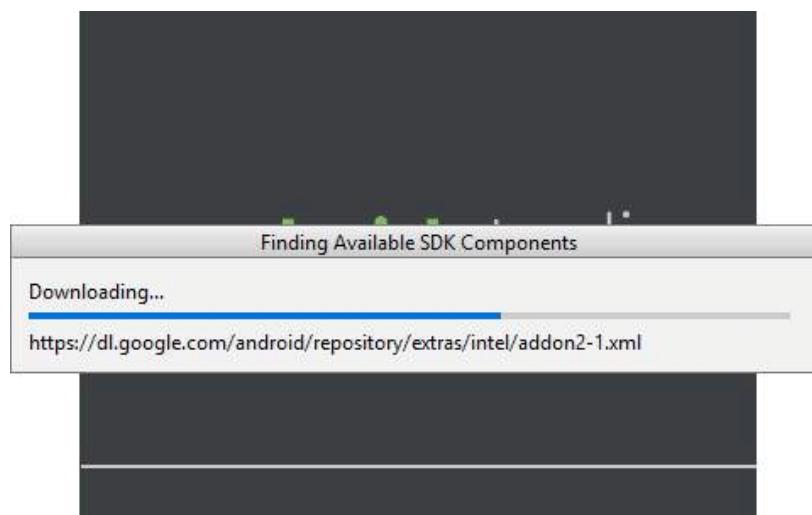
Ya instalado, vamos a abrirlo para configurarlo por primera vez.

## Paso 8



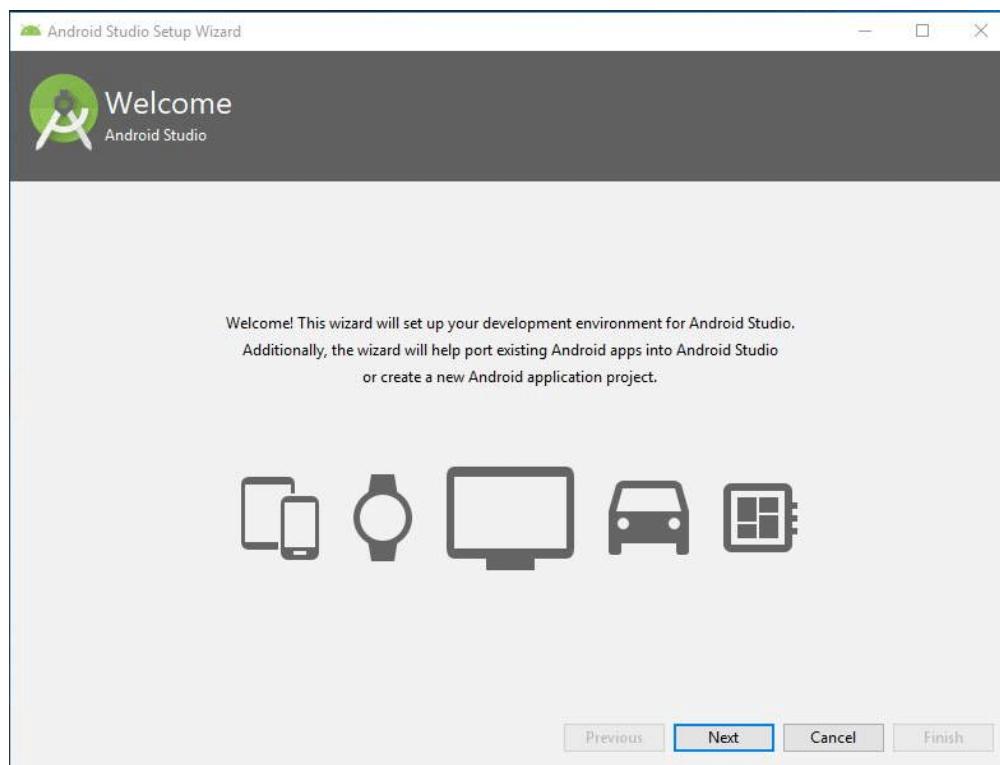
A la hora de configurar el programa, tenemos la opción de importar una configuración previa de otra instalación anterior, o bien podemos crear una nueva.

## Paso 9



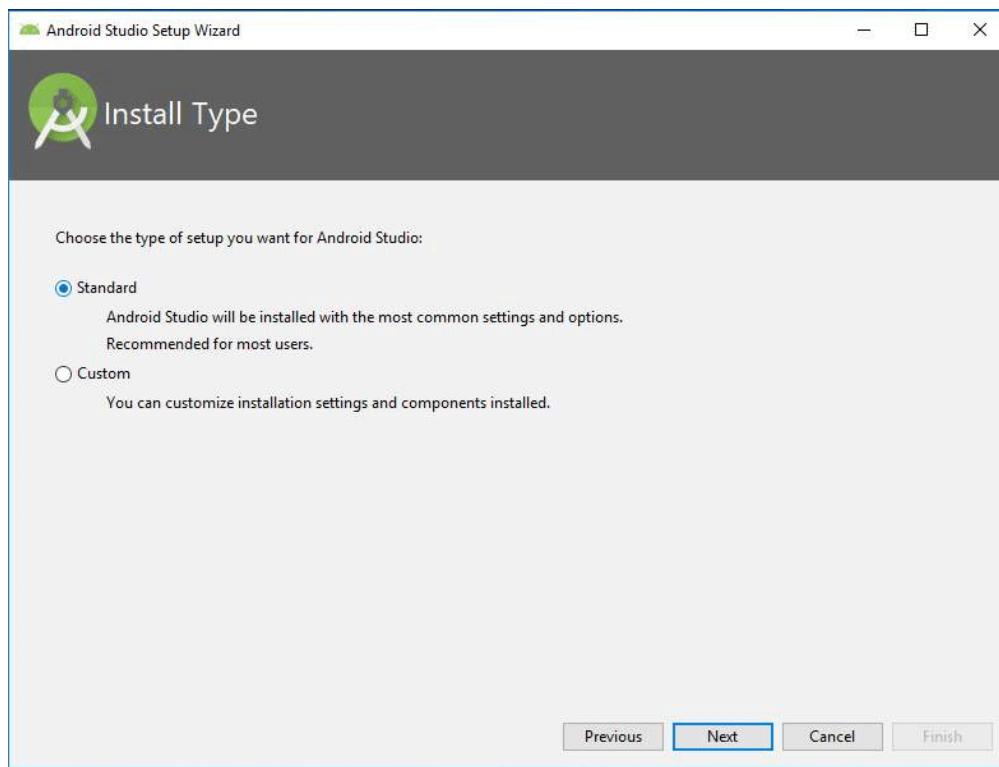
Android Studio descargará los componentes que necesite para esta configuración.

## Paso 10



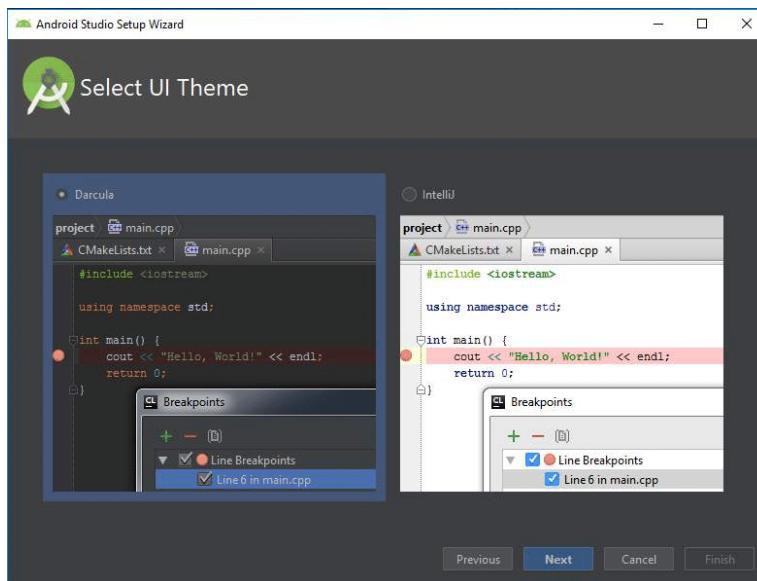
Una vez finalizado, un asistente nos guiará en los últimos pasos.

## Paso 11



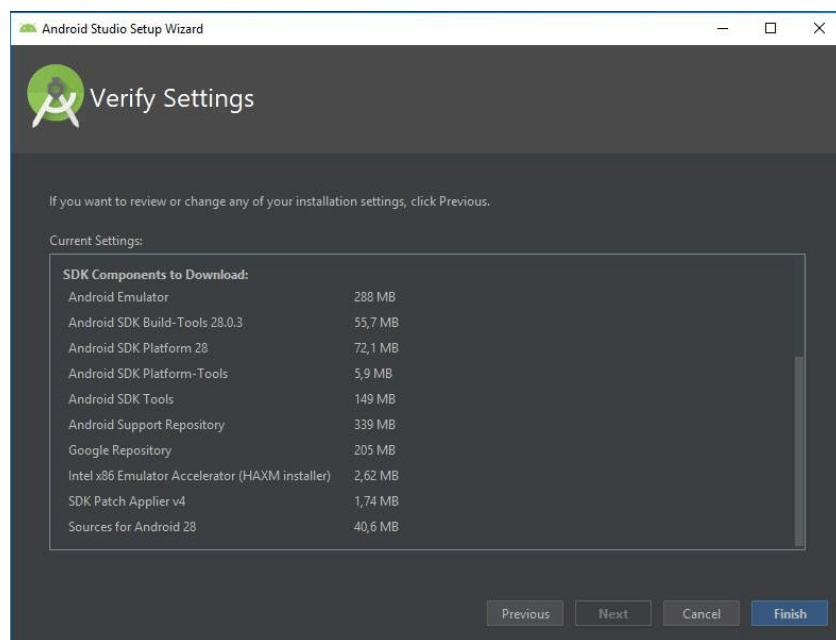
Seleccionamos la configuración y opciones por defecto, aunque podemos personalizar la instalación.

