

# ANDROID - INTRODUCCIÓN A ANDROID

---

## INTRODUCCIÓN

### CONTENIDO

<b>I</b>	<b>Introducción</b>	<b>4</b>
I.1	Objetivos . . . . .	4
<b>II</b>	<b>Plataforma Android</b>	<b>6</b>
II.1	Introducción . . . . .	6
II.2	Orígenes . . . . .	8
II.3	Arquitectura de Android . . . . .	9
II.3.1	El núcleo linux . . . . .	10
II.3.2	Runtime de Android . . . . .	10
II.3.3	Librerías nativas . . . . .	11
II.3.4	Entorno de aplicación . . . . .	12
II.3.5	Aplicaciones . . . . .	13
<b>III</b>	<b>Versiones Android</b>	<b>14</b>
III.1	Introducción . . . . .	14

III.2	Las primeras versiones . . . . .	15
III.2.1	Cupcake . . . . .	15
III.2.2	Donut . . . . .	16
III.2.3	Éclair . . . . .	16
III.2.4	Froyo . . . . .	17
III.2.5	Gingerbread . . . . .	18
III.2.6	Honeycomb . . . . .	20
III.2.7	Ice Cream Sandwich . . . . .	21
III.2.8	Jelly Bean . . . . .	23
III.2.9	KitKat . . . . .	24
III.2.10	Lollipop . . . . .	26
III.2.11	Marshmallow . . . . .	28
III.2.12	Android Nougat . . . . .	29
III.2.13	Android Oreo . . . . .	31
III.2.14	Android Pie . . . . .	33
III.2.15	Android 10 . . . . .	34

# Part I.

## Introducción

### INTRODUCCIÓN

La aparición de los dispositivos móviles ha transformado la sociedad. En poco años se ha evolucionado en el uso del móvil, pasando de un uso esporádico para la comunicación (voz y mensajes) a un uso intensivo del dispositivo, incluyendo otros mecanismos de comunicación (redes sociales, etc). Esta evolución ha hecho que todo el mundo tenga móvil y generalmente le dé otro uso además de hablar por él. Realizar aplicaciones para un dispositivo que se encuentra en el bolsillo de cualquier persona nos proporciona unas aplicaciones casi infinitas. Podemos afirmar que el nuevo ordenador personal del siglo XXI será un terminal móvil.

Hay otro punto a tener en cuenta, con la capacidades de los nuevos dispositivos se está desarrollando una nueva revolución: el uso de los móviles como plataforma de juegos. Hoy en día la capacidad de los móviles y de las tabletas es similar a la de los ordenadores, las pantallas han crecido en tamaño y resolución dotando a estos nuevos dispositivos de características muy apetecibles para jugar. Se espera una explosión exponencial en la aparición de juegos para estas nuevas plataformas, sobre todo pequeños juegos que no dure mucho la partida.

### I.1 OBJETIVOS

- Reconocer los tipos de dispositivos móviles, sus características y limitaciones.
- Identificar la evolución de la tecnología de comunicación móvil.
- Tener una visión histórica de la evolución de los dispositivos móviles.

- Conocer los sistemas operativos existentes para móviles.
  - Familiarizarse con los tipos y variedades de lenguajes de programación.
-

# Part II.

## Plataforma Android

### OBJETIVOS

- Enumerar las características más importantes de Android.
  - Describir sus orígenes.
  - Comparar las plataformas para móviles más utilizadas en la actualidad.
  - Describir la arquitectura de Android y enumerar las características de cada una de sus capas.
- 

### II.1 INTRODUCCIÓN

Existen muchas plataformas para móviles (Apple iOS, Windows Phone, BlackBerry, Palm, Java Micro Edition, Linux Mobile (LiMo), Firefox OS, etc.); sin embargo, Android presenta una serie de características que lo hacen diferente. Es el primero que combina en una misma solución las siguientes cualidades:

- **Plataforma abierta.** Es una plataforma de desarrollo libre basada en Linux y de código abierto. Una de sus grandes ventajas es que se puede usar y customizar el sistema sin pagar royalties.
- **Adaptable a diversos tipos de hardware.** Android no ha sido diseñado exclusivamente para su uso en teléfonos y tabletas. Hoy en día podemos encontrar relojes, gafas, cámaras, TV, sistema para automóviles, electrodomésticos y una gran variedad de sistemas empotrados que se basan en este sistema operativo, lo cual tiene

sus evidentes ventajas, pero también va a suponer un esfuerzo adicional para el programador. La aplicación ha de funcionar correctamente en dispositivos con una gran variedad de tipos de entrada, pantalla, memoria, etc. Esta característica contrasta con la estrategia de Apple: en iOS tenemos que desarrollar una aplicación para iPhone y otra diferente para iPad.

- **Portabilidad asegurada.** Las aplicaciones finales son desarrolladas en Java, lo que nos asegura que podrán ser ejecutadas en cualquier tipo de CPU, tanto presente como futuro. Esto se consigue gracias al concepto de máquina virtual.
- **Arquitectura basada en componentes inspirados en Internet.** Por ejemplo, el diseño de la interfaz de usuario se hace en XML, lo que permite que una misma aplicación se ejecute en un reloj de pantalla reducida o en un televisor.
- **Filosofía de dispositivo siempre conectado a Internet.** Muchas aplicaciones solo funcionan si disponemos de una conexión permanente a Internet. Por ejemplo, comunicaciones interpersonales o navegación con mapas.
- **Gran cantidad de servicios incorporados.** Por ejemplo, localización basada tanto en GPS como en redes, bases de datos con SQL, reconocimiento y síntesis de voz, navegador, multimedia, etc.
- **Aceptable nivel de seguridad.** Los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja, que hereda de Linux. Además, cada aplicación dispone de una serie de permisos que limitan su rango de actuación (servicios de localización, acceso a Internet, etc.). Desde la versión 6.0 el usuario puede conceder o retirar permisos a las aplicaciones en cualquier momento.
- **Optimizado para baja potencia y poca memoria.** En el diseño de Android se ha tenido en cuenta el hardware específico de los dispositivos móviles. Por ejemplo, Android utiliza la máquina virtual ART (o Dalvik en versiones antiguas). Se trata de una implementación de Google de la máquina virtual Java optimizada para dispositivos móviles.

