

PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES  
**TÉCNICO EN DESARROLLO DE APLICACIONES  
MULTIPLATAFORMA**

## Construcción de proyectos Android

---

02

# ÍNDICE

/ 1. Introducción y contextualización práctica	3
/ 2. Arquitectura del sistema operativo Android	4
/ 3. Instalación de Android Studio	5
/ 4. Caso práctico 1: “¿Hay más entornos de desarrollo para crear aplicaciones Android?”	6
/ 5. Creación de un emulador integrado	6
/ 6. Instalación del emulador Android NOX	7
/ 7. Creación de un proyecto	7
/ 8. Caso práctico 2: “¿Es mejor usar un emulador o depurar con mi dispositivo?”	8
/ 9. Ejecución de un proyecto	9
/ 10. Bibliografía	10

# OBJETIVOS



*Conocer la arquitectura del sistema operativo Android.*

*Instalar entornos de desarrollo para aplicaciones móviles.*

*Instalar dispositivos emuladores.*

*Crear y ejecutar un proyecto básico en Android.*



## / 1. Introducción y contextualización práctica

En esta unidad, vamos a conocer la arquitectura que utiliza el sistema operativo móvil Android.

También vamos a ver cómo podemos **instalar un entorno de desarrollo** que nos permita **crear y depurar aplicaciones** móviles Android, como es el caso de Android Studio.

También veremos cómo crear un emulador de un dispositivo Android para poder probar nuestras aplicaciones, así como la manera en la que podemos configurar nuestro dispositivo móvil para ejecutar y depurar las aplicaciones directamente.

Por último, estudiaremos cómo **crear un proyecto vacío** y compilarlo en Android Studio.

Escucha el siguiente audio en el que planteamos el caso práctico que iremos resolviendo a lo largo de esta unidad:

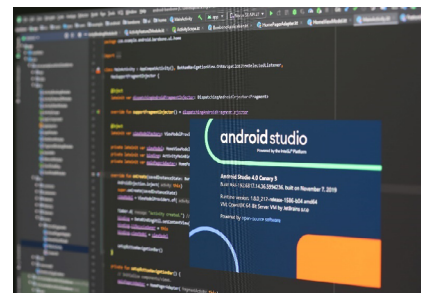


Fig. 1. Android Studio.



