



# Exposición:

## Android Custom ROM's y SO's

Arellanes Conde Esteban

**Fecha límite de entrega:** 22/10/2024 23:59:59 hrs  
(22 de octubre de 2024 a las 23:59)

### 1. Introducción:

#### 1.1. Evolución de los Sistemas Operativos Móviles:

En 1996, cuando los dispositivos móviles aún no eran comunes, *Palm* marcó un hito al lanzar *Palm OS 1.0*, el primer sistema operativo diseñado específicamente para estos terminales. Este sistema incluía aplicaciones innovadoras desarrolladas por *RIM*, como correo electrónico, agenda, memopad y gestión de tareas, ofreciendo una solución integral para la productividad móvil. *Palm OS* sentó las bases para lo que posteriormente sería la evolución de los sistemas operativos móviles, siendo uno de los primeros en integrar herramientas de uso diario en un dispositivo portátil.

### 2. Nacimiento de Android:

El nacimiento de Android se remonta a 2003, cuando **Andy Rubin**, **Rich Miner**, **Nick Sears** y **Chris White** fundaron **Android Inc.** con la intención inicial de desarrollar un sistema operativo para cámaras digitales. Sin embargo, pronto se dieron cuenta de que ese mercado no tenía

un gran potencial, por lo que decidieron enfocar sus esfuerzos en la creciente industria de los teléfonos móviles, buscando competir con plataformas como **Symbian** y **Windows Mobile**. Su objetivo era crear dispositivos que respondieran mejor a las necesidades del usuario.

En 2005, Google adquirió Android Inc., reconociendo el valor de la tecnología en desarrollo. Inicialmente, Android se concibió como una plataforma similar a **Blackberry**, basada en el teclado físico **QWERTY**. No obstante, la llegada del **iPhone** en 2007 revolucionó el mercado, lo que obligó a Android a adaptarse rápidamente a las nuevas demandas de la industria. El resultado fue el lanzamiento de **Android 1.0** en septiembre de 2008, junto con el **HTC Dream** (o T-Mobile G1), el primer teléfono en utilizar este sistema operativo. Esta primera versión ya incluía muchas de las características esenciales que sentarían las bases del éxito de Android, permitiendo su evolución hacia el sistema operativo dominante que conocemos hoy. Esta primera versión incluía muchos bloques fundamentales que han llegado hasta los dispositivos móviles que utilizamos hoy en día.

Sin embargo, en 2017, **Google**, por presiones del gobierno de EEUU rompió relaciones con Huawei. Un 19 de mayo, la empresa del buscador suspendió con **Huawei** aquellos negocios que requirieran transferencia de hardware y software, salvo aquellos cubiertos por licencias open source.

El éxito de **Android** radica en su código abierto, lo que permitió a fabricantes de todo el mundo adoptar y personalizar el sistema operativo según sus necesidades. Esta apertura, junto con la integración de los servicios de Google, permitió que Android se convirtiera en una plataforma dominante, impulsando la expansión de la tecnología móvil a nivel global.

## 2.1. ¿Por qué Android?

El nombre “**Android**” proviene del término griego “**andr-**,” que significa hombre, y “**eidōs**,” que se traduce como forma o apariencia, refiriéndose a un ser humanoide. Esta denominación refleja la intención de crear un sistema operativo que pudiera interactuar de manera intuitiva con los usuarios. También cabe destacar la inspiración sobre la mitología judía sobre origen de los Golem, en el que

un pedazo de arcilla podía cobrar vida mediante un ritual de cábala con la palabra *emet* (verdad en hebreo) y para detener sus funciones había que eliminar la letra *aleph* de la palabra *emet* que el gólem llevaba escrita en su frente, con lo que ésta pasó a significar “*muerte*”, en hebreo *met*.

Pero también la inspiración se extiende a la obra de Philip K. Dick, especialmente su novela “¿Sueñan los androides con ovejas eléctricas?”, que explora la complejidad de la inteligencia artificial y la naturaleza de la humanidad, temas centrales en el desarrollo de la tecnología y la robótica hasta tal punto de un androide ser indistinguible de un ser orgánico. Así, Android no solo alude a un sistema operativo, sino también a una visión más profunda sobre la relación entre humanos y máquinas.

## 2.2. ¿Es android un “*Linux*”?

La pregunta de si Android es simplemente **Linux** no tiene una respuesta clara, ya que el debate ha persistido durante años. Android utiliza el núcleo (o kernel) de Linux, lo que nos sugiere que es una variante de **Linux**, pero su funcionamiento completo incluye muchas modificaciones y elementos que no se encuentran en las distribuciones estándar de **GNU/Linux**. La confusión radica en que el término “**Linux**” se usa para referirse a diferentes conceptos, como el núcleo o las distribuciones completas que incluyen software adicional. Android, a través del **Proyecto de Código Abierto de Android** (AOSP), es de código abierto, pero las versiones comerciales incluyen mucho software propietario, lo que lo hace sentir más cerrado en comparación con las distribuciones típicas de Linux. En resumen, aunque Android se basa en Linux, sus diferencias y personalizaciones lo distinguen como un sistema operativo único y especializado, mayoritariamente para sistemas operativos embebidos.

## 2.3. Open Source vs FOSS

El término “**Open Source**” (código abierto) se refiere a un enfoque de desarrollo de software que permite a los usuarios acceder, modificar y distribuir el código fuente de un programa. Por otro lado, “**FOSS**” (Free and Open Source Software) combina dos conceptos: **la libertad del software**

(en términos de uso, modificación y distribución) y el acceso al código fuente. **Mientras que todo software FOSS es Open Source, no todo Open Source es necesariamente considerado "libre"** en el sentido ético, ya que puede haber restricciones en su uso. En resumen, **FOSS** enfatiza tanto la libertad como el acceso al código, promoviendo una filosofía de colaboración y comunidad.