- **Alta calidad de gráficos y sonido.** Gráficos vectoriales suavizados, animaciones, gráficos en 3D basados en OpenGL. Incorpora los codecs estándares más comunes de audio y vídeo, incluyendo H.264 (AVC), MP3, AAC, etc.

Como hemos visto, Android combina características muy interesantes. No obstante, la pregunta del millón es: ¿se convertirá Android en el sistema operativo (SO) estándar para dispositivos móviles? Para contestar a esta pregunta habrá que ver la evolución del iPhone de Apple y cuál es la respuesta de Windows con el lanzamiento de su SO para móviles. No obstante, Android ha alcanzado un 85% de cuota de mercado (90% en España), cosa que lo deja en una posición predominante que es difícil que pierda a corto plazo.

En conclusión, Android nos ofrece una forma sencilla y novedosa de implementar potentes aplicaciones para diferentes tipos de dispositivos. A lo largo de este texto trataremos de mostrar de la forma más sencilla posible cómo conseguirlo.

## II.2 ORÍGENES

Google adquiere Android Inc. en el año 2005. Se trataba de una pequeña compañía, recién creada, orientada a la producción de aplicaciones para terminales móviles. Ese mismo año empiezan a trabajar en la creación de una máquina virtual Java optimizada para móviles (Dalvik VM).

En el año 2007 se crea el consorcio Handset Alliance<sup>[1]</sup> con el objetivo de desarrollar estándares abiertos para móviles. Está formado por Google, Intel, Texas Instruments, Motorola, T-Mobile, Samsung, Ericson, Toshiba, Vodafone, NTT DoCoMo, Sprint Nextel y otros. Una pieza clave de los objetivos de esta alianza es promover el diseño y difusión de la plataforma Android. Sus miembros se han comprometido a publicar una parte importante de su propiedad intelectual como código abierto bajo licencia Apache v2.0.

En noviembre de 2007 se lanza una primera versión del Android SDK. Al año siguiente aparece el primer móvil con Android (T-Mobile G1). En octubre, Google libera el código fuente de Android, principalmente bajo licencia de código abierto Apache (licen-

cia GPL v2 para el núcleo). Ese mismo mes se abre Android Market, para la descarga de aplicaciones. En abril de 2009, Google lanza la versión 1.5 del SDK, que incorpora nuevas características como el teclado en pantalla. A finales de 2009 se lanza la versión 2.0 y a lo largo de 2010, las versiones 2.1, 2.2 y 2.3. Durante el año 2010, Android se consolida como uno de los sistemas operativos para móviles más utilizados, con resultados cercanos a iOS e incluso superando al sistema de Apple en EE.UU.

En el año 2011 se lanza la versión 3.x (Honeycomb), específica para tabletas, y la 4.0 (Ice Cream Sandwich), tanto para móviles como para tabletas. Durante ese año, Android se consolida como la plataforma para móviles más importante y alcanza una cuota de mercado superior al 50.

En 2012, Google cambia su estrategia en su tienda de descargas online, reemplazando Android Market por Google Play Store, donde en un solo portal unifica tanto la descarga de aplicaciones como la de contenidos. Ese año aparecen las versiones 4.1 y 4.2 (Jelly Bean). Android mantiene su espectacular crecimiento y alcanza, a finales de año, una cuota de mercado del 70.

En 2013 se lanzan las versiones 4.3 y 4.4 (KitKat). En 2014 se lanza la versión 5.0 (Lollipop). A finales de ese año, la cuota de mercado de Android llega al 85 %. En octubre de 2015 ha aparecido la versión 6.0, con el nombre de Marshmallow. En 2016 se lanzó la versión 7.0, Android Nougat. A finales de 2017 aparece la versión 8.0, con nombre Android Oreo. En agosto de 2018 se lanza la versión 9.0, Android Pie.

## II.3 ARQUITECTURA DE ANDROID

El siguiente gráfico muestra la arquitectura de Android. Como se puede ver está formada por cuatro capas. Una de las características más importantes es que todas las capas están basadas en *software* libre.



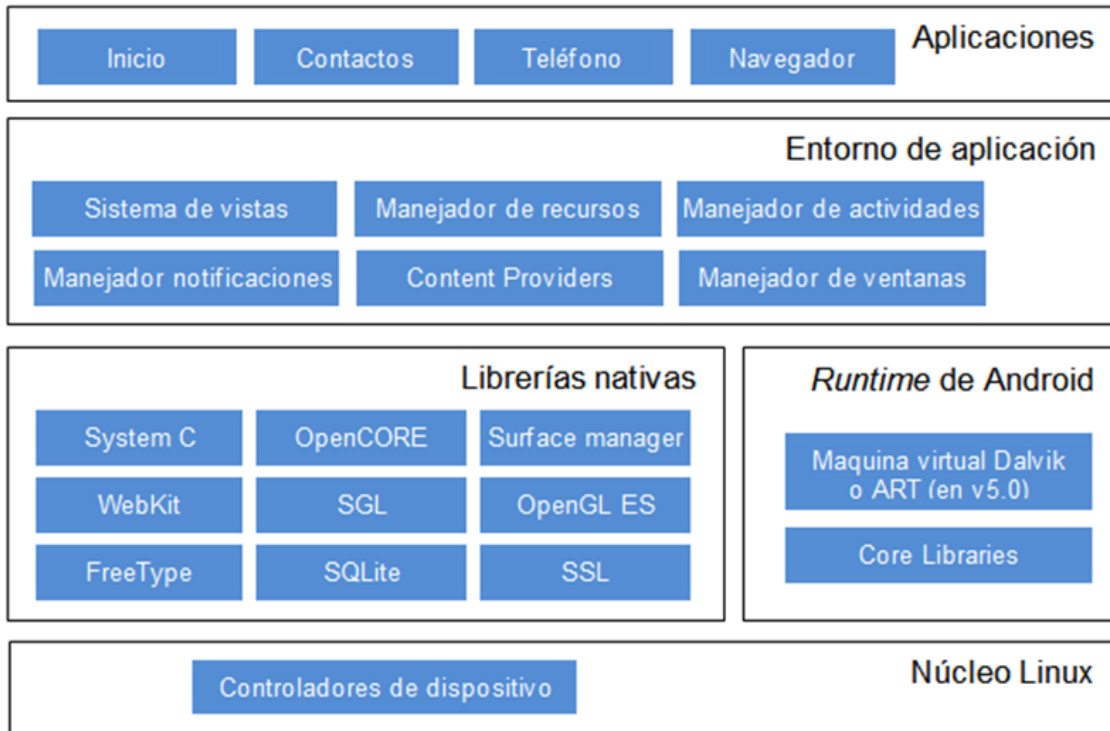


Figure 1: Arquitectura de Android

### 11.3.1 El núcleo linux

El núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos.

Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

### 11.3.2 Runtime de Android

Está basado en el concepto de máquina virtual utilizado en Java. Dadas las limitaciones de los dispositivos donde ha de correr Android (poca memoria y procesador limitado), no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones.

Entre las características de la máquina virtual Dalvik que facilitan esta optimización de recursos se encuentra la ejecución de ficheros Dalvik ejecutables (.dex) –formato optimizado para ahorrar memoria–. Además, está basada en registros. Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik. Delega al kernel de Linux algunas funciones como *threading* y el manejo de la memoria a bajo nivel.

A partir de Android 5.0 se reemplaza Dalvik por ART. Esta nueva máquina virtual consigue reducir el tiempo de ejecución del código Java hasta en un 33

También se incluye en el *runtime* de Android el módulo Core Libraries, con la mayoría de las librerías disponibles en el lenguaje Java.

### 11.3.3 Librerías nativas

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:

- **System C library:** una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
- **Media Framework:** librería basada en OpenCORE de PacketVideo. Soporta codecs de reproducción y grabación de multitud de formatos de audio y vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- **Surface Manager:** maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- **WebKit/Chromium:** soporta un moderno navegador Web utilizado en el navegador Android y en la vista Webview. En la versión 4.4, WebKit ha sido reemplazada por Chromium/Blink, que es la base del navegador Chrome de Google.
- **SGL:** motor de gráficos 2D.

- **Librerías 3D:** implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- **FreeType:** fuentes en bitmap y renderizado vectorial.
- **SQLite:** potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- **SSL:** proporciona servicios de encriptación Secure Socket Layer (capa de conexión segura).

#### 11.3.4 Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones, etc.).

Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas (sujetas a las restricciones de seguridad). Este mismo mecanismo permite a los usuarios reemplazar componentes.

Los servicios más importantes que incluye son:

- **Views:** extenso conjunto de vistas, (parte visual de los componentes).
- **Resource Manager:** proporciona acceso a recursos que no son en código.
- **Activity Manager:** maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- **Notification Manager:** permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- **Content Providers:** mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

Una de las mayores fortalezas del entorno de aplicación de Android es que se aprovecha el lenguaje de programación Java. El SDK de Android no acaba de ofrecer para su estándar todo lo disponible del entorno de ejecución Java (JRE), pero es compatible con una fracción muy significativa de este

### 11.3.5 Aplicaciones

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de correr en la máquina virtual Dalvik para garantizar la seguridad del sistema.

Normalmente las aplicaciones Android están escritas en Java o Kotlin. Para desarrollar este tipo de aplicaciones podemos utilizar el Android SDK.

# Part III.

## Versiones Android

### OBJETIVOS

- Enumerar las diferentes versiones que han aparecido de Android y describir sus aportaciones.
- 

### III.1 INTRODUCCIÓN

Antes de empezar a programar en Android hay que elegir la versión del sistema para la que deseamos realizar la aplicación. Es muy importante observar que hay clases y métodos que están disponibles a partir de una versión; si las vamos a usar, hemos de conocer la versión mínima necesaria.

Cuando se ha lanzado una nueva plataforma, siempre ha sido compatible con las versiones anteriores. Es decir, solo se añaden nuevas funcionalidades, y en el caso de modificar alguna funcionalidad, no se elimina, sino que se etiqueta como obsoleta, pero se puede continuar utilizando.

A continuación se describen las plataformas lanzadas hasta la fecha, con una breve descripción de las novedades introducidas. Las plataformas se identifican de tres formas alternativas: versión, nivel de API y nombre comercial. El nivel de API corresponde a números enteros, comenzando desde 1. Para los nombres comerciales se han elegido postres en orden alfabético: Cupcake (v1.5), Donut (v1.6), Éclair (v2.0), Froyo (v2.2), Gingerbread (v2.3), etc. Las dos primeras versiones, que hubieran correspondido a las letras A y B, no recibieron nombre.

## III.2 LAS PRIMERAS VERSIONES

**ANDROID 1-0 NIVEL API 1 (SEPTIEMBRE 2008)** Primera versión de Android. Nunca se utilizó comercialmente, por lo que no tiene mucho sentido desarrollar para esta plataforma.

**ANDROID 1-2 NIVEL API 2 (SEPTIEMBRE 2009)** No se añadieron apenas funcionalidades: simplemente se arreglaron algunos errores de la versión anterior. Es la opción a escoger si queremos desarrollar una aplicación compatible con todos los dispositivos Android. No obstante, apenas existen usuarios con esta versión.

### III.2.1 Cupcake

**ANDROID 1.5 NIVEL DE API 3 (ABRIL 2009)** Es la primera versión con algún usuario, aunque en la actualidad apenas quedan. Como novedades, se incorpora la posibilidad de teclado en pantalla con predicción de texto (ya no es necesario que los terminales tengan un teclado físico), así como la capacidad de grabación avanzada de audio y vídeo. También aparecen los widgets de escritorio y live folders. Incorpora soporte para Bluetooth estéreo, por lo que permite conectarse automáticamente a auriculares Bluetooth. Las transiciones entre ventanas se realizan mediante animaciones.