Audio Intro. "Reutilización de conocimientos"  
<https://bit.ly/2Zoulsv>

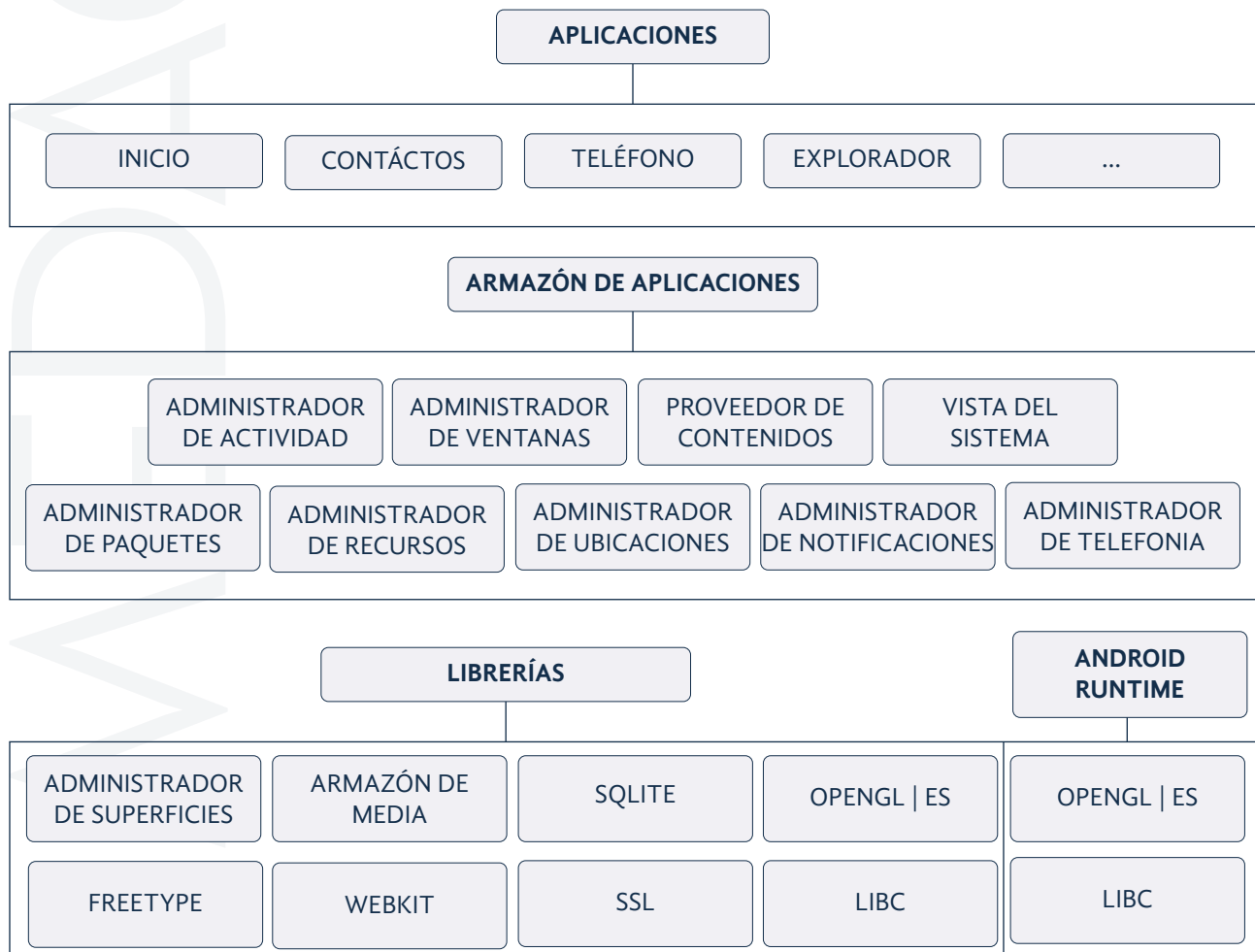




## / 2. Arquitectura del sistema operativo Android

El sistema operativo Android posee la siguiente arquitectura:

- **Capa de aplicaciones:** En esta capa vamos a encontrar todas las aplicaciones que forma la base de Android, como puede ser un cliente de correo electrónico, un programa de gestión de SMS, etc. Hay que destacar que todas estas aplicaciones están escritas en lenguaje Java.
- **Capa del marco de aplicaciones:** Esta capa existe para simplificar la reutilización de componentes por el usuario.
- **Capa de bibliotecas:** Esta capa está escrita en C/C++ y la usan muchos de los componentes que forman el sistema operativo Android, como las aplicaciones de gráficos, bases de datos, etc.
- **Capa de runtime:** Conforman un conjunto de bibliotecas que proporcionan muchas de las funciones del sistema operativo.
- **Núcleo Linux:** Es la principal de todas, ya que es el núcleo o kernel el que va a proporcionar todo lo relacionado con la gestión de los recursos del dispositivo, ya sean memoria, almacenamiento, comunicaciones de red, etc.



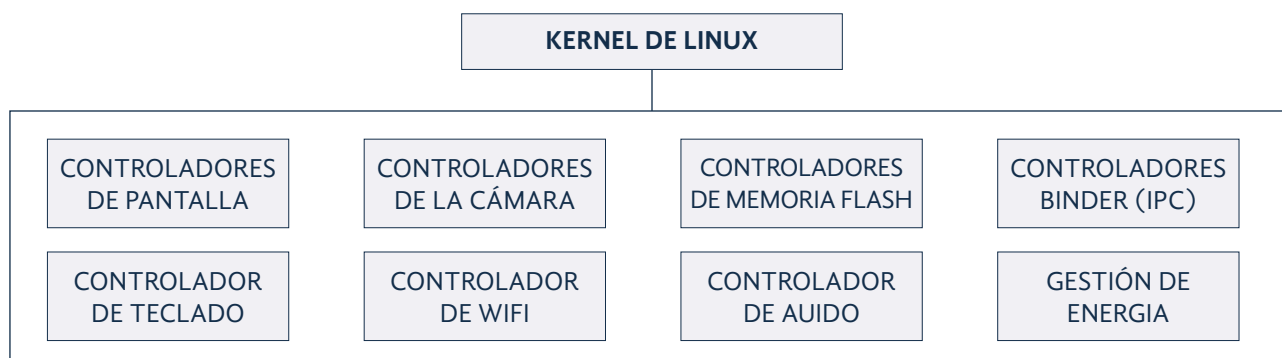
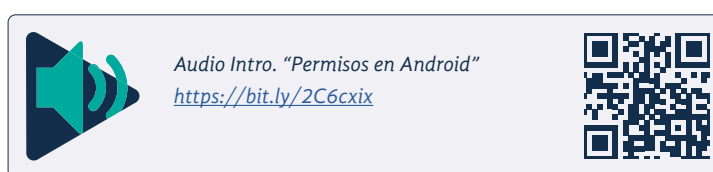


