

# TEMA 1: INTRODUCCIÓN A ANDROID PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

Cicle Formativo de Grado Superior en Desarrollo de Aplicaciones Multiplataforma  
Departamento de Informática // 2º curso 2024-2025  
MARA VAÑÓ ALONSO

**Cicles  
Formatius**



**Reconocimiento – NoComercial – CompartirIgual (by-nc-sa)**

No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

# Índice

Índice.....	3
1. OBJETIVOS DE LA UNIDAD .....	4
2. TECNOLOGÍAS MÓVILES.....	4
2.1. Características de los dispositivos .....	4
2.2. Dispositivos soportados .....	5
2.3. Tecnologías de comunicación .....	6
3. SISTEMAS OPERATIVOS MOVILES .....	7
3.1. Android.....	7
3.2. IOS Apple.....	8
3.3. Otros.....	9
4. LENGUAJES DE PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES.....	9
4.1. Android.....	9
5. FUNDAMENTOS .....	11
5.1. Diferencias entre programación para Desktop y Android.....	11
5.2. Limitaciones que pueden presentar los dispositivos .....	11
6. ENTORNO DE DESARROLLO.....	13
6.1. Android STUDIO .....	13
6.2. Entorno de la aplicación .....	13
BIBLIOGRAFÍA.....	14

## 1. OBJETIVOS DE LA UNIDAD

- Reconocer los distintos tipos de dispositivos móviles existentes en el mercado.
- Tener una visión histórica de su evolución, así como de la evolución tecnológica.
- Conocer los distintos tipos de lenguaje de programación existentes.
- Conocer los fundamentos de las aplicaciones en Android
- Instalar y configurar Android Studio como entorno de desarrollo.

## 2. TECNOLOGÍAS MÓVILES

Por todos es conocido que en el mercado existen diferentes tipos de tecnologías móviles, en este apartado nos aproximamos al concepto de dispositivo móvil, viendo sus características principales, una clasificación de los diferentes tipos de dispositivos móviles existentes, así como los diferentes sistemas operativos que utilizan.

### 2.1. *Características de los dispositivos*

Podemos definir un dispositivo móvil como un aparato con capacidad de procesamiento y almacenamiento de tamaño reducido y que dispone de autonomía de funcionamiento es decir no necesita cableado. Actualmente los teléfonos inteligentes son capaces de simplificar en un solo dispositivo una multitud de tareas para las que hasta ahora necesitamos diferentes dispositivos como calculadora, brújula, cámara de fotos, etc.

A la hora de clasificar los dispositivos móviles lo hacemos fijándonos en cuatro características principales la primera es la capacidad de procesamiento asociada a la capacidad de cálculo y a su capacidad de almacenamiento, por otro lado, los podemos clasificar en función de su tamaño, de su movilidad y de su conectividad.

## 2.2. Dispositivos soportados

En la actualidad, los dispositivos móviles se han extendido de forma exponencial, dando lugar a variantes de estos que se incorporan en nuestro día a día. Podemos encontrarlos con diferentes configuraciones, tipos y características según el uso al que estén destinados.

Podemos encontrar los siguientes tipos:

- **Smartphones y móviles:** este tipo de dispositivo es el más común al que estamos habituados. Entre las características que ofrecen los diferentes fabricantes, podemos encontrar:
  - Tamaño de pantalla.
  - Tipo y número de cámaras.
  - Sensores (GPS, brújula, acelerómetros).
  - Memoria, CPU y batería.
  
- **Smartwatch:** los relojes inteligentes han visto cómo sus funcionalidades han aumentado gracias a la miniaturización y el desarrollo de procesadores más potentes y de bajo consumo, lo que ha permitido crear sistemas con características como Bluetooth, Wifi, acelerómetros, sensores de ritmo de cardíaco, GPS o brújulas. Este tipo de dispositivos se ha abierto camino rápidamente en el mundo del deporte y la salud, gracias a los sensores que incorporan es posible crear aplicaciones que monitorizan nuestra actividad física, como la calidad del sueño. Google ha creado un sistema operativo específico para este tipo de dispositivos llamado WearOS: al estar basado en Android, nos permite crear aplicaciones desde el mismo ecosistema de herramientas.
  
