




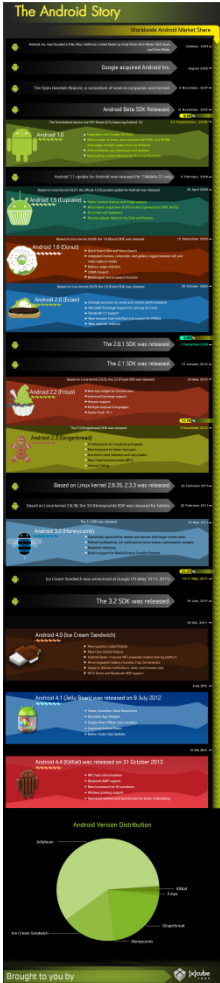
Android

Android



- Android es un sistema operativo diseñado especialmente para dispositivos móviles (lo que no quita que se use para muchas más cosas).
- También es el nombre de la empresa que lo desarrolló (Android Inc.)
- Y el del *simpático* muñeco (también conocido como Andy). 
- El nombre *Android* hace alusión a la novela de *¿Sueñan los androides con ovejas eléctricas?*, que posteriormente fue adaptada al cine como *Blade Runner*.

Historia



Open Handset Alliance



Cualidades de la plataforma Android



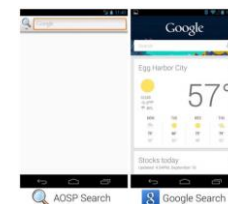
Android, the world's most popular mobile platform

- **Plataforma realmente 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".
- **Portabilidad asegurada.** Las aplicaciones finales son desarrolladas en Java lo que nos asegura que podrán ser ejecutadas en gran variedad de dispositivos, tanto presentes como futuros. 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 móvil de pantalla reducida o en un "phablet".
- **Filosofía de dispositivo siempre conectado a Internet.**
- **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...
- **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...)
- **Optimizado para baja potencia y poca memoria.**
- **Alta calidad de gráficos y sonido.** Gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en 3 dimensiones basados en OpenGL. Incorpora codecs estándar más comunes de audio y vídeo, incluyendo H.264 (AVC), MP3, AAC,...

¿Plataforma abierta?



- Sí pero no! Google está y es un negocio.
- *Las autoridades antimonopolio europeas están examinando las condiciones de uso de Android, ya que consideran que Google podría haber abusado de la posición de Android como sistema operativo líder en smartphones.* Entre otras cosas, **porque impone su motor de búsqueda** (que le da buena parte de sus ingresos por publicidad) en todo smartphone que quiera tener acceso a todas las facilidades que la plataforma Android ofrece.
- El sitio web de los desarrolladores de Android, califica la plataforma Android como **“la plataforma móvil más popular del mundo”**. Esa plataforma está compuesta por el **sistema operativo** como tal, **servicios y soluciones** que conectan los componentes de Android y también una serie de **aplicaciones** móviles de Google.
- Así, tendríamos una base que tiene como pilar el **Android Open Source Project (AOSP)**, a la que se sumarían los **Google Mobile Services** (Google Play, Google Maps, Google Now, por ejemplo) y las **Google Apps** (Gmail, Google Drive, Google Keep, por ejemplo).
- [AOSP](#) es un proyecto en el que colabora la comunidad de usuarios y que ofrece una alternativa a la plataforma como tal, aunque la propia Google es la principal responsable de la gestión y desarrollo del proyecto. Podría considerarse como una **plataforma abierta** en sí misma, ya que además del sistema operativo móvil ofrece algunas aplicaciones nativas que permiten por ejemplo reproducir música, realizar y recibir llamadas o sacar fotos con el móvil o tablet que estamos manejando.
- Las aplicaciones y servicios móviles (*Google Mobile Services* y *Google Apps*) son **totalmente cerrados y propietarios**. Y son precisamente esas aplicaciones y servicios las que dan acceso a unas prestaciones muy demandadas por los usuarios y por los fabricantes.
- Cualquier fabricante **puede meter AOSP en su teléfono móvil** y, en esencia, dispondrá de un dispositivo funcional. Ahora, si quiere acceder a los servicios de Google, tendrá que asumir los términos de Google y sus requisitos (ciertas especificaciones hardware, por ejemplo), además de pasar una validación técnica de Google (y pagar?).
- Las formas en las que Google se asegura (cada vez más) el control de Android y de los fabricantes que quieren utilizarlo no son nuevas, ni son esencialmente “diabólicas” y nadie puede negar que pocas empresas han hecho lo que Google: AOSP es una plataforma móvil real y funcional.