Fig. 2. Capas del sistema operativo Android.



## / 3. Instalación de Android Studio

El entorno de desarrollo que vamos a utilizar para crear nuestras **aplicaciones** Android será **Android Studio**.

Android Studio es el **compilador oficial para desarrollar aplicaciones** para el sistema operativo Android. Este se basó en IntelliJ IDEA de JetBrains y está publicado de forma gratuita a través de la Licencia Apache 2.0. Además, podremos utilizarlo en las plataformas Microsoft Windows, macOS y GNU/Linux.

Podemos **descargar** Android Studio desde su propia página web: <https://developer.android.com/studio>

Una vez descargado, simplemente debemos **aceptar** todas las pantallas de instalación, haciendo clic en «siguiente».

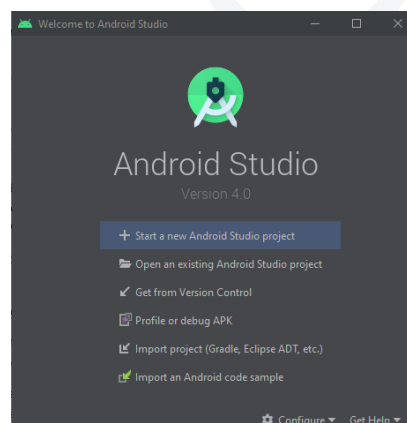


Fig. 3. Pantalla inicial de Android Studio.

Tras instalarlo, tendremos que iniciar Android Studio para que se instale toda la JDK necesaria para crear aplicaciones Android. Una vez hecho esto, aceptaremos todas las pantallas pulsando en «**siguiente**» y comenzará la descarga del compilador y demás herramientas de Android (esto puede tardar varios minutos, dependiendo de la conexión a internet).

Un punto que debemos tener en cuenta en la instalación es el entorno visual que queremos en Android Studio. En la instalación se nos ofrecerá tanto el tema claro (Light) como el tema oscuro (Darcula). Nosotros **recomendamos instalar el tema oscuro**, aunque esto se podrá cambiar fácilmente en la configuración de Android Studio una vez esté instalado.

Para realizar esta **instalación** en cualquier plataforma (GNU/Linux, Windows o Mac), podemos utilizar el **siguiente enlace** que nos explica fácilmente cómo hacerlo:



**Enlace de interés...**

<https://developer.android.com/studio/install.html?hl=es-419>



## / 4. Caso práctico 1: “¿Hay más entornos de desarrollo para crear aplicaciones Android?”

**Planteamiento:** En el mundo de la programación es normal que haya diferentes entornos de desarrollo para crear aplicaciones en un mismo lenguaje, y las aplicaciones Android no son una excepción. Cuando queremos iniciarnos en el desarrollo de aplicaciones Android, lo primero que tenemos que elegir, aparte del sistema operativo que vamos a usar, es el entorno de desarrollo con el que trabajaremos. En nuestro caso vamos a poder elegir entre dos entornos totalmente diferentes: Android Studio y Eclipse.

**Nudo:** ¿Crees que es mejor utilizar Android Studio o Eclipse para el desarrollo de aplicaciones Android?

**Desenlace:** Aquí entramos en tema de gustos y puede haber profesionales que prefieran usar uno u otro porque simplemente están acostumbrados a su uso.