## Paso 12



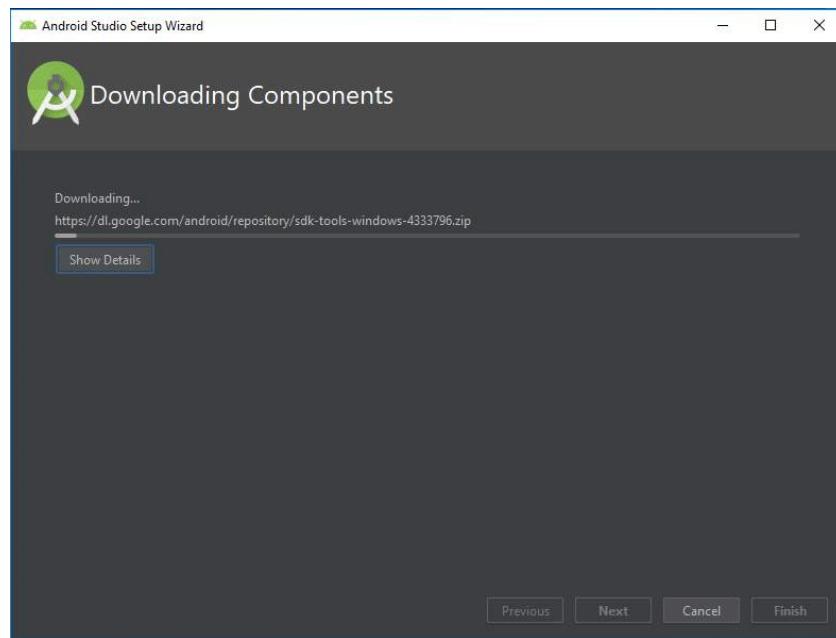
Elegimos un tema de interfaz de usuario.

## Paso 13



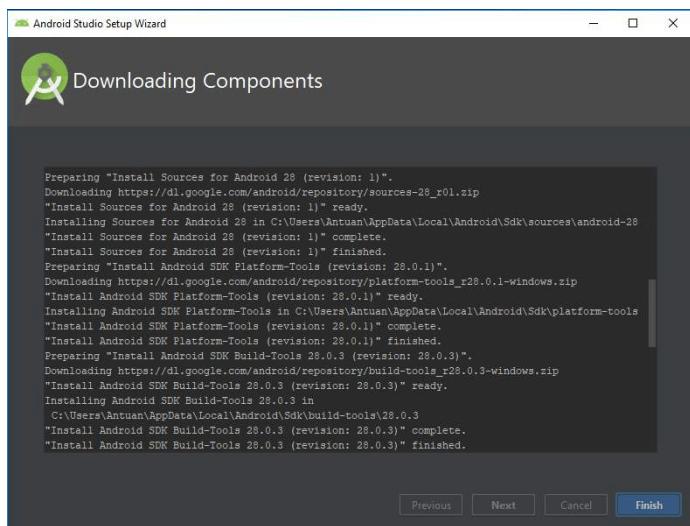
Vemos un resumen de nuestra elección y la configuración inicial, en la que se encuentra el último SDK.

## Paso 14



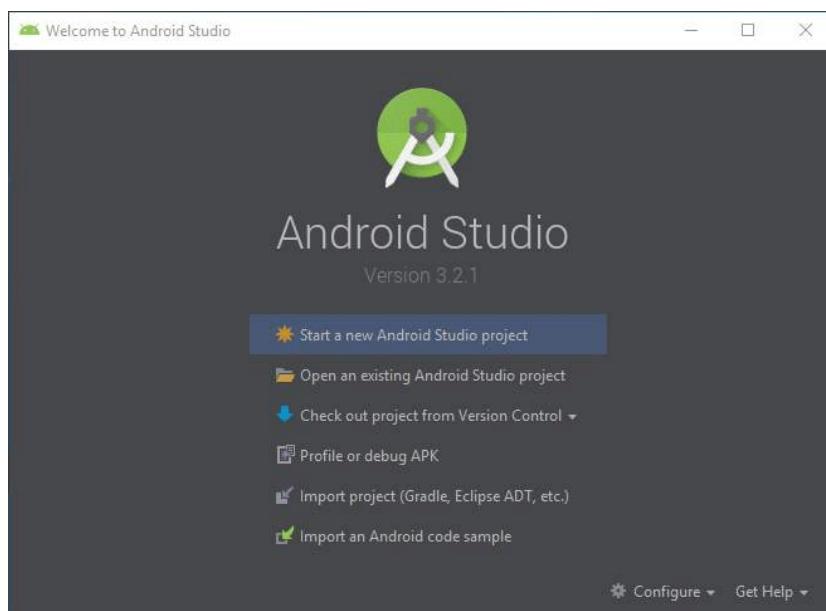
Al continuar, descargará de nuevo los componentes necesarios.

## Paso 15



Una vez finalizado nos muestra un resumen.

## Paso 16



Ya podemos ver la pantalla de INICIO.

## Paso 17

Ya tenemos Android Studio instalado y listo para su uso.

## Trabajar con Android Studio

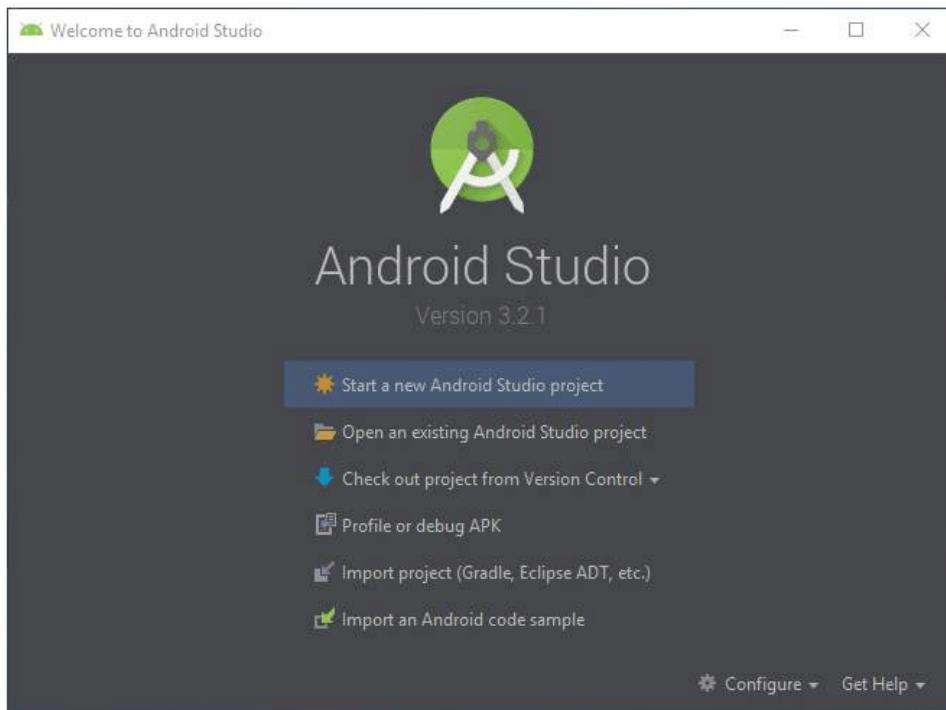
Vamos a crear un proyecto "Hola Mundo" como primer ejemplo y así iremos conociendo nuestro entorno de desarrollo y trabajo.

Veamos el paso a paso:

1

*Start a new Android Studio Project.*

Desde el menú principal que aparece cuando abrimos por primera vez Android Studio seleccionamos *Crear un nuevo Proyecto*.



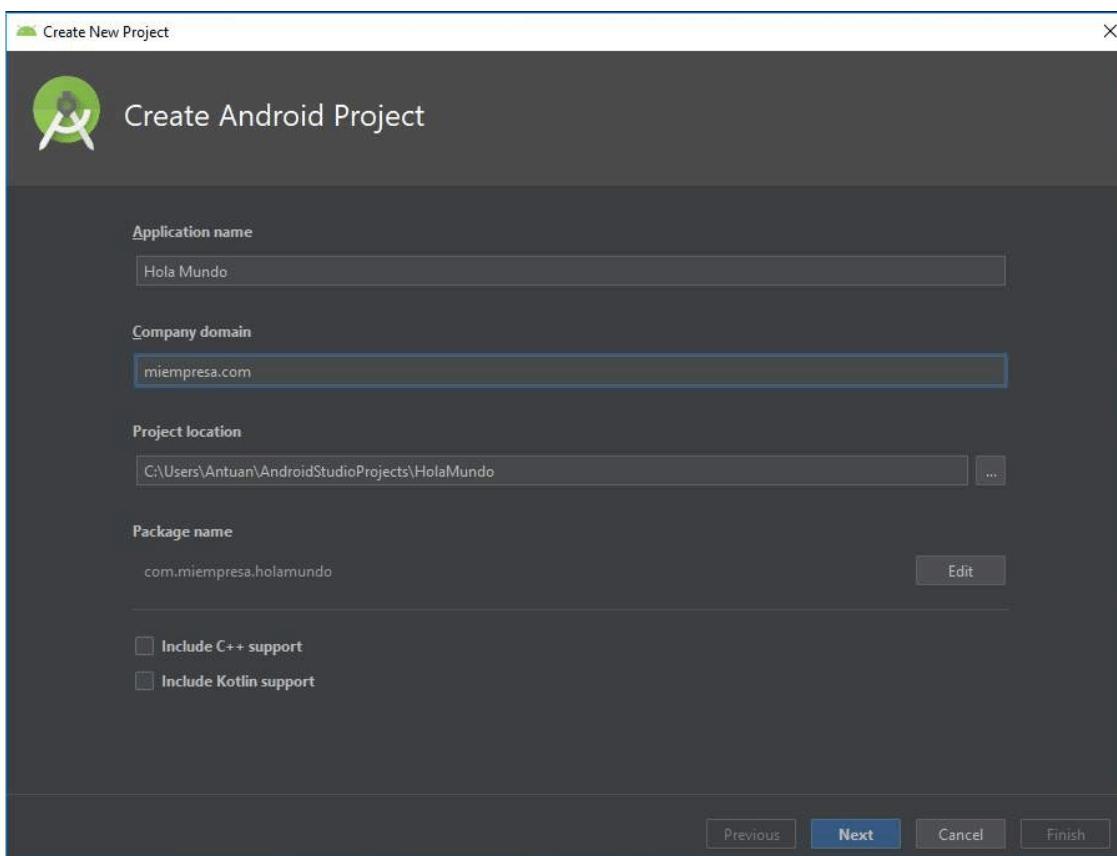
Ventana principal cuando empezamos con **Android Studio**; si cerramos todos los proyectos "**Close project**" desde el menu "**File**", volveremos a la **ventana principal**.