- **Android Auto:** la tecnología móvil no tardaría en llegar a nuestros coches, de modo que nos permite integrar sistemas de navegación y de entretenimiento. Google ha desarrollado un sistema adaptado a las necesidades particulares de los automóviles para evitar las distracciones, de modo que la interfaz de las aplicaciones es minimalista e integrada con el reconocimiento de voz. En este aspecto, los dispositivos soportados están limitados a la integración por parte del fabricante en los modelos, aunque hoy en día es muy común que estas estén en los modelos de serie.

- **Smart TV y Android TV:** aunque no es un dispositivo móvil, se usa el mismo concepto y sistema para los televisores, consiguiendo que se integren con el resto de los dispositivos, lo que permite recibir contenido, tanto vídeo como música, desde dispositivos móviles a la vez que se pueden ejecutar aplicaciones en nuestro televisor.
- El ecosistema de dispositivos que utilizan tecnologías móviles ha crecido en los últimos años, llegando a más dispositivos. Compartir sistema y herramientas de desarrollo hace fácil que los desarrolladores adopten estos nuevos dispositivos, creando aplicaciones para los usuarios y proliferando su crecimiento.

### 2.3. *Tecnologías de comunicación*

La red de comunicación móvil ha ido evolucionando a lo largo del tiempo con diferentes generaciones que van aportando cambios a la anterior. Las diferentes generaciones y sus principales características son: - Generación 0: Puede considerarse en esta generación las primeras comunicaciones entre dispositivos móviles utilizando ondas de radio.

- **Primera generación** (tecnología 1G): la primera red de comunicación móvil automatizada se lanzó en 1979 en Japón tenía problemas en las comunicaciones de voz y limitaciones de seguridad ya que las llamadas de voz se reproducen en las torres de radio.
- **Segunda generación** (tecnología 2G): Años 90. Fue la primera en facilitar voz y datos digitales, así como roaming internacional. - Segunda generación y media (tecnología 2.5G-2.75G: Disponible a partir de los años 2000 fue la primera en introducir en la red paquetes para proporcionar transferencia e Internet de alta velocidad de datos.
- **Tercera generación** (tecnología 3G): Su principal objetivo fue ofrecer un aumento en las tasas de datos facilitando el crecimiento y ofreciendo una mayor capacidad de voz y de datos además se ofrecía soporte a diferentes aplicaciones y se disminuía el coste de las transmisiones.
- **Cuarta generación** (tecnología 4G): El principal objetivo es proporcionar alta velocidad alta calidad alta capacidad y todo esto a unos precios más ajustados tanto para voz como para datos, además se incrementa la seguridad de las comunicaciones.
- **Quinta generación** (tecnología 5G): Está tecnología que inicio su lanzamiento comercial en 2019 es capaz de multiplicar por 10 la velocidad del 4G llegando

a 1Gbps de velocidad de transmisión.

- **Sexta generación** (tecnología 6G): Su principal objetivo será reducir la latencia (si quieres saber que es la latencia haz click aquí) de las conexiones y aumentar la velocidad de transmisión. Se estima que su comercialización será en 2030 y que los primeros casos de uso reales podrían llegar a partir de 2026. Además, podemos considerar el Wi-Fi o el Bluetooth ya que, por ejemplo, los dispositivos como Tablets o Wearables generalmente no disponen de una tarifa de datos propia por lo que utilizan conexiones de datos por Wi-Fi o incluso la transferencia de archivos por Bluetooth (en campo cercano).

### 3. SISTEMAS OPERATIVOS MOVILES

El sistema operativo es el conjunto de programas que permiten abstraer el hardware y que provee servicios a las aplicaciones que se ejecutan sobre él. Generalmente están basados en el modelo de capas, entre las que podemos mencionar:

- **Núcleo o kernel:** permite el acceso a los diferentes elementos de hardware que conforman el dispositivo.
- **Middleware:** es el conjunto de módulos software que hacen posible la existencia de las propias aplicaciones para móviles. Es totalmente transparente para el usuario final y proporciona servicios como el motor de mensajería, servicios de seguridad o códecs multimedia.
- **Entorno de ejecución de aplicaciones:** se trata de un gestor de aplicaciones y un conjunto de interfaces programables (APIs) abiertas y accesibles por los programadores para la creación de las diferentes aplicaciones.
- **Interfaz de usuario:** realizan la presentación visual de la app. Incluyen los componentes gráficos
- (pantallas, botones...) y el marco de interacción.
- **Aplicaciones nativas:** son aquellas que están desarrolladas para un equipo o plataforma determinada. Funciona en el equipo sin necesidad de ningún programa externo.