Dispositivos con Android



Teléfonos



Tablets



Wear



TV



Auto

"Android - La imaginación es el límite"

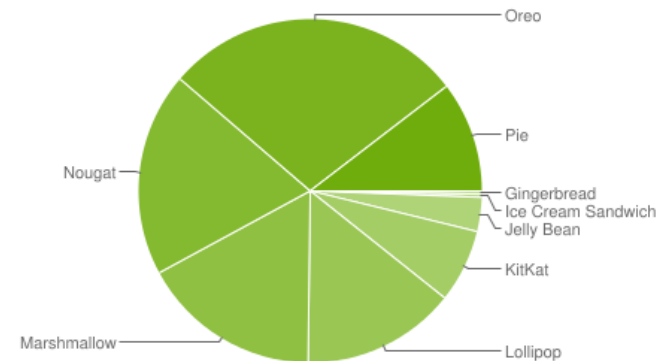


Recon Instruments
Gafas de Esquí

Versiones Android



- Un sistema operativo Android tiene tres tipos de denominar a las versiones de Android, aunque las tres hacen referencia a la misma versión:
 - La **comercial** con el nombre de postre/dulce. Por ejemplo: KitKat
 - La de los **fabricantes** (y también comercial) con la versión y subversión. Por ejemplo: 4.4
 - La de **desarrollador** con el nivel del API (ésta nos interesa mucho para desarrollar en Android): Por ejemplo: 19



Data collected during a 7-day period ending on May 7, 2019.
Any versions with less than 0.1% distribution are not shown.



Android 10 Pie (API 28)



- [Características](#)
- [Enlace a artículo](#)

Android 10

(No hay postre!)



android 

[What is Android](#)

[Android 10](#)

[Phones & Tablets](#)



Android 10 Highlights

Some of the newest ways to control your privacy, customize your phone, and get things done. It's Android, just the way you want it.

Live Caption



Smart Reply



Sound Amplifier



Gesture Navigation



Dark theme



Privacy Controls



Location Controls



Security Updates



Focus mode



Family Link



Arquitectura de dispositivos Android



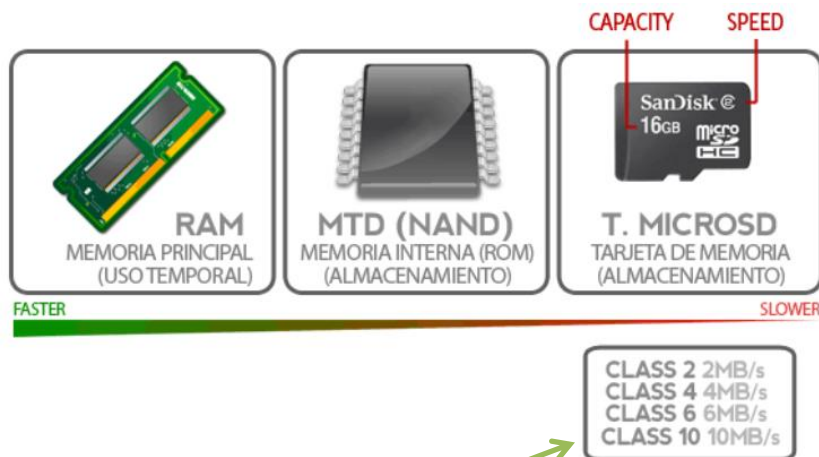
- Siguen la de cualquier otro ordenador :
 - Hardware
 - Firmware
 - Ensamblador
 - Kernel
 - S.O. y apps
- Como desarrolladores y/o usuarios de aplicaciones Android tan solo trabajaremos con la capa más superficial de la arquitectura (la de “S.O. y aplicaciones”).