2

*Create Android Project.*

Escribimos un nombre para nuestra app, que en nuestro ejemplo será "Hola Mundo". También escribimos una URL para hacer referencia al dominio; esto es importante, porque los usuarios encontrarán en Google Play nuestra app a través de esta URL. Normalmente la ruta es algo como: com.miempresa.nombredelaapp.

No olvides **especificar la ubicación del proyecto**. Por defecto, esta ubicación se encontrará en la carpeta *documentos* de nuestro usuario.



Recuerda elegir un nombre de dominio válido en "**Company domain**" porque esta es la dirección que necesitan los usuarios para poder encontrar nuestra app en **Google Play**.

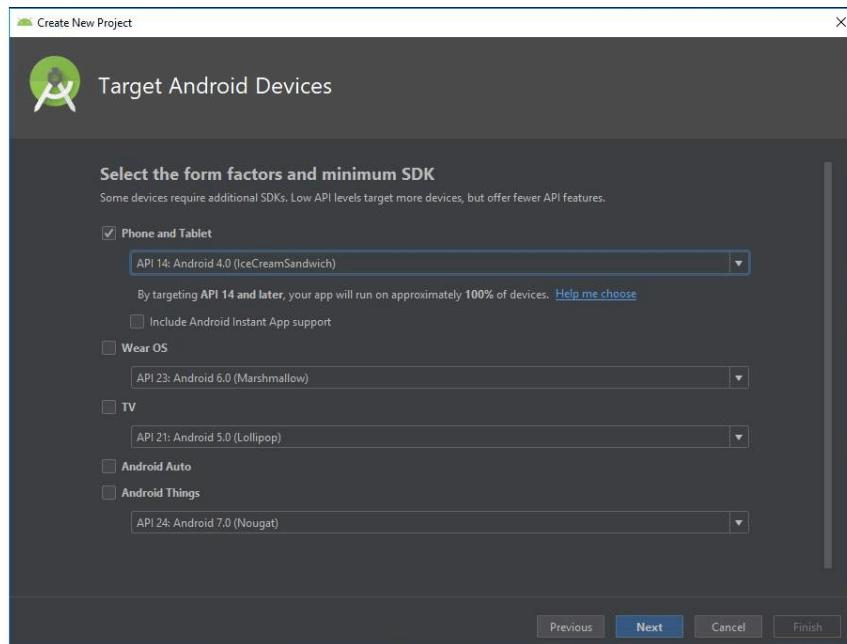
3

### Target Android Devices

En este paso debemos **seleccionar la versión de Android que deseamos utilizar como mínima para nuestro desarrollo**. En nuestro ejemplo, vamos a seleccionar una versión más actual, como la API 24: Android 7.0 (Nougat).

Si, por ejemplo, vamos a usar características actuales, probablemente la app no funcionará completamente en versiones anteriores; es decir, en función de la versión que elegimos de Android, nuestra app podrá funcionar en terminales modernos, o algunos usuarios no podrán descargarla porque su *smartphone* no cumplirá con los requisitos.

- (i) ¡Ojo! Debemos elegir correctamente la versión de Android para el desarrollo de nuestra app. En caso contrario, nuestros usuarios no podrán instalarla en sus dispositivos porque no cumplirán los requisitos.**



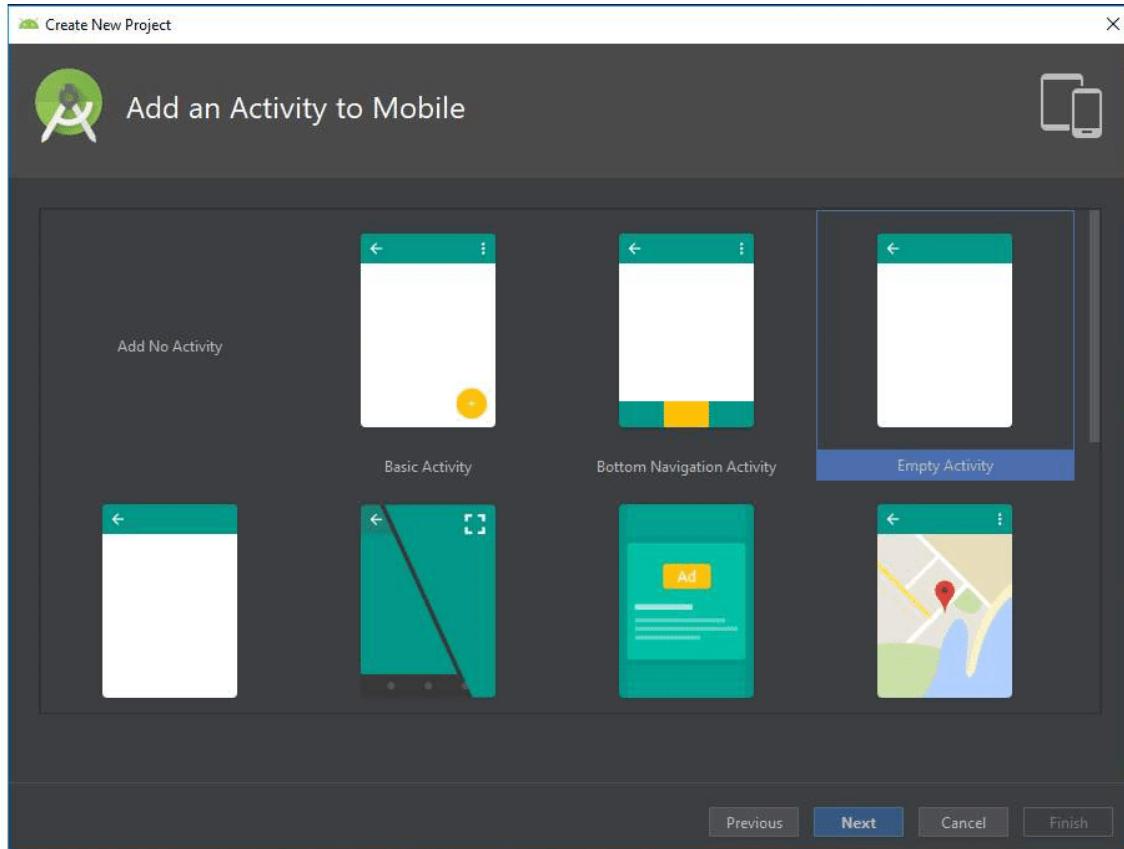
Si no elegimos la **API** adecuada para el desarrollo de nuestra app nuestros usuarios no podrán descargarla ni usarla.

4

*Add an Activity to Mobile.*

Las *Activities* constituyen la interfaz gráfica con la que el usuario interactúa con nuestra app. En una *Activity* encontraremos todo tipo de elementos gráficos: botones, cajas de texto, imágenes, listas, etc.

Para nuestro proyecto "Hola Mundo" seleccionaremos *Empty Activity* como plantilla y la usaremos en este ejemplo.



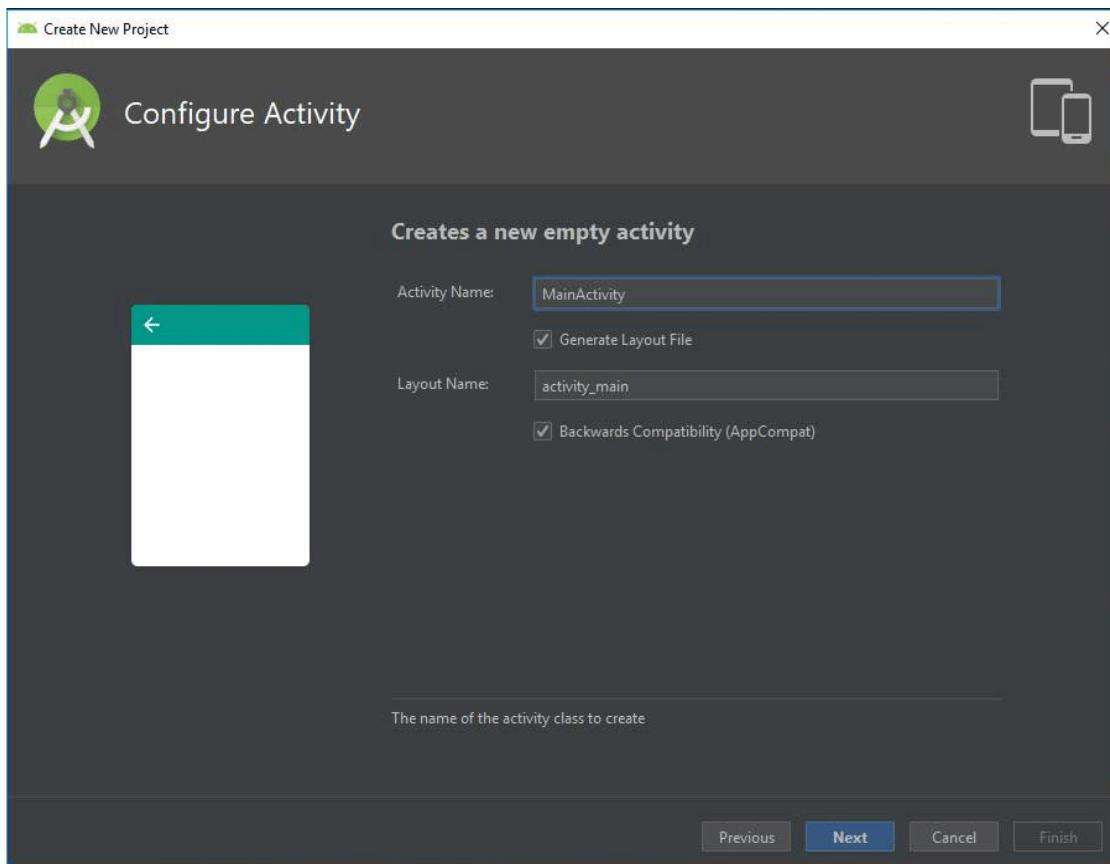
**¡Importante!** Para empezar a desarrollar una app en **Android Studio** es necesario elegir una *template* adecuada. El asistente nos ofrece muchas *templates* para facilitarnos una parte del desarrollo.