#### 3.1. Android

Android es un sistema operativo de última generación basado en Linux creado por Google para sus dispositivos. Está basada en la tecnología DMI (Direct Manipulation Interface) como un tipo de interacción hombre-máquina que representará los

objetos gráficos en la pantalla de forma rápida.

Este sistema operativo está diseñado principalmente para smartphones o tablets pero también puede utilizarse en televisores inteligentes así como en cámaras digitales y otros dispositivos electrónicos. Aunque la base es de software libre podemos encontrar dispositivos que utilizan una combinación de ambos software libre y software propietario. Aunque el desarrollo de Android comenzó en 2003, no se lanzó una primera versión comercial hasta 2008, a día de hoy es el sistema operativo móvil más utilizado con más de tres mil millones de usuarios activos.

A día de hoy (septiembre de 2022), la última versión es Android 13, que introduce mejoras de privacidad, nuevas APIs para desarrolladores y un rediseño casi total de la interfaz de usuario del sistema, a través de Material You, otorgando al usuario el control sobre la apariencia de la interfaz.

Una de las mayores dificultades que encontramos al programar para Android es tratar de dar soporte el mayor número posible de APIs. cuando desarrollemos deberemos especificar una versión mínima de API para la que funcionará nuestra aplicación y una versión máxima, con estos parámetros se definirá el rango de versiones para las que funcionará la aplicación. Esto se define con los parámetros `android:minSdkVersion` y `android:maxSdkVersion`.



### 3.2. *IOS Apple*

Fue desarrollado por Apple, y originariamente se denominaba iPhone OS, siendo un derivado de Mac OS X. Se trata de un sistema operativo basado en la manipulación directa cuya arquitectura también está basada en capas





### 3.3. Otros

Además existen en el mercado otros sistemas operativos desarrollados por diferentes compañías que han ido surgiendo a lo largo de los años con mayor o menor éxito como es el caso de Windows Phone desarrollado por Microsoft, BlackBerry OS desarrollado por la RIM (Research In Motion), Symbian, desarrollado por una alianza de varias empresas de telefonía móvil como SAMSUNG, SIEMENS o Nokia, Ubuntu estamos yendo a por Linux Touch o Firefox OS. Destacar también Harmony OS, un sistema operativo desarrollado por Huawei Technologies especialmente diseñado para el interconexión de internet de las cosas.

Todos ellos quedan obsoletos y en la práctica únicamente tanto Android como iOS ocupan todo el mercado.

## 4. LENGUAJES DE PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES

A la hora de programar para un dispositivo móvil podríamos optar por realizar el desarrollo nativo un desarrollo multiplataforma compilado o un desarrollo multiplataforma basado en HTML5, a lo largo de este módulo nos centraremos en la programación para Android por ser como ya hemos visto el sistema operativo con mayor penetración de mercado y que además se basa en software libre.

### 4.1. Programación en Android

En este tema traduciré a la programación en Android, aunque podría programarse a través de Netbeans utilizando el plugin NBPPlugin o con ADT de Google con Eclipse, en este módulo se utilizará Android Studio para el desarrollo de las diferentes aplicaciones y actividades. Es importante destacar que no es necesario

disponer de un dispositivo Android para seguir este módulo utilizaremos un emulador que estará proporcionado por el SDK de Android que funciona con una imagen del software de Android.

Kotlin ha sido nombrado, junto con Java, lenguaje oficial para el desarrollo en Android. Es de prever que Google va a mantener ambos lenguajes.

### Ventajas en Java

- Java es más sencillo y rápido de aprender que Kotlin. El requisito fundamental cuando se diseñó Java era que fuera lo más simple posible.
- Java es el lenguaje de programación más popular del mundo. Aprender Java te abrirá muchas puertas en el mundo laboral.
- La mayoría de los ejemplos, o consultas en StackOverflow están en Java. Mucha de la documentación oficial de Google está disponible solo en Java.