Características hardware: Memoria



- En Android existen tres tipos de memoria (*de más rápida a más lenta*):



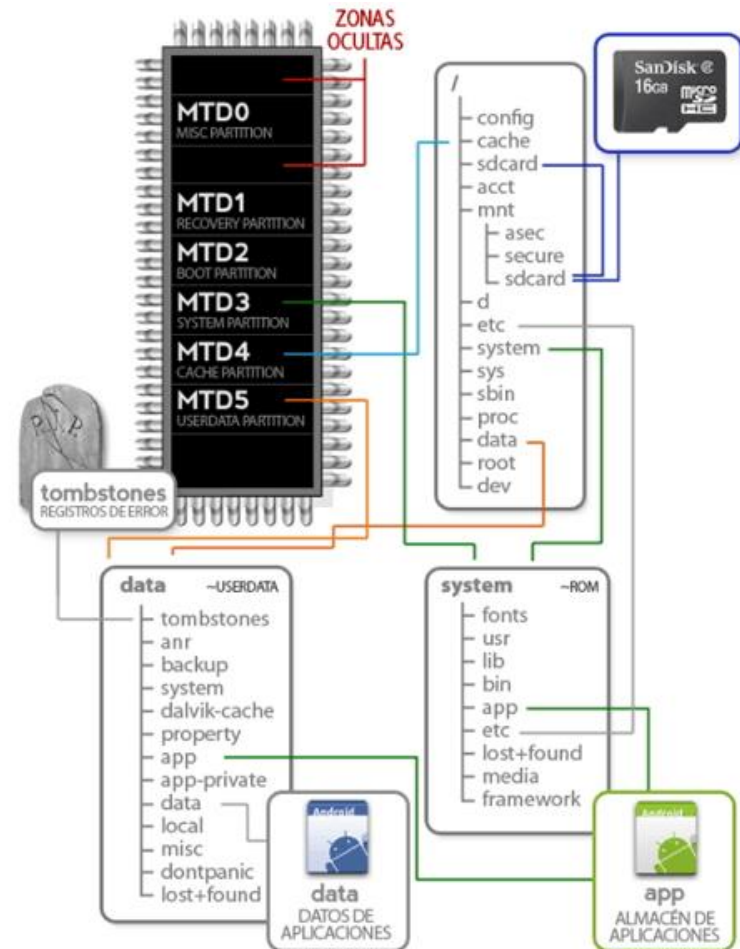
Rendimiento en cuanto a velocidad. Cada clase representa la velocidad de transferencia mínima garantizada

- Memoria RAM:** Es memoria temporal y no se utiliza para almacenamiento, sino para tareas rápidas y aplicaciones en uso. Disponer de muy poca RAM, es algo que repercute de forma importante en el rendimiento del dispositivo, que debe finalizar aplicaciones para cargar otras cuando las necesite.
- Memoria interna:** También llamada **memoria NAND** o **MTD** (o *incorrectamente, ROM*) es el equivalente al disco duro de nuestro dispositivo. Aquí se guardará la versión actual del sistema operativo, el sistema de recuperación de Android, los diferentes logos o imágenes de arranque, las aplicaciones, etc... Un detalle muy importante, es que los fabricantes mencionan la **capacidad total** de la memoria interna del dispositivo (*y no el espacio libre*), y algunos dispositivos no tienen demasiado espacio para aplicaciones.
- Memoria externa:** La denominada **memoria externa** o **memoria SD** (*también incorrectamente, ya que son tarjetas microSDHC*) se utiliza principalmente para almacenamiento de archivos de gran tamaño y de uso poco frecuente (*fotografías, archivos MP3, vídeos, etc...*).

Características Hardware: Memoria interna



- Aunque la estructura de la memoria interna depende de cada fabricante (*la especificada en esta imagen es de una HTC Magic*), la memoria se divide en varias particiones entre las que se encuentran:
 - una partición de arranque,
 - un recovery (*menú de recuperación de errores*),
 - una partición de caché
 - y, sobretodo las dos más importantes:
 - la partición de sistema y
 - la partición de datos de usuario.



Características Hardware: Procesador



- La principal plataforma de hardware para Android es la arquitectura **ARM** que es de tipo RISC (Reduced Instruction Set Computer= Ordenador con Conjunto Reducido de Instrucciones) desarrollada por la compañía ARM Holdings y licenciada a las compañías productoras de procesadores móviles. También es soportado por MIPS e IBM POWER.