5

*Configure Activity.*

Una *Activity* es una clase Java, pero nosotros no podemos mostrar al usuario un código de programación Java. El usuario interactúa con nuestra app a través de las *activities*, pero para que pueda hacerlo nuestra *activity* necesita un *Layout*. Un *Layout* es un archivo XML que permite al usuario ver la Interfaz gráfica.

Debemos especificar un nombre para nuestra *Activity*. En este paso también estamos configurando un *Layout* que pueda usar nuestra *Activity*; nuestro *Layout* contendrá los componentes gráficos, como mencionamos anteriormente.

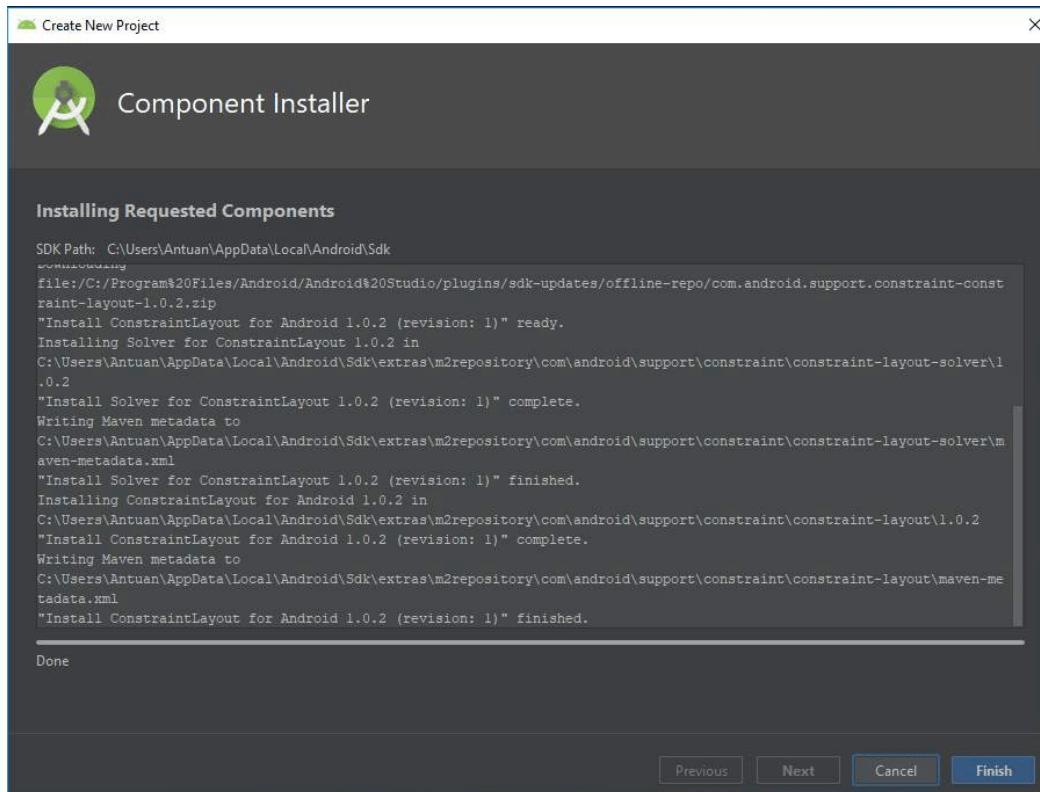


Una *activity* es un "fichero java" que necesita un *layout* "fichero xml" para que el usuario pueda interactuar con una app.

6

## Instalación.

Si es necesario, Android Studio instalará nuevos componentes para este proyecto.

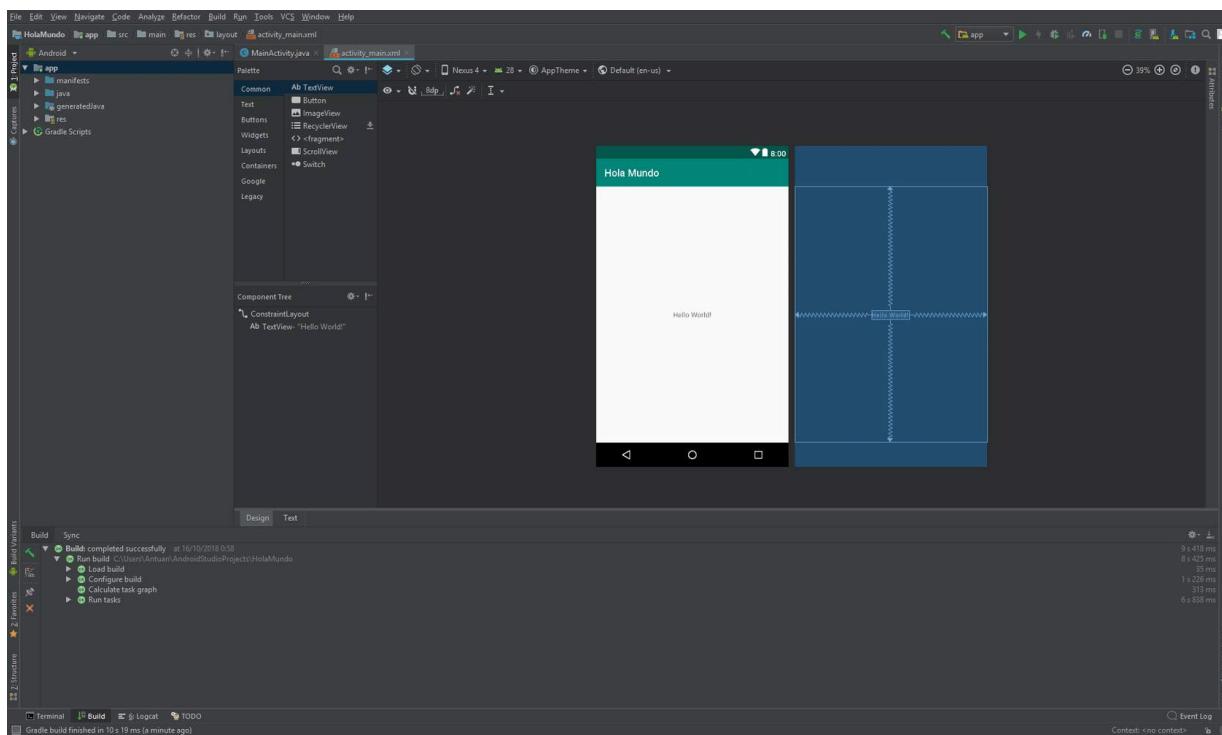


7

## Project Structure.

Cuando finalizamos los pasos del asistente, creando un proyecto en Android Studio, lo primero que vemos son nuestra *Main Activity* y su *Layout* en las pestañas del editor. Podemos cambiar en las diferentes pestañas para ver el fichero *MainActivity.java* y el fichero *activity\_main.xml*.

Nuestro proyecto "Hola Mundo" contiene una estructura de ficheros que compone nuestra app. Precisamente el *folder* principal en el panel de la izquierda tiene el nombre de "app", y dentro contiene otros *folders*: *manifests*, *java*, *res*, etc.



Estructura del proyecto "Hola Mundo" en Android Studio.

## Crear un emulador

---

A lo largo de este apartado vamos a mostrarte cómo manejar el emulador de Android.

Una de las funciones que nos ofrece Android Studio es la posibilidad de montar una máquina virtual de Android con la que poder hacer uso del sistema operativo de Google, y de sus aplicaciones, desde nuestro ordenador.

---

Esta versión ha supuesto para Google un gran desafío, ya que la compañía ha trabajado duro en implementar una serie de novedades, mejoras y cambios que los usuarios llevaban tiempo pidiendo y que, por motivos técnicos, no se habían podido implementar antes.

Android Emulator ejecuta una pila completa de sistema de Android, hasta el nivel del *kernel*, que incluye un conjunto de apps preinstaladas (como el teléfono) a las que puedes acceder desde tus apps. Puedes elegir la versión del sistema Android que deseas ejecutar en el emulador al crear AVD.

Las imágenes de sistema de Android disponibles a través del Administrador de AVD contienen código para el *kernel* de Linux de Android, las bibliotecas nativas, la VM y los diferentes paquetes de Android (por ejemplo, las apps preinstaladas y el framework de Android).