Figure 2: Cupcake

### III.2.2 Donut

**ANDROID 1.6 NIVEL DE API 4 (SEPTIEMBRE 2009)** Permite capacidades de búsqueda avanzada en todo el dispositivo. También se incorpora *gestures* y la síntesis de texto a voz. Asimismo, se facilita que una aplicación pueda trabajar con diferentes densidades de pantalla. Soporte para resolución de pantallas WVGA. Aparece un nuevo atributo XML, *onClick*, que puede especificarse en una vista. Soporte para CDMA/EVDO, 802.1x y VPNs



Figure 3: Donut

### III.2.3 Éclair

**ANDROID 2.0 NIVEL DE API 5 (OCTUBRE 2009)** Esta versión de API apenas cuenta con usuarios, dado que la mayoría de fabricantes pasaron directamente de la versión 1.6 a la 2.1. Como novedades cabría destacar que incorpora un API para manejar el bluetooth 2.1. Nueva funcionalidad que permite sincronizar adaptadores para conectarlo a cualquier dispositivo. Ofrece un servicio centralizado de manejo de cuentas. Mejora la gestión de contactos y ofrece más ajustes en la cámara. Se ha optimizado la velocidad de hardware. Se aumenta el número de tamaños de ventana y resoluciones soportadas. Nueva interfaz del navegador y soporte para HTML5. Mejoras en el calendario y soporte para Microsoft Exchange. La clase *MotionEvent* ahora soporta eventos en pantallas multitáctil.



Figure 4: Éclair

**ANDROID 2.1 NIVEL DE API 7 (ENERO 2010)** Se considera una actualización menor, por lo que la siguieron llamando Éclair. Destacamos el reconocimiento de voz, que permite introducir un campo de texto dictando sin necesidad de utilizar el teclado. También permite desarrollar fondos de pantalla animados. Se puede obtener información sobre la señal de la red actual que posea el dispositivo. En el paquete WebKit se incluyen nuevos métodos para manipular bases de datos almacenadas en Internet.

#### III.2.4 Froyo

**ANDROID 2.2 NIVEL DE API 8 (MAYO 2010)** Como característica más destacada se puede indicar la mejora de velocidad de ejecución de las aplicaciones (ejecución del código de la CPU de 2 a 5 veces más rápido que en la versión 2.1 de acuerdo a varios *benchmarks*). Esto se consigue con la introducción de un nuevo compilador JIT de la máquina Dalvik.

Se añaden varias mejoras relacionadas con el navegador web, como el soporte de Adobe Flash 10.1 y la incorporación del motor Javascript V8 utilizado en Chrome.

El desarrollo de aplicaciones permite las siguientes novedades: se puede preguntar al usuario si desea instalar una aplicación en un medio de almacenamiento externo (como una tarjeta SD), como alternativa a la instalación en la memoria interna del dispositivo; las aplicaciones se actualizan de forma automática cuando aparece una nueva versión; proporciona un servicio para la copia de seguridad de datos que se puede realizar desde la propia aplicación para garantizar al usuario el mantenimiento de sus datos; y por





Figure 5: Froyo

último, se facilita que las aplicaciones interaccionen con el reconocimiento de voz y que terceras partes proporcionen nuevos motores de reconocimiento.

Se mejora la conectividad: ahora podemos utilizar nuestro teléfono para dar acceso a Internet a otros dispositivos (*tethering*), tanto por USB como por Wi-Fi. También se añade el soporte a Wi-Fi IEEE 802.11n y notificaciones *push*.

Se añaden varias mejoras en diferentes componentes: en la API gráfica OpenGL ES; por ejemplo, se pasa a soportar la versión 2.0. Para finalizar, permite definir modos de interfaz de usuario («automóvil» y «noche») para que las aplicaciones se configuren según el modo seleccionado por el usuario.

### III.2.5 Gingerbread

**ANDROID 2.3 NIVEL DE API 9 (DICIEMBRE 2010)** Debido al éxito de Android en las nuevas tabletas ahora soporta mayores tamaños de pantalla y resoluciones (WXGA y superiores).

Incorpora un nuevo interfaz de usuario con un diseño actualizado. Dentro de las mejoras de la interfaz de usuario destacamos la mejora de la funcionalidad de cortar, copiar y pegar y un teclado en pantalla con capacidad multitáctil. Se incluye soporte



Figure 6: Gingerbread

nativo para varias cámaras, pensado en la segunda cámara usada en videoconferencia. La incorporación de esta segunda cámara ha propiciado la inclusión de reconocimiento facial para identificar el usuario del terminal.

La máquina virtual de Dalvik introduce un nuevo recolector de basura que minimiza las pausas de la aplicación, ayudando a garantizar una mejor animación y el aumento de la capacidad de respuesta en juegos y aplicaciones similares. Se trata de corregir así una de las lacras de este sistema operativo móvil, que en versiones previas no ha sido capaz de cerrar bien las aplicaciones en desuso. Se dispone de mayor apoyo para el desarrollo de código nativo (NDK). También se mejora la gestión de energía y control de aplicaciones. Y se cambia el sistema de ficheros, que pasa de YAFFS a ext4.

Entre otras novedades destacamos en soporte nativo para telefonía sobre Internet VoIP/SIP. El soporte para reproducción de vídeo WebM/VP8 y codificación de audio AAC. El soporte para la tecnología NFC. Las facilidades en el audio, gráficos y entradas para los desarrolladores de juegos. El soporte nativo para más sensores (como giroscopios y barómetros). Un gestor de descargas para las descargas largas.

### III.2.6 Honeycomb

**ANDROID 3.0 NIVEL DE API 11 (FEBRERO 2011)** Para mejorar la experiencia de Android en las nuevas tabletas se lanza la versión 3.0 optimizada para dispositivos con pantallas grandes. La nueva interfaz de usuario ha sido completamente rediseñada con paradigmas nuevos para la interacción y navegación. Entre las novedades introducidas destacan: Los fragments, con los que podemos diseñar diferentes elementos del interfaz de usuario. La barra de acciones, donde las aplicaciones pueden mostrar un menú siempre visible. Las teclas físicas son reemplazadas por teclas en pantalla; se mejoran las notificaciones, arrastrar y soltar y las operaciones de cortar y pegar.