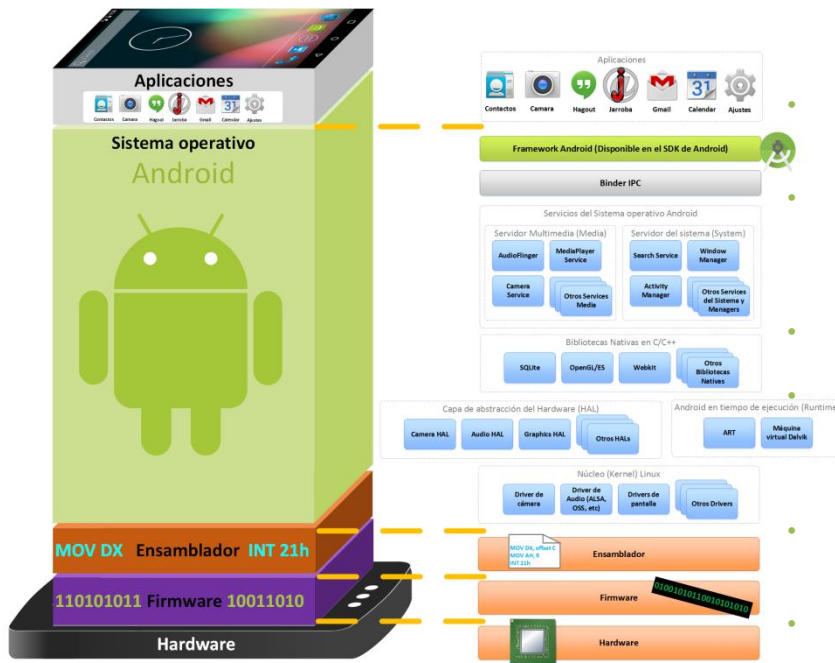
- Este procesador principal es el que determinará la velocidad con la que actúe.
- Además, existe un procesador gráfico auxiliar llamado GPU que se encarga de las operaciones relacionadas con los gráficos (y así libera de cierta carga al procesador principal).

Características



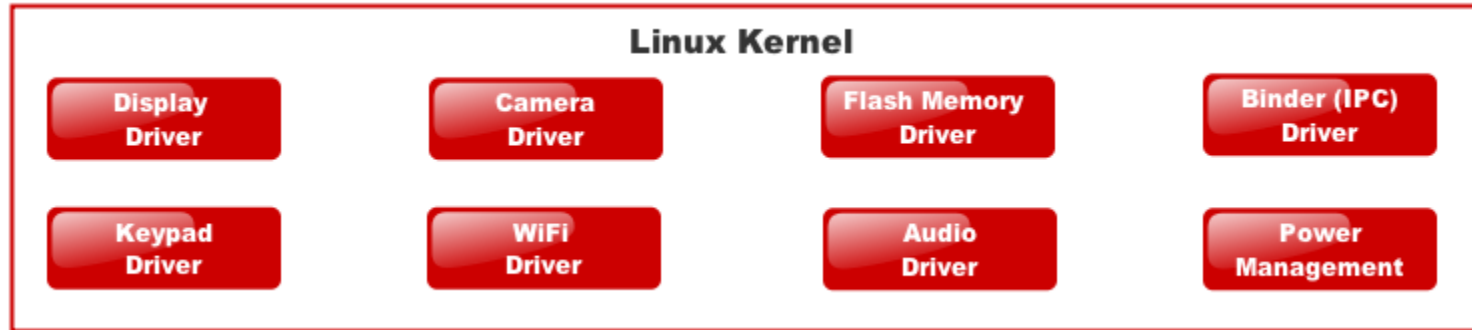
Diseño de dispositivo	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
Almacenamiento	SQLite , una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
Conectividad	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE , IDEN , CDMA , EV-DO , UMTS , Bluetooth , Wi-Fi , LTE , HSDPA , HSPA+ , NFC y WiMAX , GPRS , UMTS y HSDPA+ .
Mensajería	SMS y MMS son formas de mensajería, incluyendo mensajería de texto, además del servicio de Firebase Cloud Messaging (FCM) siendo la nueva versión de Google Cloud Messaging (GCM) bajo la marca Firebase con los nuevos SDK para realizar el desarrollo de mensajería en la nube mucho más sencillo.
Navegador web	El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit , emparejado con el motor JavaScript V8 de Google Chrome . El navegador por defecto de Ice Cream Sandwich obtiene una puntuación de 100/100 en el test Acid3 .
Soporte de Java	Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y se ejecuta en la Máquina Virtual Dalvik . Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. A partir de la versión 5.0, se utiliza el Android Runtime (ART) . El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner . ⁴⁵
Soporte multimedia	Android soporta los siguientes formatos multimedia: WebM , H.263 , H.264 (en 3GP o MP4), MPEG-4 SP , AMR , AMR-WB (en un contenedor 3GP), AAC , HE-AAC (en contenedores MP4 o 3GP), MP3 , MIDI , Ogg Vorbis , WAV , JPEG , PNG , GIF y BMP . ⁴⁴
Soporte para streaming	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player . Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player .
Soporte para hardware adicional	Android soporta cámaras de fotos, de video, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
Entorno de desarrollo	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. Inicialmente el entorno de desarrollo integrado (IDE) utilizado era Eclipse con el plugin de Herramientas de Desarrollo de Android (ADT). Ahora se considera como entorno oficial Android Studio , descargable desde la página oficial de desarrolladores de Android.
Google Play	Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
Multi-táctil	Android tiene soporte nativo para pantallas capacitivas con soporte multitáctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero . La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). ⁴⁶ Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multitáctil de forma nativa. ⁴⁷
Bluetooth	El soporte para A2DP y AVRCP fue agregado en la versión 1.5; ⁴⁸ el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0; ⁴⁹ y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2. Los cambios incluyeron: ⁵⁰
Videollamada	Android soporta videollamada a través de Hangouts (antiguo Google Talk) desde su versión HoneyComb .
Multitarea	Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
Características basadas en voz	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema. ⁵¹
Tethering	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo PdaNet). Para permitir a un PC usar la conexión de datos del móvil Android se podría requerir la instalación de software adicional. ⁵²