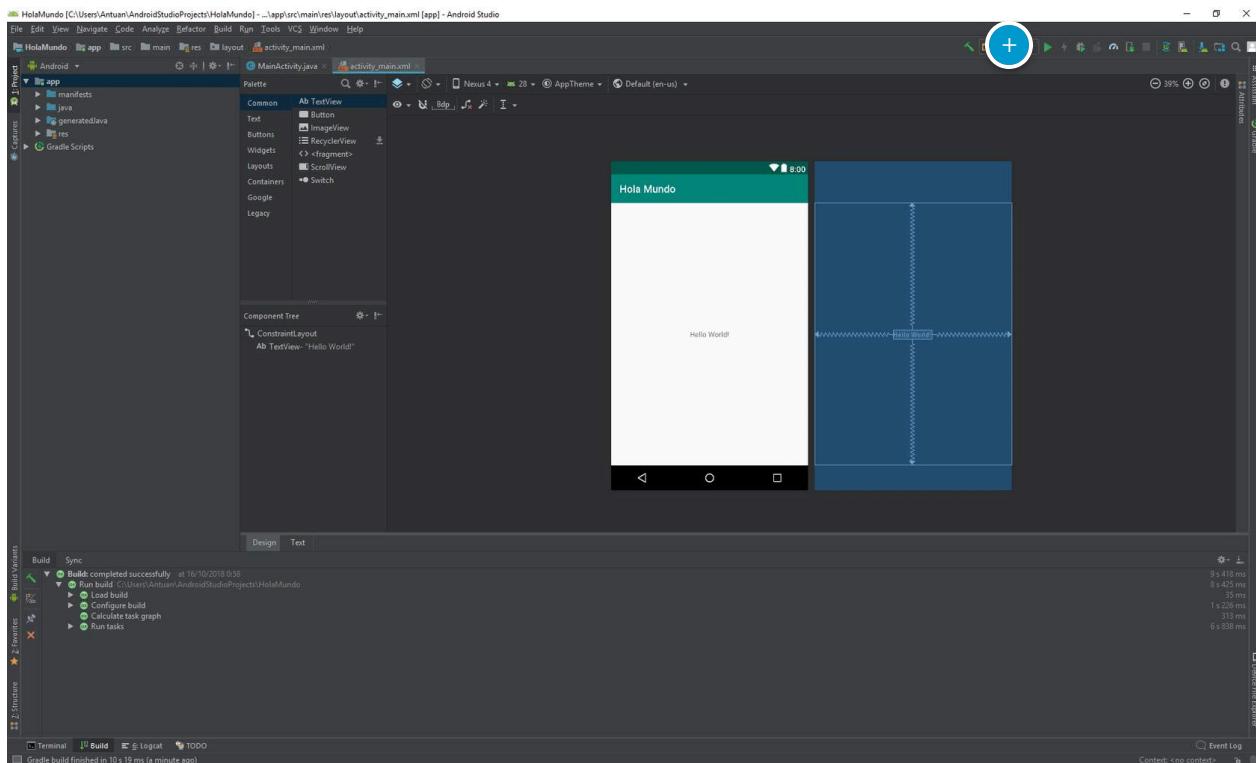
Hay que tener en cuenta que desde el emulador no son accesibles todas las características del dispositivo, y que no incluye hardware virtual para lo siguiente:

- WiFi.
- Bluetooth.
- NFC.
- Inserción/expulsión de tarjetas SD.
- Auriculares conectados a dispositivos.
- USB.



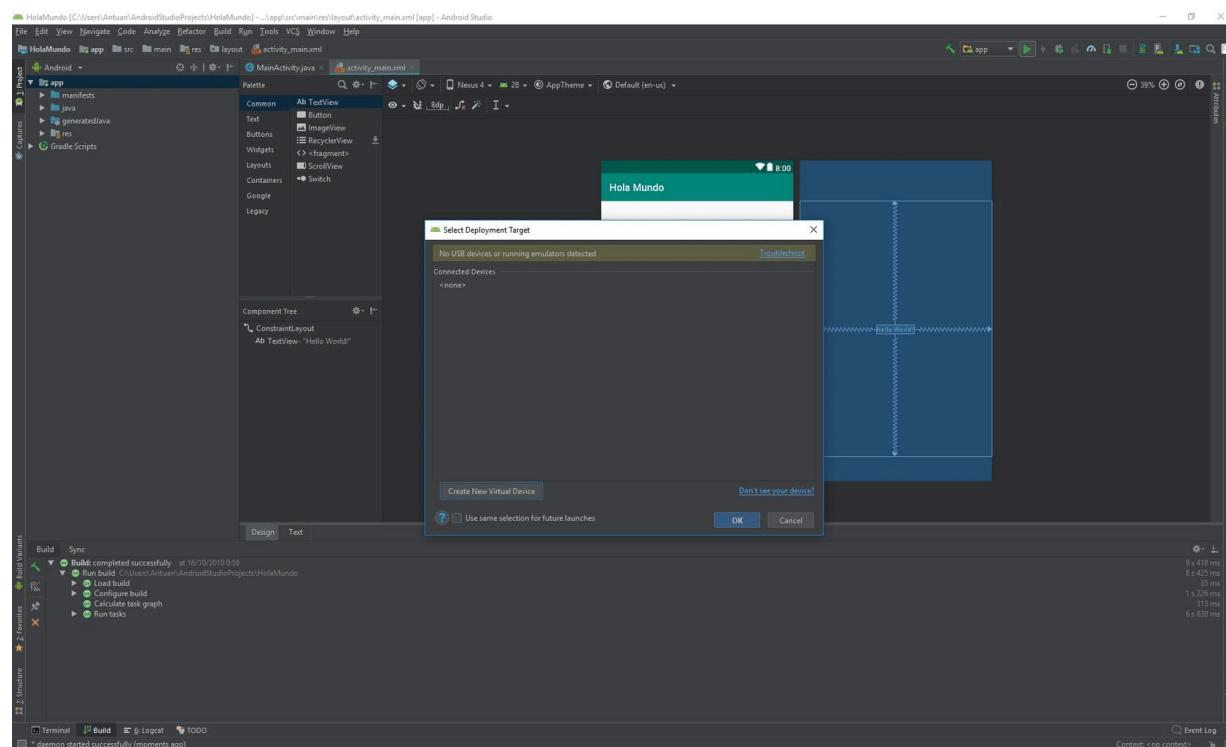
## Crear un emulador

Una vez que estemos en nuestro proyecto, por ejemplo, la aplicación "Hola Mundo" que hemos creado, pulsamos en "Run App".