### Ventajas de Kotlin

- Es conciso y expresivo: vas a tener que escribir mucho menos para hacer lo mismo. Además, el código resulta mucho más fácil de entender.
- Es un lenguaje moderno: incorpora novedades como acceso directo a propiedades, lambdas, funciones de extensión, clases de datos, funciones inline, soporte nativo de delegados, y un largo etcétera.

Si eres un **programador novato**: Te recomiendo que comiences por Java. Cuando lo domines puedes plantearte pasarte a Kotlin. No te preocupes que no habrá sido tiempo perdido.

Si eres un **programador experimentado o ya conoces Java**: Te recomiendo que te enfrentes a este nuevo lenguaje. Tiene una curva de aprendizaje rápida y ya verás que acaba enganchando.

## 5. FUNDAMENTOS

Como ya se ha comentado Android es un sistema operativo basado en Linux, dónde cada aplicación tiene un propietario, es decir el sistema asignará un identificador único a cada aplicación de tal forma que solo el propietario podrá tener acceso a los archivos de la misma. Esto significa que Android utiliza el principio de menor privilegio otorgando los permisos asociados a la aplicación en el momento de la instalación de la misma o incluso en el momento de utilizar la aplicación. Así, podemos decir que Android es un sistema operativo seguro.

### 5.1. *Diferencias entre programación para Desktop y Android*

A la hora de programar para dispositivos Android lo primero que debemos tener en cuenta es que un dispositivo Android será un dispositivo móvil y pequeño con una capacidad de procesamiento más limitado, además debemos contar con pantallas táctiles y de menor tamaño que en los equipos de sobremesa, teclados táctiles y que aparecen y desaparecen sobre la pantalla ( siendo además de reducido tamaño), y que además son manejados en situaciones cotidianas de la vida ( el usuario no está sentado utilizando su dispositivo si no que puede estar caminando, viendo la televisión, etc), por otro lado, las conexiones móviles pueden sufrir interrupciones y por último, los recursos del dispositivo son limitados (especialmente en cuanto a procesamiento de datos y capacidad de batería) por lo que a la hora de diseñar nuestras aplicaciones deberemos tener en cuenta estos aspectos.

### 5.2. *Limitaciones que pueden presentar los dispositivos*

Los dispositivos móviles plantean serias limitaciones a la hora de ejecutar aplicaciones que debemos tener en consideración durante el desarrollo de la aplicación para tener un producto de calidad y robusto. Entre ellas, definimos las más relevantes:

- **Desconexión:** Al tratarse de dispositivos que podemos llevar con nosotros en cualquier momento, lugar y permanentemente conectados, estos pueden sufrir desconexiones tanto de forma total como parcial, lo que, en el caso de las aplicaciones que utilicen datos de un servidor, provocará que se vean afectadas por

la variabilidad de la conexión. La aplicación que desarrollemos debe controlar esta situación y poder ofrecer un mecanismo que asegure, en la medida de lo posible, la disponibilidad de los datos.