Dispositivo Android: explicación



- **Aplicaciones:** cualquier tipo de aplicación Android. Es la capa que utiliza el usuario.
- **Framework de Android:** Acceso al API de Android para reutilizar componentes o modificarlos. Es la capa en la que trabajaremos como desarrolladores Android (trabajaremos con SDK). Todo en esta capa será una representación exacta de la capa HAL.
- **Bibliotecas nativas en C/C++:** Es la capa en la que podremos trabajar como desarrolladores Android a bajo nivel (se trabaja con el NDK y se programa en C/C++).
- **Runtime de Android:** bibliotecas del lenguaje Java que se ejecutan sobre:
 - Una única instancia en la máquina virtual Dalvik (antigua)
 - Sobre el entorno de ejecución Android Runtime (o conocido por sus siglas ART)
- **Comunicación ligada entre procesos (Binder IPC, Inter-Proces Communication):** Permite a las APIs interactuar con los servicios del sistema operativo Android.
- **Servicios del Sistema Android:** Se agrupan en:
 - System: Incluye gestores de ventanas, de notificaciones, etc
 - Media: Incluye aquellos involucrados en la reproducción y la grabación multimedia (imágenes, audio, video, etc)
- **Capa de abstracción del Hardware (HAL, Hardware Abstraction Layer):** Interfaz que permite al sistema operativo Android llamar a la capa de drivers (controladores) del dispositivo.
- **Kernel (núcleo) Linux:** Capa de abstracción del hardware y servicios de seguridad, gestión de memoria, de procesos, pila de red, modelo de los controladores, etc.
- **Ensamblador:** lenguaje de bajo nivel para circuitos integrados programables.
- **Firmware:** Instrucciones máquina grabadas en un chip para propósitos específicos.
- **Hardware:** Son las partes físicas y los componentes.

Linux Kernel



- La base del sistema operativo Android es un Linux Kernel versión 3.4 (no confundir Linux Kernel con Linux) solo que adaptado a las características del hardware en el que se ejecutará Android
- Permite que una aplicación pueda acceder a los componentes hardware sin necesidad de conocer el modelo o características precisas de los que están instalados en cada teléfono. Para cada elemento hardware del teléfono existe un controlador (o *driver*) dentro del kernel que permite utilizarlo desde el software.
- Además se encarga de gestionar los diferentes recursos del teléfono (energía, memoria, ...) y del sistema operativo en sí: procesos, elementos de red, etc.