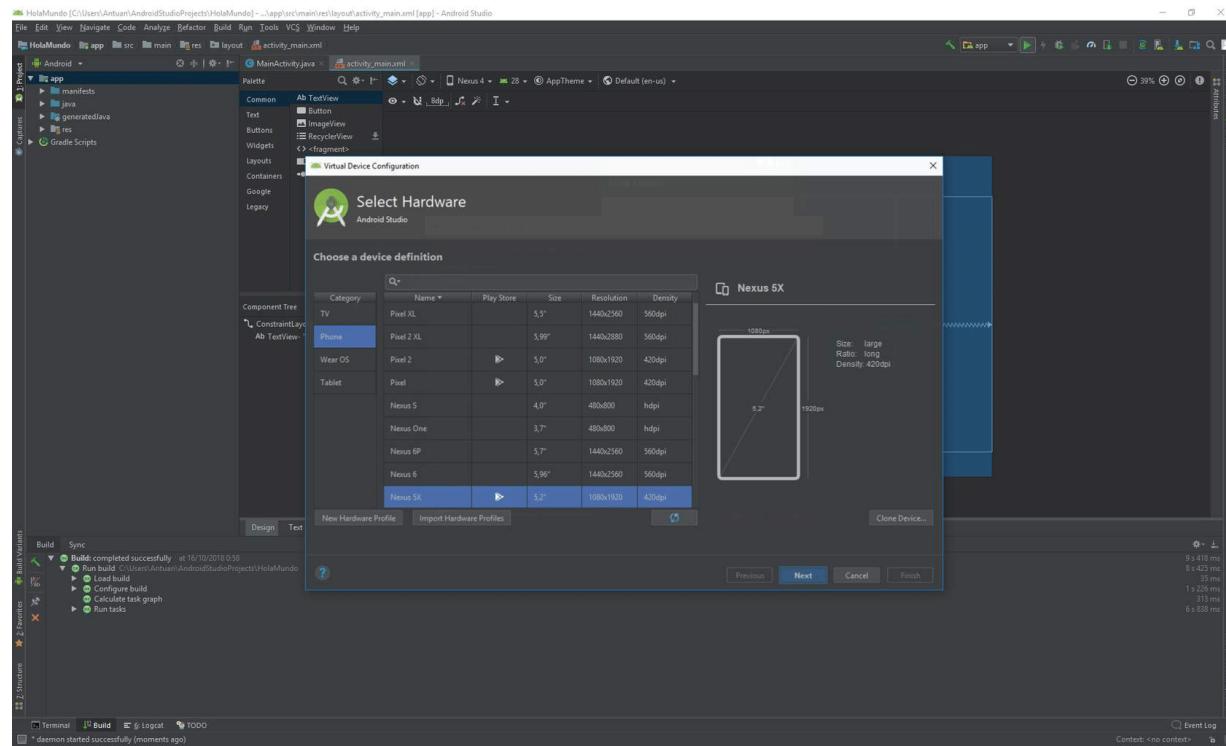


**Run App:** Es el botón con un triangulo verde.

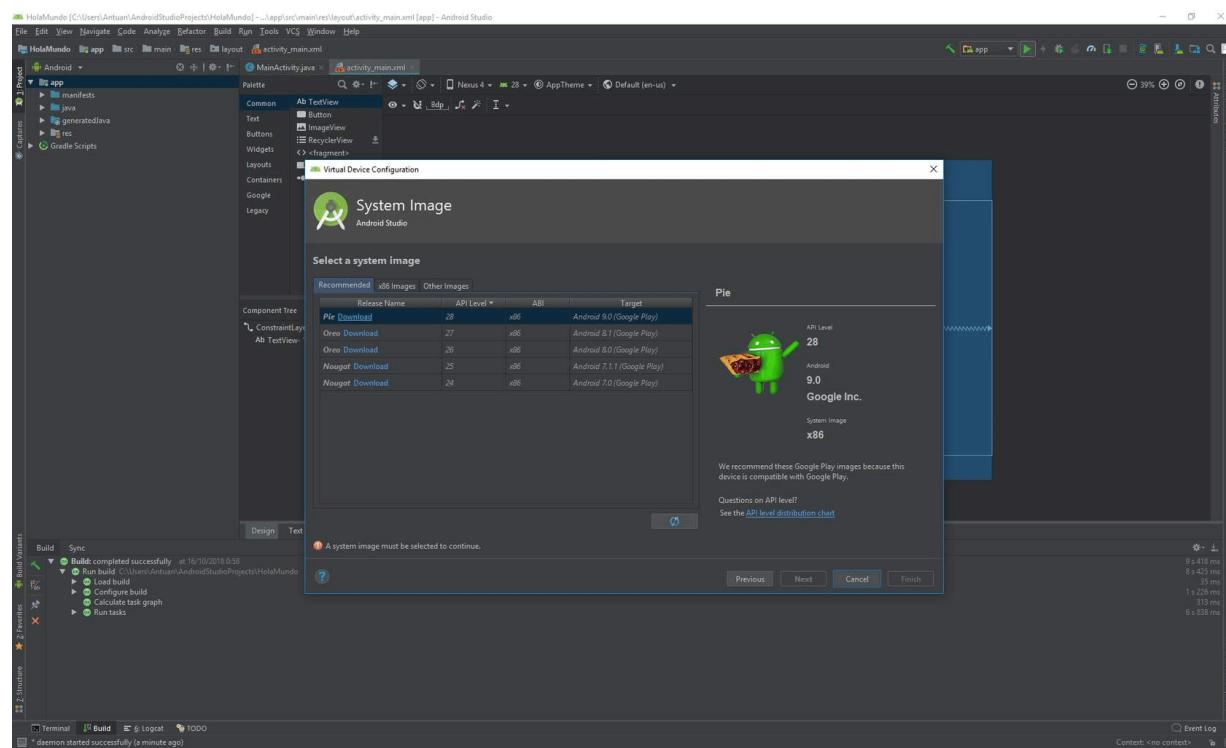
Aparecerá una ventana para seleccionar el dispositivo que tengamos conectado. Si no poseemos ninguno, podemos crear un dispositivo virtual, para emularlo.



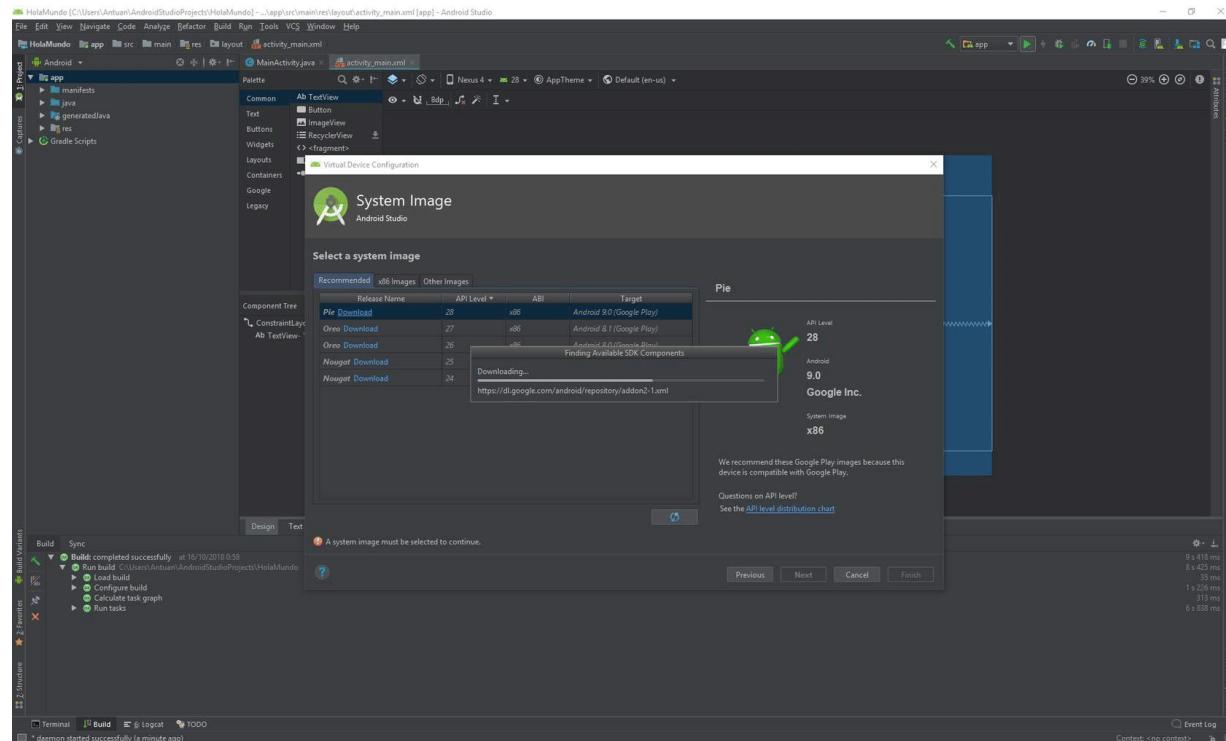
Pulsamos sobre "Create New Virtual Device" y nos aparecerá una ventana donde podremos elegir qué dispositivo queremos emular, y sus características.



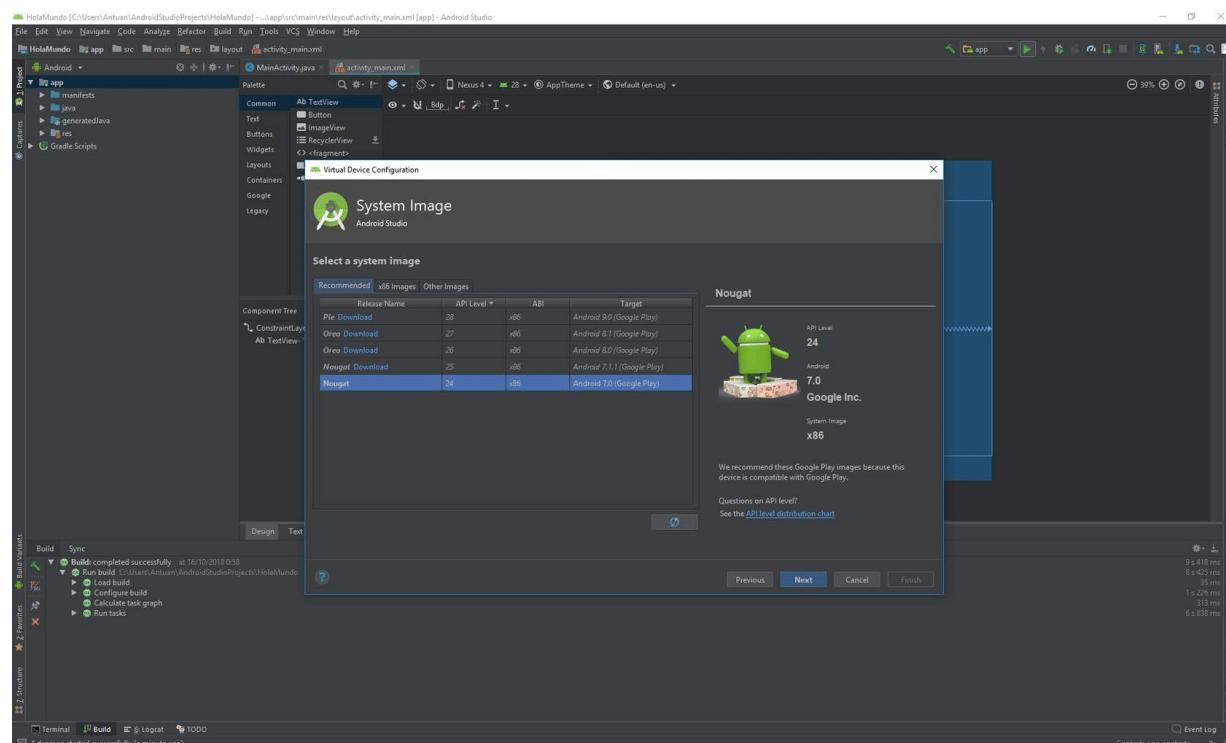
Ahora, elegiremos una **imagen de sistema**, que contendrá la versión elegida del sistema operativo.



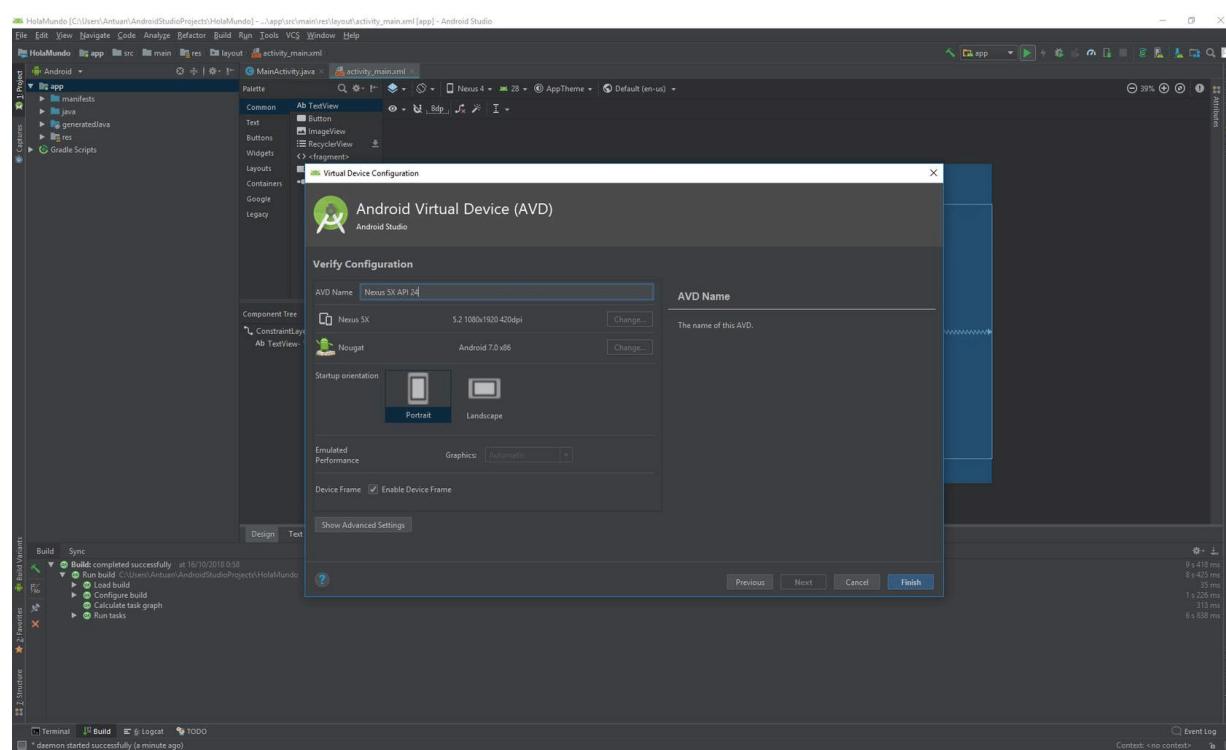
Descargará los componentes necesarios.