Figure 7: Honeycomb

La nueva interfaz se pone a disposición de todas las aplicaciones, incluso las construidas para versiones anteriores de la plataforma. Esto se consigue gracias a la introducción de librerías de compatibilidad que pueden ser utilizadas en versiones anteriores a la 3.0.

Se mejoran los gráficos 2D/3D gracias al renderizador OpenGL acelerado por *hardware*. Aparecerá el nuevo motor de gráficos Rederscript, que saca mayor rendimiento al *hardware* e incorpora su propia API. Se incorpora un nuevo motor de animaciones mucho más flexible, conocido como animación de propiedades.

Primera versión de la plataforma que soporta procesadores multinúcleo. La máquina virtual Dalvik ha sido optimizada para permitir multiprocesado, lo que permite una ejecución más rápida de las aplicaciones, incluso aquellas que son de hilo único.

Se incorporan varias mejoras multimedia, como listas de reproducción M3U a través de HTTP Live Streaming, soporte a la protección de derechos musicales (DRM) y soporte para la transferencia de archivos multimedia a través de USB con los protocolos MTP y PTP.

En esta versión se añaden nuevas alternativas de conectividad, como las nuevas APIS de Bluetooth A2DP y HSP con *streaming* de audio. También, se permite conectar teclados completos por USB o Bluetooth.

Se mejora el uso de los dispositivos en un entorno empresarial. Entre las novedades introducidas destacamos las nuevas políticas administrativas con encriptación del almacenamiento, caducidad de contraseña y mejoras para administrar los dispositivos de empresa de forma eficaz.

A pesar de la nueva interfaz gráfica optimizada para tabletas, Android 3.0 es compatible con las aplicaciones creadas para versiones anteriores.

**ANDROID 3.1 NIVEL DE API 12 (MAYO 2011)** Se permite manejar dispositivos conectados por USB (tanto host como dispositivo). Protocolo de transferencia de fotos y vídeo (PTP/MTP) y de tiempo real (RTP).

**ANDROID 3.2 NIVEL DE API 13 (JULIO 2011)** Optimizaciones para distintos tipos de tableta. Zoom compatible para aplicaciones de tamaño fijo. Sincronización multimedia desde SD.

### III.2.7 Ice Cream Sandwich

**ANDROID 4.0 NIVEL DE API 14 (OCTUBRE 2011)** La característica más importante es que se unifican las dos versiones anteriores (2.x para teléfonos y 3.x para tabletas) en una sola compatible con cualquier tipo de dispositivo. A continuación destacamos algunas de las características más interesantes.



Figure 8: Ice Cream Sandwich

Se introduce una nueva interfaz de usuario totalmente renovada; por ejemplo, se reemplazan los botones físicos por botones en pantalla (como ocurría en las versiones 3.x). Nueva API de reconocimiento facial que, entre otras muchas aplicaciones, permite al propietario desbloquear el teléfono. También se mejora en el reconocimiento de voz; por ejemplo, se puede empezar a hablar sin esperar la conexión con el servidor.

Aparece un nuevo gestor de tráfico de datos por Internet, donde podremos ver el consumo de forma gráfica y donde podemos definir los límites de ese consumo para evitar cargos inesperados con la operadora. Incorpora herramientas para la edición de imágenes en tiempo real, para distorsionar, manipular e interactuar con la imagen en el momento de ser capturada. Se mejora la API para comunicaciones por NFC y la integración con redes sociales.

En diciembre de 2011 aparece una actualización de mantenimiento (versión 4.0.2) que no aumenta el nivel de API.

**ANDROID 4.0.3 NIVEL DE API 15 (DICIEMBRE 2011)** Se introducen ligeras mejoras en algunas APIs incluyendo el de redes sociales, calendario, revisor ortográfico, texto a voz y bases de datos entre otros. En marzo de 2012 aparece la actualización 4.0.4.

### III.2.8 Jelly Bean

**ANDROID 4.1 NIVEL DE API 16 (JULIO 2012)** En esta versión se hace hincapié en mejorar un punto débil de Android: la fluidez de la interfaz de usuario. Con este propósito se incorporan varias técnicas: sincronismo vertical, triple búfer y aumento de la velocidad del procesador al tocar la pantalla.

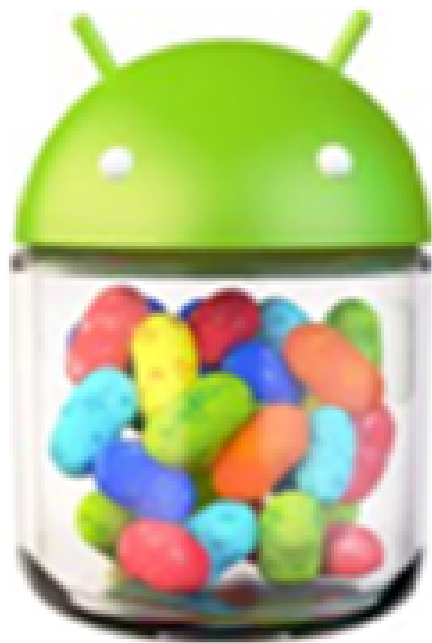


Figure 9: Jelly Bean

Se mejoran las notificaciones con un sistema de información expandible personalizada. Los widgets de escritorio pueden ajustar su tamaño y hacerse sitio de forma automática al situarlos en el escritorio. El dictado por voz puede realizarse sin conexión a Internet (de momento, solo en inglés).

Se introducen varias mejoras en Google Search. Se potencia la búsqueda por voz con resultados en forma de ficha. La función Google Now permite utilizar información de posición, agenda y hora en las búsquedas.

Se incorporan nuevos soportes para usuarios internacionales, como texto bidireccional y teclados instalables. Para mejorar la seguridad, las aplicaciones son cifradas. También se permiten actualizaciones parciales de aplicaciones.

**ANDROID 4.2 NIVEL DE API 17 (NOVIEMBRE 2012)** Una de las novedades más importantes es que podemos crear varias cuentas de usuario en el mismo dispositivo. Aunque esta característica solo está disponible en tabletas. Cada cuenta tendrá sus propias aplicaciones y su propia configuración.