Librerías nativas



- La capa que se sitúa justo sobre el kernel la componen las bibliotecas nativas de Android.
- Estas bibliotecas están escritas en C o C++ y compiladas para la arquitectura hardware específica del teléfono, tarea que normalmente realiza el fabricante, que también se encarga de instalarlas en el terminal antes de ponerlo a la venta.
- Su cometido es proporcionar funcionalidad a las aplicaciones y se exponen a los desarrolladores a través del framework.
- Entre las librerías más importantes ubicadas aquí, se pueden encontrar las siguientes:





Funciones de las librerías

Es la encargada de componer los diferentes elementos de navegación de pantalla. Gestiona también las ventanas pertenecientes a las distintas aplicaciones activas en cada momento.

Proporciona todos los códecs necesarios para el contenido multimedia (vídeo, audio, imágenes estáticas y animadas, etc.)

Librerías gráficas:
OpenGL/SL maneja gráficos en 3D y SGL en 2D



Motor de bases de datos relacionales

Proporciona un motor para las aplicaciones de tipo navegador

Posibilita la utilización de dicho protocolo para establecer comunicaciones seguras.

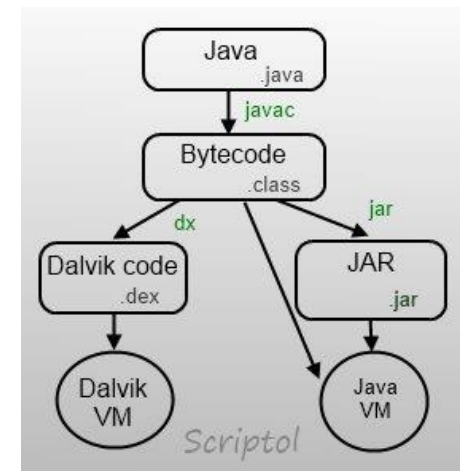
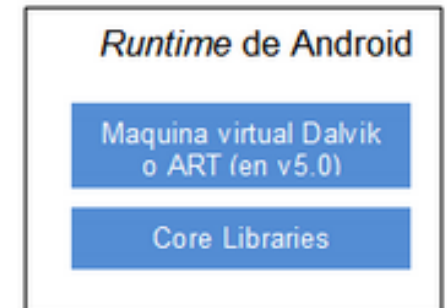
Permite mostrar fuentes tipográficas, tanto basadas en mapas de bits como vectoriales

Incluye todas las cabeceras y funciones según el estándar del lenguaje C.

Entorno de ejecución



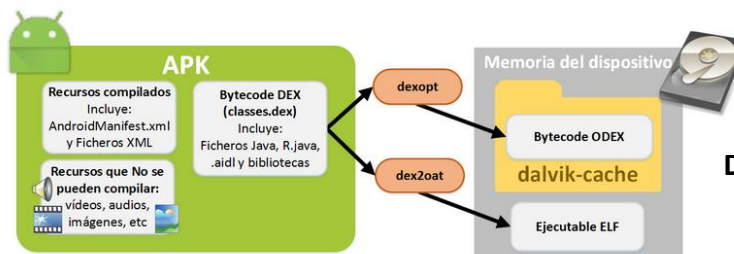
- Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de correr Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual **Dalvik**, que respondiera mejor a estas limitaciones. 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%.
- El módulo **Core Libraries** proporciona muchas de las funciones disponibles en la librería base de Java así como funciones específicas de Android.