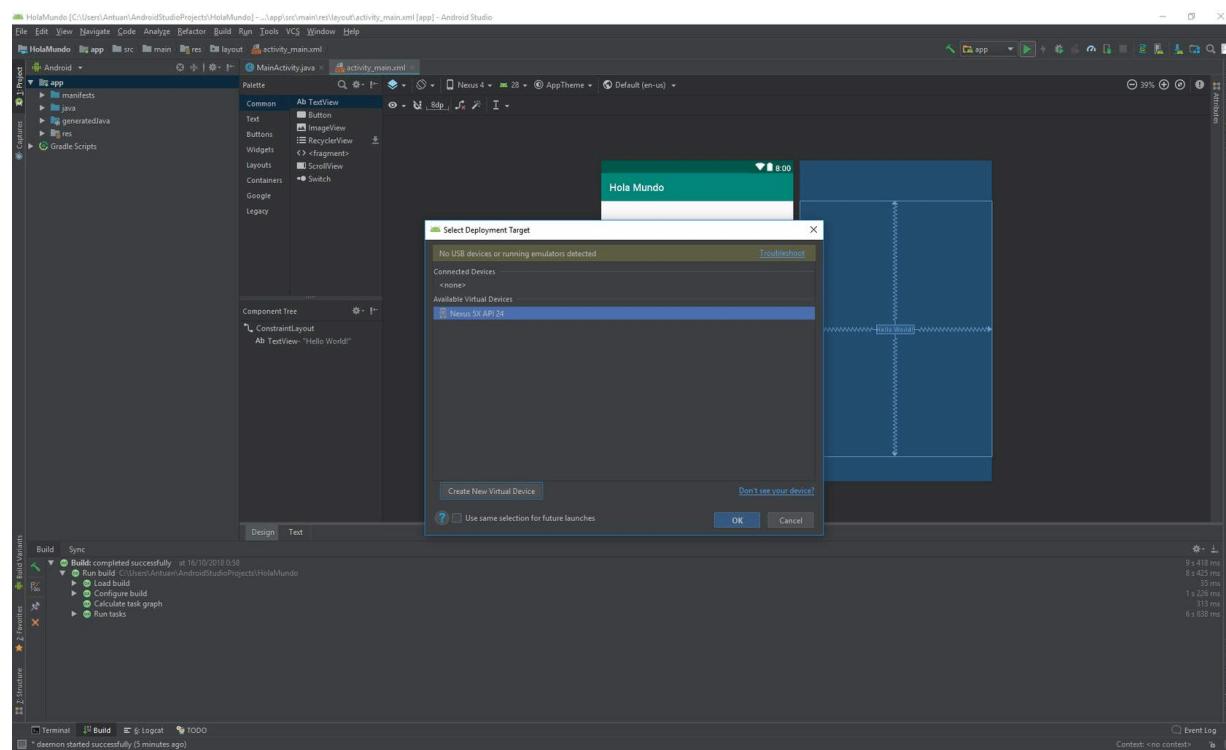
Los instalará y nos indicará que están instalados. Ahora, podremos elegir la imagen de sistema que hemos descargado.



Confirmamos el nuevo dispositivo que vamos a crear.



Y aparecerá en nuestra lista de dispositivos virtuales.

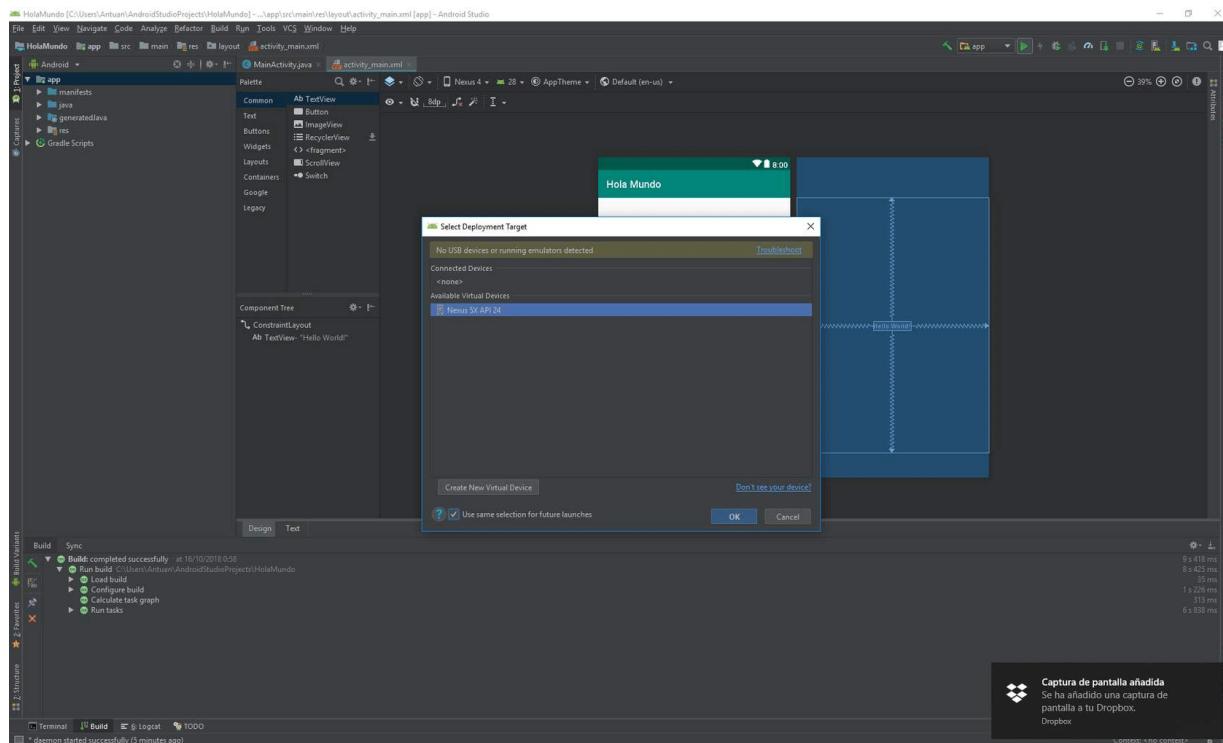


---

**Comprobemos en nuestra aplicación "Hola Mundo" el uso del emulador.**

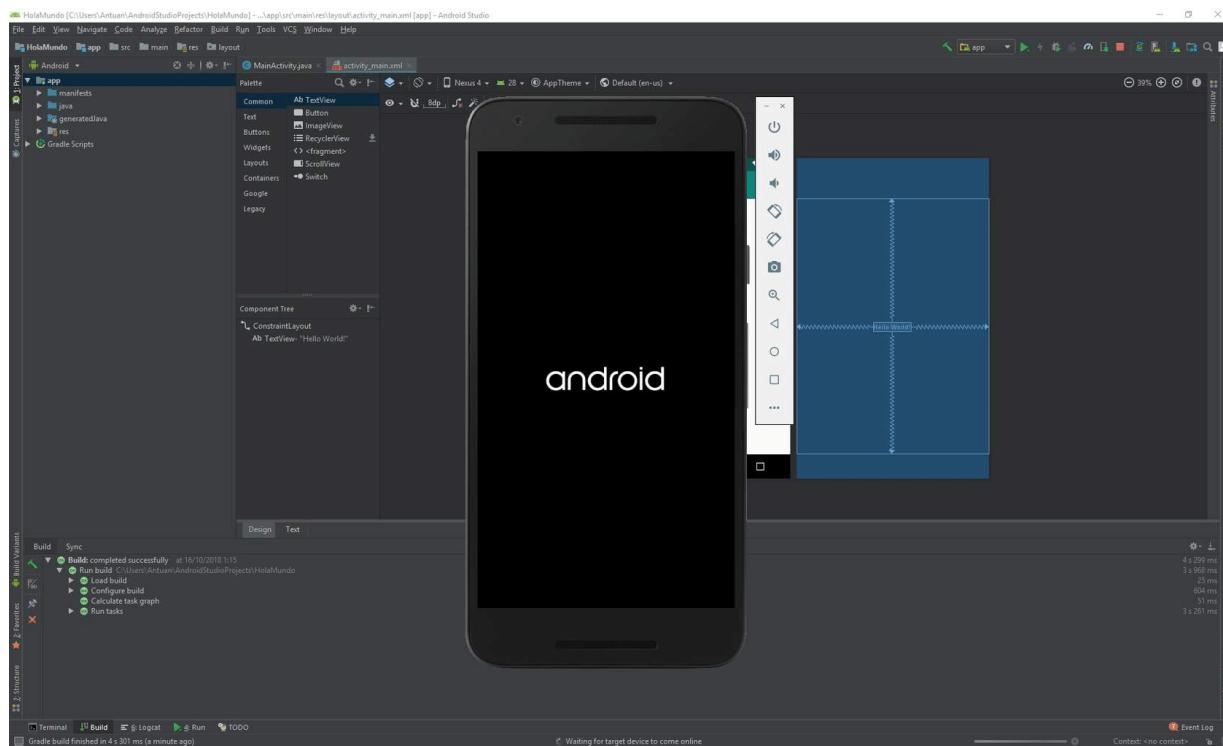
### Paso 1

Pulsamos "Run App" arriba a la izquierda. Cuando veamos nuestra lista de dispositivos virtuales, elegimos el que nos interese, en este caso, el que hemos creado, "Nexus 5x API 24".