Si nos ponemos a investigar un poco, sería mejor usar Android Studio, por una razón muy simple: es el entorno de desarrollo ofrecido por Google para el desarrollo de sus aplicaciones móviles, es decir, es el entorno de desarrollo oficial para Android.

Esto no quiere decir que usar Eclipse sea contraproducente, pero el uso de Android Studio está más generalizado, por lo que se dispondrá de una mayor comunidad en lo que se refiere a ayuda y documentación. Además, debemos considerar un tema importante con respecto a la creación de proyectos. Los proyectos que se crean para el desarrollo de aplicaciones Android en Eclipse y en Android Studio tienen estructuras totalmente diferentes los unos de los otros y, en el caso de que queramos abrir un proyecto de uno, en el otro tendremos que iniciar una operación de migración.

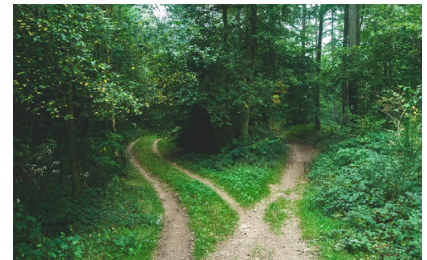


Fig. 4. Siempre tendremos que elegir algún camino.

## / 5. Creación de un emulador integrado

Como todo lenguaje de programación, en Android Studio debemos disponer de alguna forma de probar y depurar nuestras aplicaciones antes de publicarlas.

La peculiaridad de este sistema es que vamos a necesitar un dispositivo Android para poder instalar, probar y depurar nuestras aplicaciones. Para ello, tenemos tres formas de hacerlo:

1. Mediante un **emulador integrado** en Android Studio.
2. Mediante un **emulador de terceros**.
3. Mediante nuestro **propio dispositivo**.

La creación de un emulador integrado en **Android Studio** es muy fácil y rápida y nos va a permitir probar nuestras aplicaciones sin necesidad de utilizar nuestro dispositivo como emulador ni emuladores de terceros.

Para ello, en la **ventana de inicio** de Android Studio seleccionamos la opción **Configure** y luego seleccionamos **AVD Manager**. Aquí nos aparecerá la pantalla de creación de un emulador.

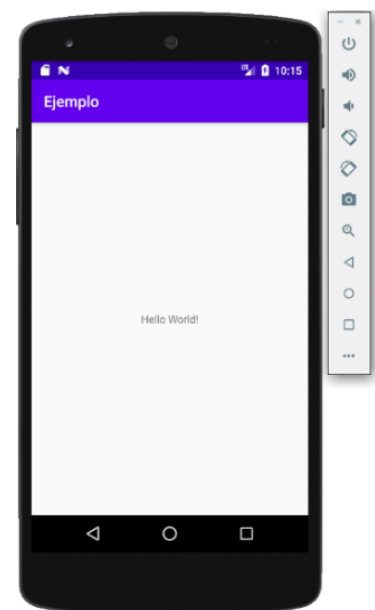


Fig. 5. Emulador Android.



Si pulsamos en el **botón Create Virtual Device...** nos aparecerá el asistente de creación de un dispositivo virtual. En primer lugar, debemos elegir el dispositivo móvil que queremos emular. Contamos con una grandísima variedad de ellos, y diferenciamos entre varias categorías como TV, smartphones, tablets, etc. En esta asignatura elegiremos la categoría «smartphones».

El siguiente paso será **elegir el sistema operativo Android** que queramos, pudiendo elegir entre todos los sistemas operativos que han aparecido en Android, aunque lo más lógico es elegir uno reciente, como Android 8, por ejemplo. En el caso de que no tengamos descargada la imagen del sistema operativo, debemos descargarla. Hay que tener en cuenta que tendremos que elegir la imagen que tenga las Google Apis<sup>1</sup> integradas.



Video 1. "Configuración de un dispositivo móvil"

<https://bit.ly/38RT5BV>



## / 6. Instalación del emulador Android NOX

Como hemos visto anteriormente, una de las formas de probar y depurar nuestras aplicaciones consiste en instalar un emulador de terceros y usarlo como dispositivo de pruebas.