Los Widgets de escritorio pueden aparecer en la pantalla de bloqueo. Se incorpora un nuevo teclado predictivo deslizante al estilo Swype. Posibilidad de conectar dispositivo y TVHD mediante wifi (Miracast). Mejoras menores en las notificaciones. Nueva aplicación de cámara que incorpora la funcionalidad Photo Sphere para hacer fotos panorámicas inmersivas (en 360°).

**ANDROID 4.3 NIVEL DE API 18 (JULIO 2013)** Esta versión introduce mejoras en múltiples áreas. Entre ellas los *perfiles restringidos* (disponible sólo en tabletas) que permiten controlar los derechos de los usuarios para ejecutar aplicaciones específicas y para tener acceso a datos específicos. Igualmente, los programadores pueden definir restricciones en las *apps*, que los propietarios puedan activar si quieren. Se da soporte para Bluetooth Low Energy (BLE) que permite a los dispositivos Android comunicarse con los periféricos con bajo consumo de energía. Se agregan nuevas características para la codificación, transmisión y multiplexación de datos multimedia. Se da soporte para OpenGL ES 3.0. Se mejora la seguridad para gestionar y ocultar las claves privadas y credenciales.

### III.2.9 KitKat

**ANDROID 4.4 NIVEL DE API 19 (OCTUBRE 2013)** Aunque se esperaba la versión número 5.0 y con el nombre Key Lime Pie, Google sorprendió con el cambio de nombre, que se debió a un acuerdo con Nestlé para asociar ambas marcas.

El principal objetivo de la versión 4.4 es hacer que Android esté disponible en una gama aún más amplia de dispositivos, incluyendo aquellos con tamaños de memoria RAM de solo 512 MB. Para ello, todos los componentes principales de Android han sido recortados para reducir sus requerimientos de memoria, y se ha creado una nueva API que permite adaptar el comportamiento de la aplicación en dispositivos con poca memoria.



Figure 10: KitKat

Más visibles son algunas nuevas características de la interfaz de usuario. El modo de inmersión en pantalla completa oculta todas las interfaces del sistema (barras de navegación y de estado) de tal manera que una aplicación puede aprovechar el tamaño de la pantalla completa. *WebViews* (componentes de la interfaz de usuario para mostrar las páginas Web) se basa ahora en el software de Chrome de Google y por lo tanto puede mostrar contenido basado en HTML5.

Se mejora la conectividad con soporte de NFC para emular tarjetas de pago tipo HCE, varios protocolos sobre Bluetooth y soporte para mandos infrarrojos. También se mejoran los sensores para disminuir su consumo y se incorpora un sensor contador de pasos.

Se facilita el acceso de las aplicaciones a la nube con un nuevo marco de almacenamiento. Este marco incorpora un tipo específico de *content provider* conocido como *document provider*, nuevas intenciones para abrir y crear documentos y una ventana de diálogo que permite al usuario seleccionar ficheros. Se incorpora un administrador de impresión para enviar documentos a través de Wi-Fi a una impresora. También se añade un *content provider* para gestionar los SMS.

Desde una perspectiva técnica, hay que destacar la introducción de la nueva máquina virtual ART, que consigue tiempos de ejecución muy superiores a la máquina Dalvik. Sin



embargo, todavía está en una etapa experimental. Por defecto se utiliza la máquina virtual Dalvik, y se permite a los programadores activar opcionalmente ART para verificar que sus aplicaciones funcionan correctamente.

### III.2.10 Lollipop

**ANDROID 5.0 NIVEL DE API 21 (NOVIEMBRE 2014)** La novedad más importante de Lollipop es la extensión de Android a nuevas plataformas, incluyendo Android Wear, Android TV y Android Auto. Hay un cambio significativo en la arquitectura, al utilizar la máquina virtual ART en lugar de Dalvik. Esta novedad ya había sido incorporada en la versión anterior a modo de prueba. ART mejora de forma considerable el tiempo de ejecución del código escrito en Java. Además se soporta dispositivos de 64 bits en procesadores ARM, x86, y MIPS. Muchas aplicaciones del sistema (Chrome, Gmail,...) se han incorporado en código nativo para una ejecución más rápida.



Figure 11: Lollipop

Desde el punto de vista del consumo de batería, hay que resaltar que en Lollipop el modo de ahorro de batería se activa por defecto. Este modo desconecta algunos

componentes en caso de que la batería esté baja. Se incorpora una nueva API (*android.app.job.JobScheduler*) que nos permite que ciertos trabajos se realicen solo cuando se cumplan determinadas condiciones (por ejemplo con el dispositivo cargando). También se incluyen completas estadísticas para analizar el consumo que nuestras aplicaciones hacen de la batería.

En el campo Gráfico Android Lollipop incorpora soporte nativo para OpenGL ES 3.1. Además esta versión permite añadir a nuestras aplicaciones un paquete de extensión con funcionalidades gráficas avanzadas (fragment shader, tessellation, geometry shaders, ASTC,...).

Otro aspecto innovador de la nueva versión lo encontramos en el diseño de la interfaz de usuario. Se han cambiado los iconos, incluyendo los de la parte inferior (Retroceder, Inicio y Aplicaciones), que ahora son un triángulo, un círculo y un cuadrado. El nuevo enfoque se centra en Material Design (<http://www.google.com/design/material-design.pdf>). Consiste en una guía completa para el diseño visual, el movimiento y las interacciones a través de plataformas y dispositivos. Google pretende aplicar esta iniciativa a todas las plataformas, incluyendo wearables y Google TV. La nueva versión también incluye varias mejoras para controlar las notificaciones. Ahora son más parecidas a las tarjetas de Google Now y pueden verse en la pantalla de bloqueo.



Se incorporan nuevos sensores como el de pulso cardiaco, el de inclinación (para reconocer el tipo de actividad del usuario), y sensores de interacción compuestos para detectar ciertos gestos.