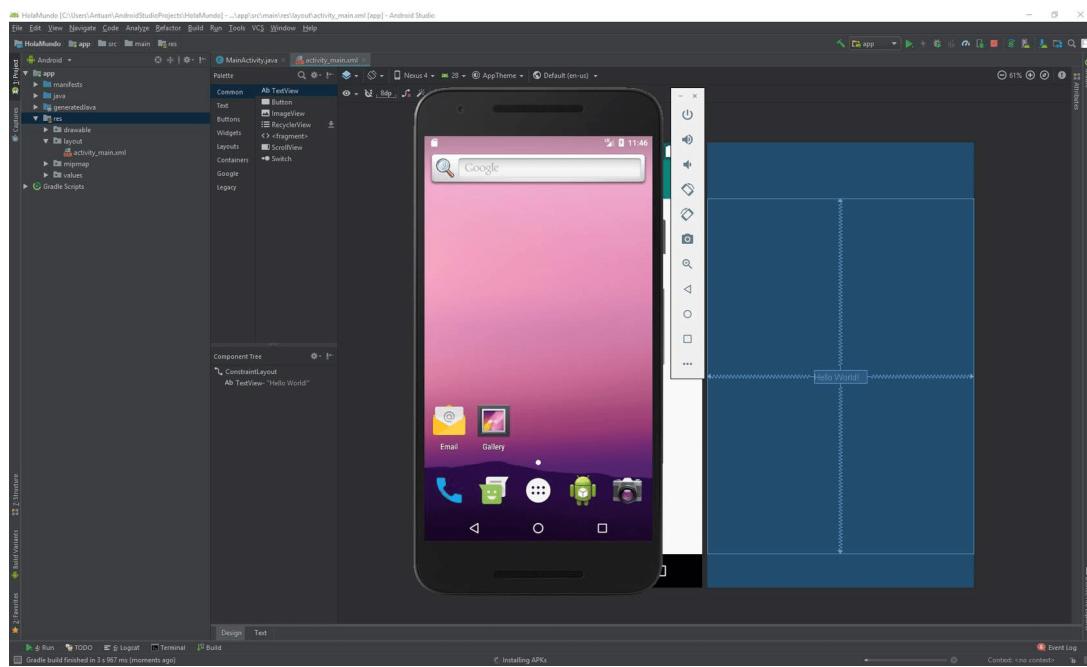
## Paso 2

Se abrirá el emulador y empezará a cargar como si fuese un dispositivo real.



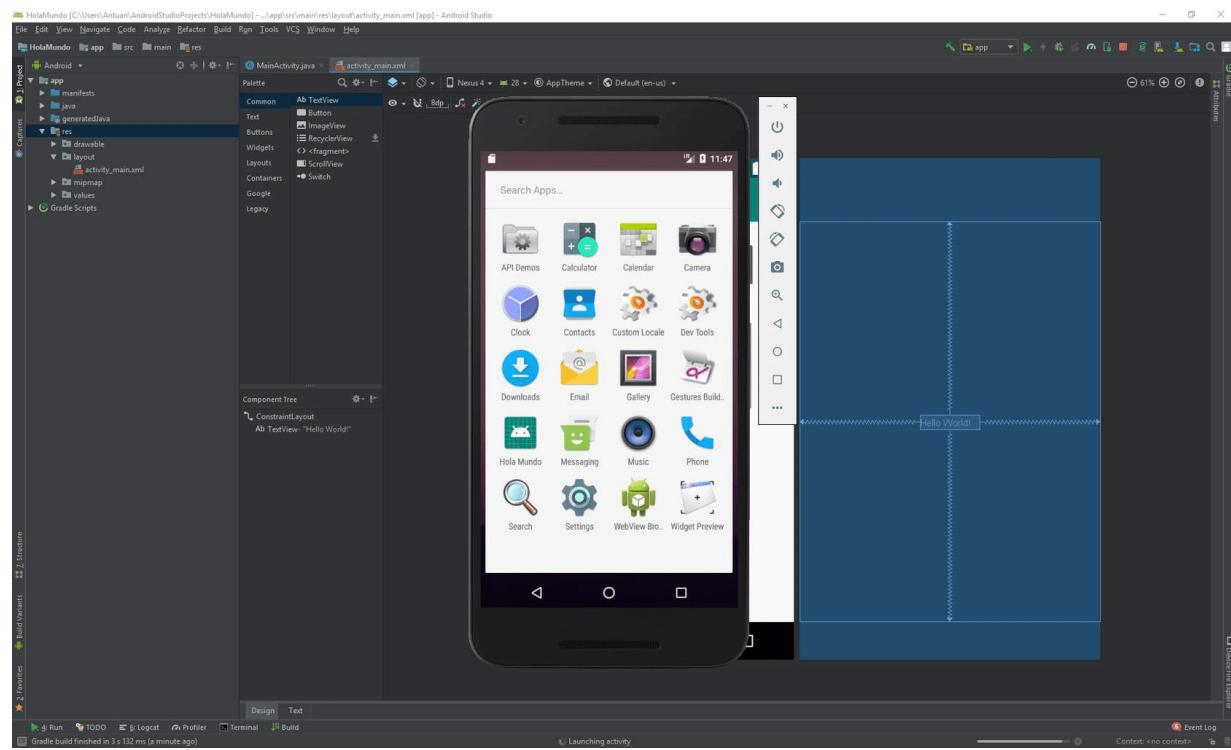
## Paso 3

Una vez cargado, se nos muestra el dispositivo.



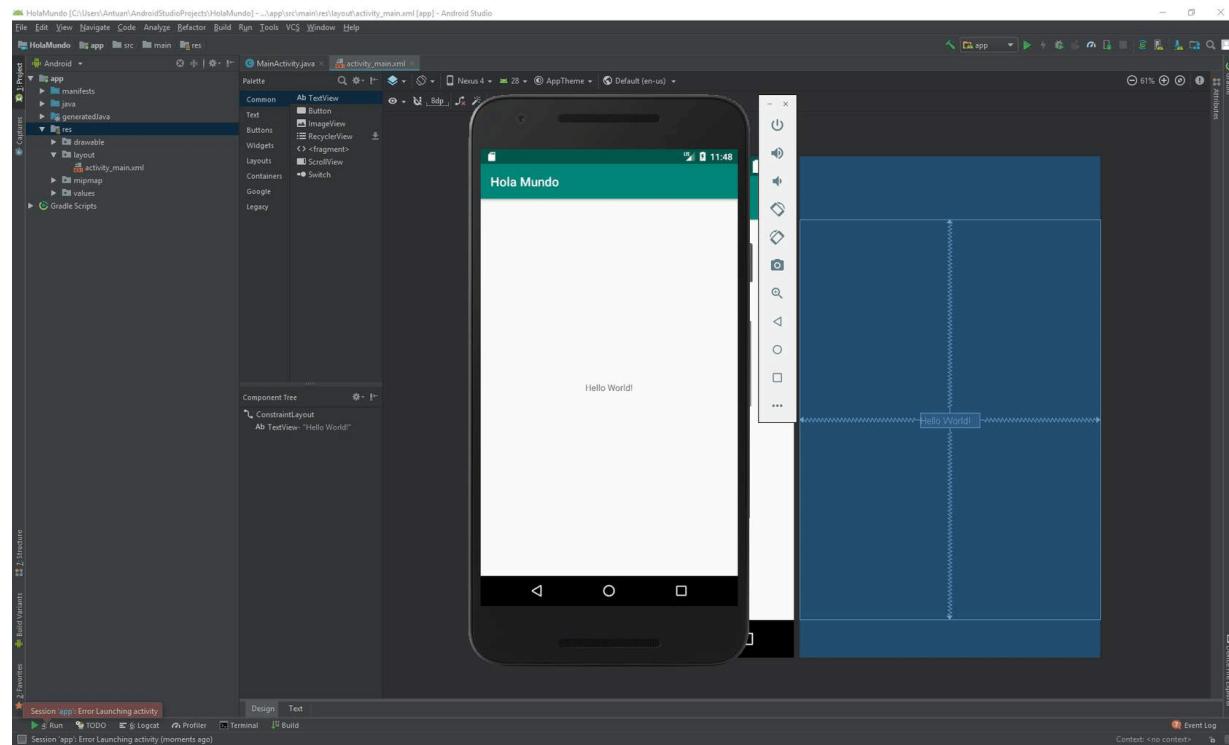
## Paso 4

Navegamos por él como por un dispositivo real, buscando nuestra aplicación y abriéndola.



## Paso 5

Comprobamos el funcionamiento de nuestra aplicación.



## Resumen

---

Hemos terminado la lección, veamos los puntos más importantes que hemos tratado.

- Hemos aprendido que un [Entorno de Desarrollo Integrado](#) (IDE) es una aplicación informática que contiene un kit de herramientas para facilitarnos el desarrollo de software.
- También hemos comprendido que los IDE presentan un único programa en el que se lleva a cabo todo el desarrollo.
- Hemos descubierto que **Android Studio** es el IDE oficial de Google para desarrollar aplicaciones para Android.
- Y, para finalizar, hemos aprendido a **instalar Android Studio y cómo funcionar con su emulador**, para poder verificar el código que estemos desarrollando.



**PROEDUCA**