**Emuladores de Android** hay muchísimos: BlueStacks, ARChon, Genymotion, Bliss OS, etc. Sin embargo, nosotros vamos a elegir NOX por su gran facilidad de uso.

**NOX** es un **emulador de dispositivos** Android disponible para Windows y para MacOS. Este emulador se centra en el uso de videojuegos, lo cual nos permite jugar a juegos Android desde nuestro PC sin tener que instalarlos en los dispositivos móviles. Igualmente, se podrá utilizar NOX para emular a un dispositivo físico con el objetivo de probar el funcionamiento de las aplicaciones.

Podemos descargar NOX de su página web: <https://es.bignox.com/>

Una vez descargado, lo único que tendremos que hacer es **instalarlo** como una aplicación normal y corriente. Pero hay que tener cuidado de **no aceptar que nos instale ningún sistema antivirus**, ya que nosotros únicamente queremos el emulador de Android.

Tras instalarlo podremos ejecutarlo y se abrirá con una serie de aplicaciones ya preinstaladas, las cuales no necesitaremos.

En la configuración de NOX podremos hacer que la pantalla esté en horizontal o vertical, según nuestras necesidades.

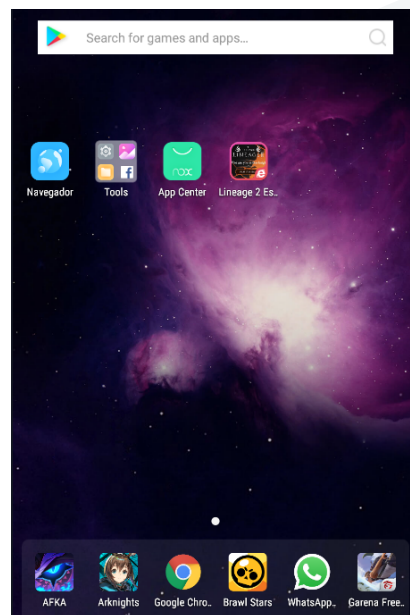


Fig. 6. Emulador NOX.

## / 7. Creación de un proyecto

Para crear un proyecto nuevo en Android Studio debemos seguir los siguientes pasos:

1. **Seleccionamos** 'Start a new Android Studio project'.

1. Para más información sobre las Google Apis: [https://es.wikipedia.org/wiki/API\\_de\\_Google](https://es.wikipedia.org/wiki/API_de_Google)



2. **Indicamos el tipo de actividad** que queremos, en este caso vamos a seleccionar una actividad vacía. Una actividad es una pantalla de una aplicación que se compone de la parte gráfica (escrita en XML) y la parte de código (escrita en Java o en Kotlin). Una aplicación Android tendrá varias actividades, cada una con su parte gráfica y su parte de código. Pulsamos 'siguiente'.

3. **Seleccionamos las SDK mínimas** para que nuestro proyecto funcione, lo cual implica que cuando instalemos nuestra aplicación en un dispositivo móvil este deberá tener una SDK igual o superior a la que indiquemos en este paso. Otra cosa a elegir es el lenguaje de programación: podemos elegir entre Kotlin o Java, y nosotros elegiremos Java. Aquí podremos seleccionar también la ruta donde se guardará nuestro proyecto. Pulsamos en el botón 'Finish' para crear el proyecto.

Tras seguir todos los pasos se creará nuestro proyecto. Después, se sincronizarán todos los ficheros del mismo, lo cual puede tardar unos minutos dependiendo de las características de nuestro ordenador. Hay que tener en cuenta que hasta que no se terminen de sincronizar todos los ficheros del proyecto no podremos empezar a programar nuestra aplicación Android.