Como curiosidad la nueva versión introduce un modo de bloqueo que impide al usuario salir de una aplicación y bloquea las notificaciones. Esto podría utilizarse, por ejemplo, para que mientras un usuario realiza un examen, no pueda ver las notificaciones, acceder a otras aplicaciones, o volver a la pantalla de inicio.

**ANDROID 5.1 NIVEL DE API 22 (MARZO 2015)** Se añaden algunas mejoras a nivel de usuario en los ajustes rápidos. A nivel de API se añade soporte para varias tarjetas SIM en un mismo teléfono; la clase *AndroidHttpClient* se marca como obsoleta; y

se añade un API para que las empresas proveedoras de servicios de telecomunicación puedan distribuir software de forma segura a través de Google Play. La característica más interesante es que para poder acceder a esta API la aplicación ha de estar firmada con un certificado que coincida con el que el usuario tiene en su tarjeta UICC.

### III.2.11 Marshmallow

**ANDROID 6.0 NIVEL DE API 23 (OCTUBRE 2015)** Una de las novedades más interesantes es el administrador de permisos. Los usuarios podrán conceder o retirar ciertos permisos a cada aplicación. Con esto el sistema da mucha más protección a la privacidad de los usuarios.



Figure 12: Marshmallow

Ahora, el sistema realiza una copia de seguridad automática de todos los datos de las aplicaciones. Esto resulta muy útil al cambiar de dispositivo o tras restaurar valores de fábrica. Para disponer de esta funcionalidad simplemente usa el targer Android 6.o. No es necesario agregar código adicional.

Android 6.o integra el asistente por voz Now on Tap. Es una evolución de Google Now más integrada con las aplicaciones. Se activa con pulsación larga de home. Aparecerán

tarjetas sobre la aplicación actual y lo que muestra. La aplicación actual podrá aportar información al asistente. En esta misma línea, se añade un API que permite interacciones basadas en voz. Es decir, si nuestra aplicación ha sido lanzada por voz, podremos solicitar una confirmación de voz del usuario, seleccionar de una lista de opciones o cualquier información que necesite.

Se introducen los enlaces de aplicación con los que podremos asociar la aplicación que abre una URL en función de su dominio web. Aunque muchos dispositivos ya lo permitían, en esta actualización se añade autenticación por huella digital a la API. Tu aplicación puede autenticar al usuario usando las credenciales para desbloquear su dispositivo (pin, patrón o contraseña). Esto libera al usuario de tener que recordar contraseñas específicas de la aplicación. Y te evita tener que implementar tu propia interfaz de autenticación.

Compartir con otros usuarios ahora es más fácil con Direct Share. Permite no solo escoger la aplicación con la que compartes, sino también el usuario. Si tu aplicación es un posible destino para compartir vas a poder indicar al sistema la lista de usuarios que pueden recibir información.

En Android 6.0 podemos utilizar parte de un dispositivo de almacenamiento externo, para que sea usado como almacenamiento interno. Podemos fragmentar, formatear y encriptar una tarjeta SD para ser usada como memoria interna. También podemos montar y extraer lápices de memoria USB de forma nativa.

Se incorpora la plataforma de pagos abierta *Android Pay* que combina NFC y Host Card Emulation. El nuevo gestor de batería, Doze, realiza un uso más eficiente de los recursos, con lo que podemos obtener dos horas extras de autonomía. Se da soporte de forma nativa a pantallas 4 K, lápices Bluetooth, múltiples tarjetas SIM y linterna. Mejoras de posicionamiento utilizando redes WiFi y dispositivos Bluetooth.

### III.2.12 Android Nougat

**ANDROID 7.0 NIVEL DE API 24 (JULIO 2016)** Ahora los usuarios pueden abrir varias aplicaciones al mismo tiempo en la pantalla. Puedes configurar tu aplicación para que

se visualice con unas dimensiones mínimas o inhabilitar la visualización de ventanas múltiples.



Figure 13: Android Nougat

Las notificaciones han sido rediseñadas para un uso más ágil. Hay más opciones para personalizar el estilo de los mensajes (*MessageStyle*). Puedes agrupar notificaciones por temas o programar una respuesta directa.

En la versión anterior se utilizaba una estrategia de compilación *Ahead of Time* (AOT): cuando se descargaba una aplicación, su código era traducido de bytecodes a código nativo, lo que mejoraba los tiempos de ejecución. En la nueva versión se incorpora también la compilación *Just in Time* (JIT), donde no se compila hasta que el código va a ser ejecutado. Android 7.0 propone un planteamiento mixto según el perfil del código. Los métodos directos se compilan previamente (AOT), mientras que otras partes no se compilan hasta que se usan (JIT). Aunque AOT puede introducir retardos en ejecución, ahorra tiempo en la precompilación y en memoria. El mayor impacto de esta técnica se nota en la instalación de las aplicaciones y actualizaciones del sistema. Mientras que en Android 6.0 una instalación podría usar varios minutos, ahora se instala en cuestión de segundos.

Android Nougat incorpora la plataforma de realidad virtual *Daydream*. Se trata de una propuesta de Google que complementa la iniciativa *Cardboard*. Incluye especificaciones software y hardware que nos permitirán diferenciar a los dispositivos compatibles. Los principales fabricantes de móviles se han unido a esta iniciativa.

En la versión anterior, el gestor de batería Doze solo se activaba cuando el dispositivo estaba en reposo. Ahora, se activa poco tiempo después de apagarse la pantalla. Esto permite ahorrar batería cuando llevamos el dispositivo en el bolsillo.

También se ha añadido la nueva API para gráficos 3D, Vulkan, como alternativa a OpenGL. Minimiza la sobrecarga de CPU en el controlador, lo que permite aumentar la velocidad de los juegos.

El usuario va a poder activar el modo de ahorro de datos cuando se encuentre en itinerancia o cuando esté a punto de agotar un paquete de datos. En este caso, tanto el sistema como las aplicaciones han de tratar de minimizar al máximo las transferencias de datos.