Dalvik vs. Art



- Como sabemos, las apps de Android se distribuyen en el formato .apk. Un apk contiene clases de Java compiladas en un formato de bytecode conocido como DEX. El formato DEX es independiente de arquitectura de procesador, por lo que para ser ejecutado necesita traducirse a “código máquina” nativo para el procesador de nuestro dispositivo. La diferencia fundamental entre Dalvik y ART es cuándo hacen esta traducción o compilación.
- **Dalvik** utiliza lo que se llama compilación “justo a tiempo” (just-in-time, o **JIT**). Cada que iniciamos una app, la máquina virtual traduce una parte del bytecode DEX a código máquina utilizando la herramienta dexopt. Conforme continúa la ejecución de la app, se compila más bytecode y se almacena en memoria cache. Es por ello que se dice que la app está siendo compilada “justo a tiempo” conforme la usamos.
- **ART** utiliza compilación previa (ahead-of-time, o **AOT**). Las apps se compilan desde que se instalan en el dispositivo utilizando la herramienta dex2oat y ya se deja instalado el código máquina listo para ejecutarse y sin necesidad de mayor compilación.
- En general, las aplicaciones tienen mejor desempeño con ART, dado que el procesador no tiene que dedicar recursos a estar compilando la app mientras ésta se usa lo que además lo que contribuye a un menor consumo de energía. Por otro lado, las apps necesitan mayor tiempo para instalarse y una vez instaladas ocuparán más espacio de almacenamiento que antes (aproximadamente un 20% más).

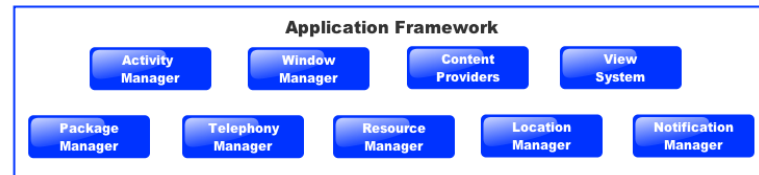


DEX: Dalvik Executable, Ejecutable para máquinas virtuales Dalvik

ODEX: Optimized DEX, DEX Optimizado

ELF: Executable and Linkable Format

Entorno de aplicación



- Representa fundamentalmente el conjunto de herramientas de desarrollo de cualquier aplicación. Toda aplicación que se desarrolle para Android, ya sean las propias del dispositivo, las desarrolladas por Google o terceras compañías, o incluso las que el propio usuario cree, utilizan el mismo conjunto de API y el mismo "framework", representado por este nivel.
- Entre las API's más importantes ubicadas aquí, se pueden encontrar las siguientes:
 - *Activity Manager*: Para gestionar el ciclo de vida de las aplicaciones en Android.
 - *Window Manager*: Gestiona las ventanas de las aplicaciones.
 - *Telephone Manager*: Incluye todas las API vinculadas a las funcionalidades propias del teléfono.
 - *Resource Manager*: proporciona acceso a recursos que no son código Java.
 - *Notification Manager*: permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
 - *Content Provider*: Permite a cualquier aplicación compartir sus datos con las demás aplicaciones.
 - *View System*: Proporciona un gran número de elementos para poder construir interfaces de usuario.
 - *Location Manager*: Posibilita a las aplicaciones la obtención de información de localización y posicionamiento.
 - ...

Aplicaciones



- La capa superior incluye todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, tanto las nativas (programadas en C o C++) como las administradas (programadas en Java), tanto las que vienen de serie con el dispositivo como las instaladas por el usuario.



- Aquí está también la aplicación principal del sistema: **Inicio** (*Home*), también llamada a veces lanzador (*launcher*), porque es la que permite ejecutar otras aplicaciones proporcionando la lista de aplicaciones instaladas y mostrando diferentes escritorios donde se pueden colocar accesos directos a aplicaciones o incluso pequeñas aplicaciones incrustadas o *widgets*, que son también aplicaciones de esta capa.

Lenguajes de desarrollo



- Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android SDK (Kit de desarrollo de Android que proporciona las bibliotecas API y herramientas de desarrollo necesarias para crear, probar y depurar aplicaciones para Android).
- Además se trabaja con ficheros XML para escribir los diseños visuales.
- También se pueden desarrollar parte de las aplicaciones usando C o C++ y así reusar código ya existente mediante el Android Native Development Kit (NDK), aunque Android no lo recomienda.