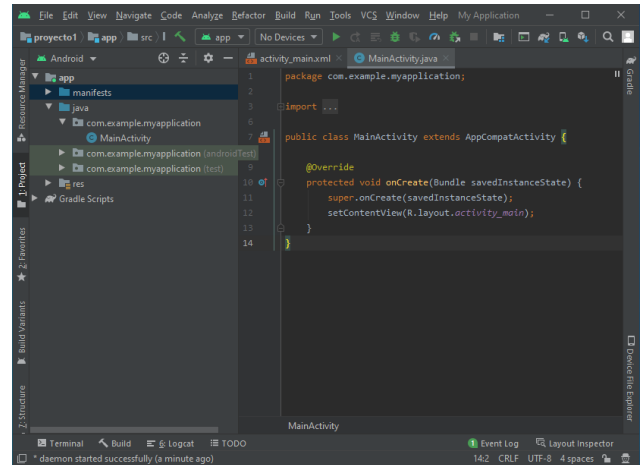


Fig. 7. Proyecto vacío en Android Studio.



Vídeo 2. "Partes de un proyecto Android"

<https://bit.ly/304APBB>



## / 8. Caso práctico 2: "¿Es mejor usar un emulador o depurar con mi dispositivo?"

**Planteamiento:** En este punto, nuestros amigos han visto tres formas diferentes de ejecutar una aplicación en Android Studio: creando un emulador integrado en el propio Android Studio, instalando un emulador de Android como NOX, o configurando los propios dispositivos móviles para la depuración de aplicaciones.

Pilar está un poco confundida le pregunta a José: «Oye, ¿cuál crees tú que es la mejor forma de probar nuestras aplicaciones?». Pero José no sabe qué responderle.

**Nudo:** ¿Qué piensas al respecto? ¿Cuál sería la mejor forma?

**Desenlace:** Las diferentes formas de depurar una aplicación Android no suponen un inconveniente, todo lo contrario: gracias a ello vamos a poder encontrar muchos más fallos, lo cual nos permitirá mejorar el código y así poder explotar al máximo todas las funcionalidades de Android.

Por ejemplo, hay situaciones en las que, por motivos obvios, en un emulador no podremos utilizar ciertas funcionalidades como el acceso al GPS, por ejemplo; en estos casos, podemos depurar nuestra aplicación en nuestro dispositivo móvil y comprobar si el acceso al GPS funciona correctamente. Esto también puede darse si queremos utilizar la cámara del dispositivo, el bluetooth, etc.

Otra gran ventaja de tener diferentes medios para probar es que podremos tener nuestra aplicación en varios dispositivos al mismo tiempo. Por ejemplo, en el emulador de Android Studio y en nuestro dispositivo, pudiendo interactuar entre ellas o intercambiando mensajes si nos encontramos probando una aplicación de mensajería.

Por lo tanto, la mejor forma para probar es combinándolas todas.





## / 9. Ejecución de un proyecto

Para ejecutar y probar nuestra aplicación tenemos dos posibilidades:

1. **Utilizar el emulador de Android** que integra Android Studio o uno de terceros, como es el caso de NOX.
2. **Utilizar nuestro propio dispositivo** móvil para ejecutar la aplicación.

Para lanzar la prueba, tendremos que hacerlo en Android Studio, desde el botón de compilación y ejecución. Es el siguiente:

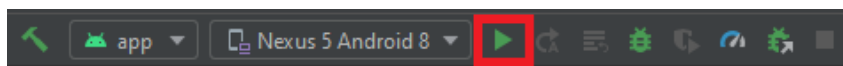


Fig. 8. Botón de ejecución.

Cuando **pulemos el botón de compilar** nuestra aplicación, nos aparecerá una lista con todos los dispositivos (tanto virtuales como físicos) que tenemos disponibles para elegir donde queremos probarla.

Una vez **elegido el dispositivo**, la **aplicación se instalará** y ejecutará automáticamente.

Cuando queramos volver a lanzar la aplicación porque hayamos realizado ciertos cambios en el código, no tendremos que cerrar el dispositivo donde estemos probando, ya que simplemente volviendo a pulsar el botón de compilar la aplicación se cerrará, se compilará, se instalará y se ejecutará automáticamente en el mismo dispositivo.

En el caso de que queramos **cambiar de dispositivo**, sí que tendremos que **cerrar el que tengamos abierto** y elegir el nuevo dispositivo.

En el caso de que queramos ejecutar nuestra aplicación en **varios dispositivos**, podremos elegir la opción 'Run on Multiple Devices' y seleccionar los dispositivos en los que deseemos que se instale nuestra aplicación.