**ANDROID 7.1 NIVEL DE API 25 (DICIEMBRE 2016)** La principal novedad son los accesos directos a aplicaciones. Desde el icono de la aplicación, con una pulsación prolongada, aparecen varias opciones que podremos seleccionar. Por ejemplo, podremos iniciar una navegación privada con Chrome de forma directa. Los accesos directos que quieras incorporar a tu aplicación, podrás configurarlo por medio de intents, que han de especificarse en un fichero de configuración.

Se incorporan otras novedades como la posibilidad de insertar imágenes desde el teclado, de la misma forma que ahora insertamos emoticonos.

### III.2.13 Android Oreo

**ANDROID 8.0 NIVEL DE API 26 (AGOSTO 2017)** Destacan las siguientes mejoras en seguridad: Se introduce Google Play Protect, que escanea regularmente las aplicaciones en busca de malware. La opción "Orígenes desconocidos" desaparece. Ahora podemos indicar que aplicaciones pueden instalar apks y cuales no. Desde la opción "Acceso especial de aplicaciones" podemos configurar que aplicaciones pueden realizar ciertas acciones.

El sistema limita más los procesos en segundo plano, para conseguir ahorro en la batería. Se mejora el tiempo de arranque del sistema.



Figure 14: Android Oreo

Pensando en los países emergentes, se lanza Android Go, una distribución adaptada para dispositivos de gama baja (1 GB de RAM o menos). Se preinstalan apps ligeras y en Google Play Store destaca aplicaciones ligeras adecuadas para estos dispositivos. Estas aplicaciones han de cubrir 3 requisitos, trabajar sin red, pesar menos de 10 MB y un buen rendimiento de batería.

Con el fin de reducir la fragmentación de Android, aparece el proyecto Treble que facilitará las actualizaciones a los fabricantes. Se reestructura la arquitectura de Android para definir una interfaz clara entre capa del Núcleo Linux (con sus drivers) de las capas del Framework. Esto permite actualizar Android sin tener que tocar la capa del Núcleo Linux.

Las notificaciones presentan varias mejoras: Podemos añadir color de fondo. Se ordenan por importancia. Las aplicaciones pueden crear canales de notificaciones y el usuario decidir cuales quiere recibir. Podemos posponer una notificación o verlas pulsando sobre el icono de la aplicación.

Los iconos tendrán que estar diseñados en dos capas: el icono y el fondo del icono. Esto permite adaptarse al dispositivo. Además el usuario podrá escoger entre iconos circulares, cuadrados o de esquinas redondeadas.

Ahora podemos reproducir un vídeo en una ventana flotante mientras utilizamos otras aplicaciones. Al seleccionar un texto se nos sugieren acciones cuando se trata de un número de teléfono o una dirección. El Autocompletar de Google, que antes estaba disponible en Chrome para guardar contraseñas, ahora se puede usar en cualquier aplicación Android.

### III.2.14 Android Pie

**ANDROID 9.0 NIVEL DE API 28 (AGOSTO 2018)** Una de las novedades más interesantes es el nuevo API WiFi RTT introducido en IEEE 802.11mc. Permite estimar la distancia entre nuestro dispositivo y los puntos de acceso cercanos, lo que permite sistemas de posicionamiento en interiores con una precisión de 1 a 2 metros. Otro importante cambio es la navegación por gestos. Se reemplazan los tres botones en pantalla (triángulo, círculo y cuadrado) por solo 2 (retroceder e inicio). El botón de inicio admite diferentes gestos que nos permite ir al asistente de Google, cambiar entre apps recientes o abrir el menú de apps.



Figure 15: Android Pie

Una interesante innovación es el uso de Inteligencia Artificial, para mejorar diferentes aspectos. La idea consiste en aprender nuestros hábitos a la hora de usar las aplicaciones. Con esta información se puede quitar preferencia sobre el uso de la CPU a las apps menos utilizadas, consiguiendo una reducción de hasta un 30%. Este menor uso de la CPU prolongará la vida de la batería. Usando técnicas similares se pretende aprender



cuando el usuario va a arrancar una aplicación o una acción de esta. De esta forma el sistema puede cargar en memoria la aplicación antes incluso que el usuario decida utilizarla.

Se introducen algunas mejoras que fomentan un uso responsable y saludable del móvil. Por ejemplo, desde el Dashboard podemos consultar el uso que hacemos cada día, en cada aplicación. Podemos establecer alarmas de uso excesivo muy interesantes para el control parental. En esta línea, se introducen nuevos modos de relajación y no molestar para favorecer la desconexión digital.

### III.2.15 Android 10

**ANDROID 10.0 NIVEL DE API 29 (SEPTIEMBRE 2019)** A partir de la versión 10, Google quiere simplificar la marca y los nombres de las versiones. Se abandonan los nombres poste, para utilizar un simple número entero. Para el logo se usa solo la cabeza del robot, e tono de verde algo más claro.



Figure 16: Android 10

Ya no son necesarios los botones para la nevagación a través del sistema operativo. Siguiendo la pauta propuesta en iOS, ahora se utiliza un control por gestos. Por ejemplo, para volver a la actividad anterior deslizaremos desde el extremo derecho a la izquierda.

Se introduce el Focus Mode que activaremos cuando queramos concentrarnos en una determinada tarea o juego y no queremos ser molestados. En este modo podemos configurar qué aplicaciones pueden lanzar una notificación y cuáles no. Implmentación nativa del modo oscuro que permite un significativo ahorro de batería en pantallas OLED.

La función Live Caption permite que el sistema introduzca subtítulos de manera automática cuando se reproduce cualquier contenido de audio o vídeo. Está pensado para reproducir estos contenidos cuando estamos en público y no queremos usar auriculares. Se obtienen de forma local por lo que no es necesario conexión a Internet. No obstante, solo puede activarse si disponemos de un procesador de gran potencia.

Para reducir la fragmentación, las actualizaciones de seguridad se instalan a través de Google Play, sin la intervención del fabricante. Aumentan las restricciones a las aplicaciones de fondo y se mejoran los permisos. Por ejemplo, podemos indicar que una aplicación tenga acceso a nuestra localización solo cuando esté en primer plano.

Se introduce soporte para 5G, WIFI 6, WPA3, teléfonos plegables y presión en pantallas táctiles.