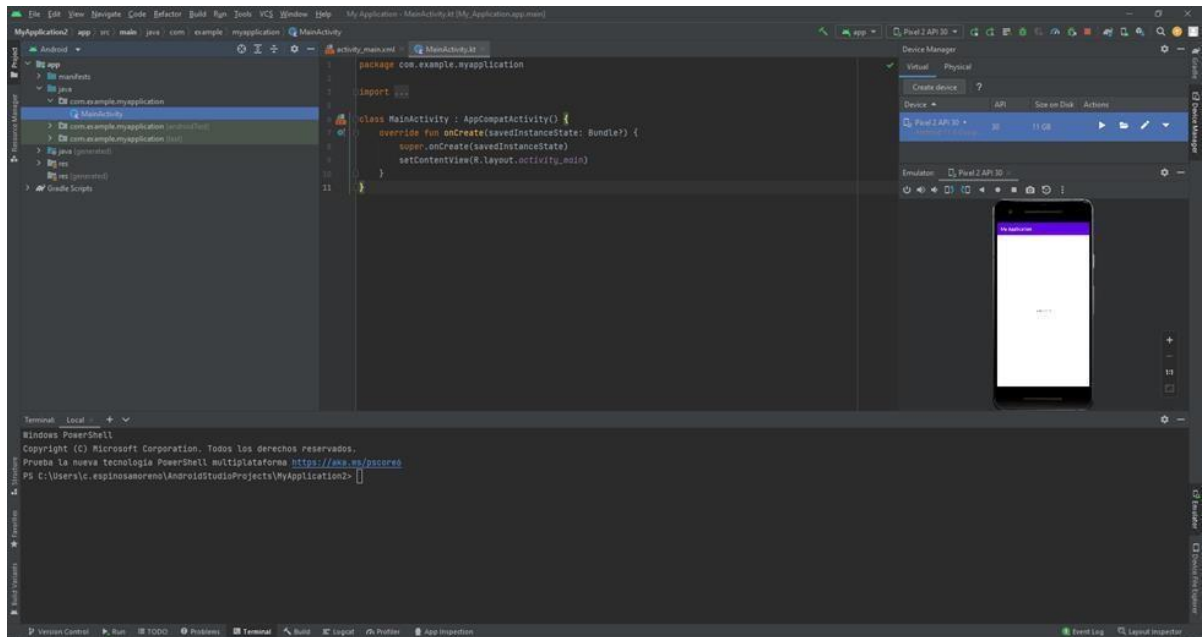
- **Seguridad:** En lo que se refiere seguridad física, la limitación viene dada por la propia virtud del dispositivo: al ser ligeros y fáciles de llevar, son a la vez susceptibles de ser más fácilmente sustraídos. En el ámbito de la seguridad de la información, este tipo de dispositivos son más vulnerables, al permitir conectarse a redes poco seguras o la posibilidad de instalar aplicaciones de dudosa procedencia, unido a la gran cantidad de sensores, como cámaras, GPS o micrófonos, de que disponen. En el desarrollo de aplicaciones debemos considerar usar los permisos mínimos necesarios para el funcionamiento de la aplicación. En el caso de que desarrollemos una aplicación con uso de datos, siempre deben viajar bajo una conexión segura o cifrada para evitar que los datos sean expuestos.
- **Consumo de batería:** Por su propia naturaleza de ser lo más ligeros posible, la batería suele ser uno de los elementos más comprometidos, y el diseño de las aplicaciones debe tener en consideración qué recursos utiliza en cada momento, liberando aquellos que no sean necesarios para ahorrar los tan preciados miliamperios de la batería.
- **Memoria y almacenamiento:** A día de hoy, los dispositivos móviles cada vez incorporan más memoria y capacidad de almacenamiento, aunque no por ello debemos desaprovecharla, ya que es un recurso limitado que no podemos ampliar sin adquirir un nuevo dispositivo. Por ello, las aplicaciones móviles deben optimizar el uso de los recursos, ya que también debe compartirlos con otras aplicaciones. Otros factores derivados del propio dispositivo: como pueden ser falta de potencia de procesamiento de aplicaciones, deterioro, desactualización, falta de mantenimiento, recursos limitados comparado con otros dispositivos.

## 6. ENTORNO DE DESARROLLO

### 6.1. *Android STUDIO*

Descarga Android Studio desde la web oficial en el siguiente enlace: <https://developer.android.com/studio>

### 6.2. *Entorno de la aplicación*



**Editor de código:** en esta sección podremos escribir y modificar el código de los diferentes archivos que componen la aplicación. Esta sección puede cambiar dependiendo del tipo de aplicación que estemos desarrollando.

**Herramientas de compilación y ejecución:** en esta barra podemos ver en qué dispositivo ejecutamos la aplicación para comprobar nuestro código.

**Barra de estado:** nos mostrará los mensajes de información y advertencias que se muestran en los diferentes procesos que ejecuta el IDE.

**Ventanas de herramientas:** cambian en función de la herramienta que hemos seleccionado, en ellas podemos encontrar la administración del proyecto.

**Barra de ventana de herramientas:** permite expandir o contraer las ventanas de herramientas.

## BIBLIOGRAFÍA

- (1) Android. Programación Multimedia y de dispositivos móviles. Garceta.
- (2) Programación Multimedia y Dispositivos Móviles. Editorial Síntesis
- (3) Android App Development For Dummies
- (4) Beginning Android Programming with Android Studio. Wrox
- (5) <https://www.cyberark.com/es/what-is/least-privilege/>
- (6) <https://academiaandroid.com/>
- (7) <https://www.gsoft.es/articulos/que-necesito-web-apps-app-nativa-o-app-hibrida/>
- (8) <https://developer.android.com/>