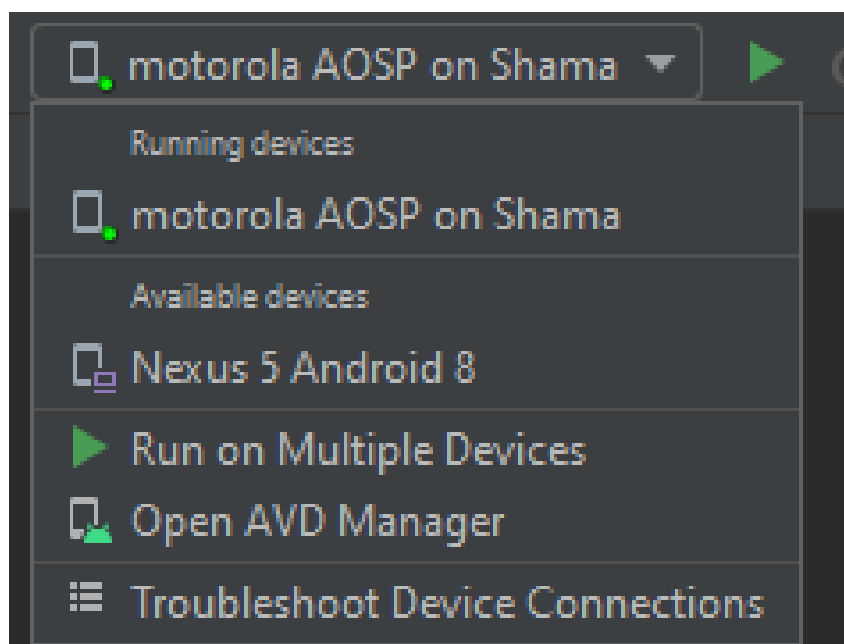


Fig.9. Eligiendo dispositivo.

## Resumen y resolución del caso práctico de la unidad

A lo largo de esta unidad, hemos visto la **estructura de la arquitectura** que usa el sistema operativo móvil **Android**.

También hemos aprendido a instalar un entorno de desarrollo de aplicaciones móviles Android como Android Studio, así como por qué hemos elegido dicho entorno de desarrollo.

A continuación, hemos expuesto cómo podemos crear uno o **varios emuladores** integrados en Android Studio, los cuales nos permitirán depurar nuestras aplicaciones. Hemos instalado también un emulador de Android externo, NOX, y hemos visto cómo podemos configurar nuestro dispositivo móvil para depurar las aplicaciones directamente en él, sin necesidad de emuladores.

Por último, hemos aprendido a crear un proyecto en **Android Studio** desde cero y a cómo ejecutarlo.

### Resolución del caso práctico de la unidad

En el mundo de la programación siempre estaremos aprendiendo cosas nuevas. Por este motivo, cuando nos disponemos a crear aplicaciones móviles es muy normal que nos preguntemos si vamos a tener que aprender otro lenguaje de programación.

En el caso de la programación de **aplicaciones Android**, vamos a usar el **lenguaje de programación Java** y también el lenguaje de marcas **XML**, por lo que todo lo aprendido durante el primer curso en las asignaturas de Programación y Lenguajes de Marcas, va a ser muy importante, ya que será la base sobre la que construiremos los nuevos conocimientos de esta asignatura.

Por otra parte, es cierto que para programar aplicaciones Android podemos utilizar el lenguaje de programación **Kotlin** pero, a efectos prácticos, es más conveniente utilizar Java para adquirir las nociones básicas y luego poder cambiar a Kotlin fácilmente (si así lo deseamos).



Fig. 10. Programando aplicaciones móviles.

## / 10. Bibliografía

Arquitectura Android - Software de Comunicaciones. (s. f.). Arquitectura Android. Recuperado el 20 de junio de 2020 de <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

Platform Architecture |. (s. f.). Android Developers. Recuperado el 20 de junio de 2020 de <https://developer.android.com/guide/platform>

Android Studio. (s.f.). En Wikipedia. Recuperado el 20 de junio de 2020 de [https://es.wikipedia.org/wiki/Android\\_Studio](https://es.wikipedia.org/wiki/Android_Studio).