Kotlin



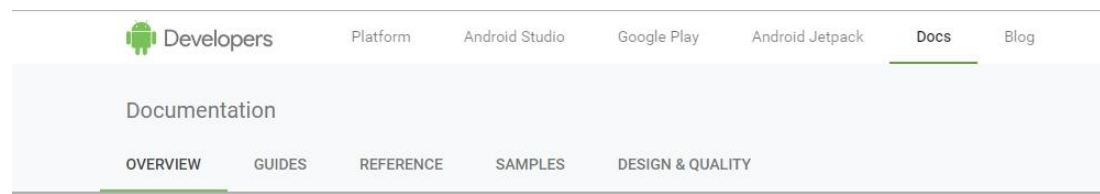
- **Agosto 2017: Kotlin entra oficialmente como lenguaje de Android, junto con Java y C++.**
- <https://kotlinlang.org/>
- <http://kotlin.es/sobre-kotlin/>
- <https://developer.android.com/kotlin/index.html>
- <https://kotlinlang.org/docs/tutorials/kotlin-android.html>



Plataforma de desarrollo



- La plataforma de desarrollo funciona en Windows, en Linux y también en Mac, lo que posibilita a los desarrolladores usar el sistema operativo de su preferencia.
- Cuenta con una magnífica [documentación en línea](#). En ella, tendremos toda la información acerca de las recomendaciones de Google para diseñar nuestra app, toda la información sobre la API de Android e información para saber publicar nuestra aplicación, sabiendo cómo promocionarla, anunciarla...

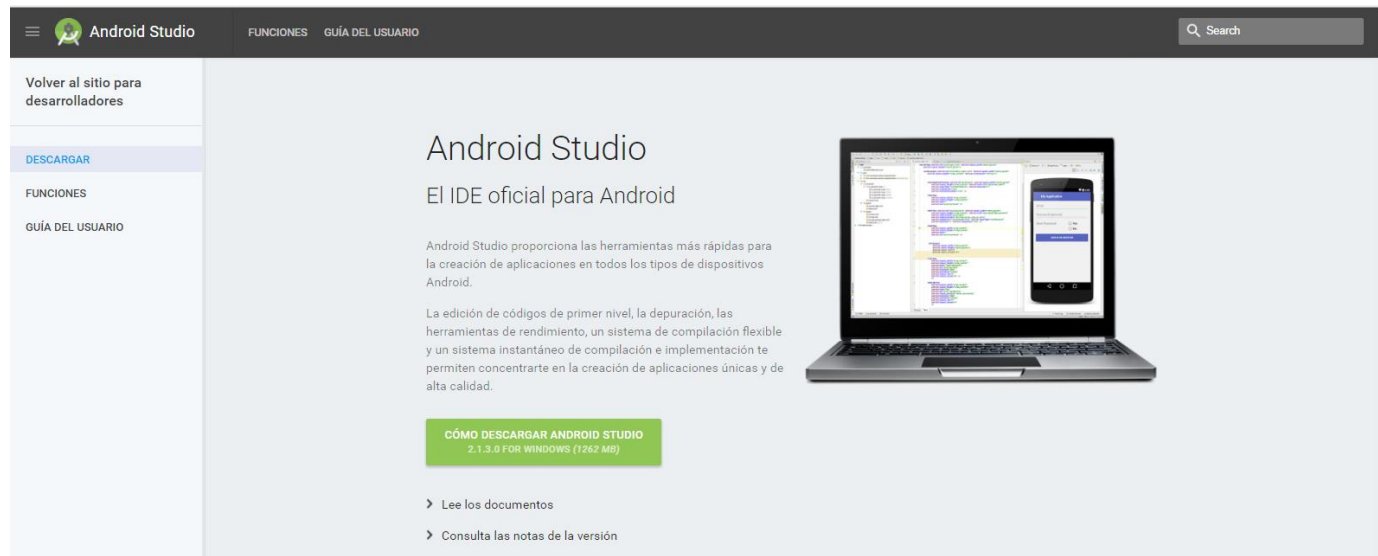


- Además hay información adicional, sobre Android, conseguir el SDK (Software Development Kit), soporte, blog, cursos...

Herramientas de desarrollo



- SDK: Incluye (entre otros) un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software.
- El entorno de desarrollo es Android Studio basado en IntelliJ IDEA (en su descarga incluye el SDK).



Otras herramientas de desarrollo



- [MIT App Inventor](#) (un entorno visual para programadores novatos)
- [gplad](#) (proyecto de lenguaje de programación gráfica para dispositivos Android)
- [Andromo](#)
- <http://www.appsgeyser.com/>
- <http://appyet.com/>