## 2.4. Android Custom ROM's

Las **Custom ROMs** son versiones modificadas de Android que permiten a los usuarios personalizar su experiencia más allá de las limitaciones de las versiones oficiales. Desarrolladas por comunidades de entusiastas, estas ROMs ofrecen características adicionales, mejoras de rendimiento y actualizaciones de seguridad que pueden no estar disponibles en los dispositivos de fábrica.

Una de las principales ventajas de las **Custom ROMs** es la personalización. Los usuarios pueden modificar tanto la apariencia como la funcionalidad del sistema, adaptándolo a sus preferencias. Además, permiten actualizar dispositivos que ya no reciben soporte oficial, prolongando su vida útil y brindando acceso a las últimas funciones de Android.

Algunas de las **Custom ROMs** más populares incluyen **GrapheneOS**, **LineageOS**, **CalyxOS**, **DivestOS**, **Paranoid Android**, entre otras, como **Resurrection Remix** y **AOSP Extended**, que ofrecen amplias opciones de personalización. Las primeras conocidas por su estabilidad, y enfoque hacia la privacidad.

Sin embargo, instalar una **Custom ROM** conlleva riesgos. Puede anular la garantía del dispositivo y, en algunos casos, resultar en inestabilidad. Además, el proceso de instalación puede ser complicado y requiere ciertos conocimientos técnicos, lo que puede llevar a problemas como el "*bricking*" del dispositivo.

En resumen, las **Custom ROMs** son una opción atractiva para quienes desean personalizar su experiencia Android y acceder a actualizaciones, pero es importante estar consciente de los riesgos y tener la preparación adecuada antes de instalarlas.

## 2.5. Rooteo

El Rooteo (Rooting o enraizamiento) en Android no se atribuye a un solo inventor o a un individuo específico. En cambio, surgió como una práctica impulsada por la comunidad entre los usuarios y desarrolladores de Android. El rooteo permite a los usuarios obtener acceso administrativo (root) al sistema operativo, lo que permite modificar archivos del sistema, instalar aplicaciones no aprobadas y personalizar sus dispositivos más allá de las limitaciones del fabricante.

El proceso se hizo popular poco después del lanzamiento de Android en 2008, y varios desarrolladores y entusiastas contribuyeron con herramientas y métodos para rootear. Figuras notables en los primeros días del rooteo de Android incluyen a los desarrolladores detrás de herramientas como Superuser, que administra los permisos de root, y otras herramientas como **ClockworkMod Recovery**, que facilitó el proceso de rooteo.

En general, el rooteo es el resultado de esfuerzos de colaboración dentro de la comunidad de Android y no la invención de una sola persona.

## 3. Arquitectura de Android (The AOSP Architecture):

### 3.1. Arquitectura ARM

La arquitectura **ARM** (Advanced RISC Machine) es un conjunto de arquitecturas de procesadores basadas en el diseño RISC (Reduced Instruction Set Computing). Desarrollada inicialmente en los años 80 por Acorn Computers, ARM se ha convertido en uno de los diseños de arquitectura más utilizados en el mundo, especialmente en dispositivos móviles y sistemas embebidos.

### Características Principales

1. **Eficiencia Energética:** ARM está diseñada para ofrecer un alto rendimiento con un bajo consumo de energía, lo que la hace ideal para dispositivos portátiles como teléfonos inteligentes

y tabletas.

2. **Escalabilidad:** La arquitectura ARM se puede escalar desde pequeños microcontroladores hasta potentes procesadores utilizados en servidores.
3. **Conjunto de Instrucciones RISC:** Utiliza un conjunto de instrucciones reducido, lo que permite un diseño más simple y una ejecución más rápida.
4. **Licenciamiento:** ARM no fabrica sus propios chips; en su lugar, licencia su tecnología a otros fabricantes, como Qualcomm, Samsung y Apple, quienes personalizan los diseños para sus necesidades específicas.
5. **Modularidad:** ARM permite a los diseñadores de chips incluir módulos adicionales, como gráficos integrados o capacidades de procesamiento digital, lo que permite personalizar el rendimiento del procesador.

## Arquitecturas ARM Comunes

1. **ARMv7:** Utilizada en muchos dispositivos móviles y tabletas. Incluye soporte para 32 bits y se caracteriza por su capacidad de ejecutar aplicaciones de alto rendimiento.
2. **ARMv8:** Introdujo soporte para arquitectura de 64 bits, lo que permite un acceso a mayores capacidades de memoria y un mejor rendimiento en aplicaciones exigentes.
3. **ARM Cortex:** Una familia de procesadores basada en la arquitectura ARM, que incluye varios modelos optimizados para diferentes aplicaciones, desde la eficiencia energética hasta el rendimiento en cálculos intensivos.

## Aplicaciones

1. **Dispositivos Móviles:** Utilizada en la mayoría de los smartphones y tabletas.

2. **Sistemas Operativos Embebidos:** Presente en electrodomésticos, automóviles y dispositivos de IoT.
3. **Servidores y Computación de Alto Rendimiento:** Cada vez más utilizada en centros de datos debido a su eficiencia energética y rendimiento.

### 3.2. Descripción general de la arquitectura

El **Proyecto de Código Abierto de Android (AOSP)** es el código fuente de Android, disponible públicamente y modificable. Cualquier persona puede descargar y modificar **AOSP** para su dispositivo. **AOSP** proporciona una implementación completa y funcional de la plataforma móvil Android.

Hay dos niveles de compatibilidad para los dispositivos que implementan **AOSP**: compatibilidad con **AOSP** y compatibilidad con Android. Un dispositivo compatible con **AOSP** debe cumplir con la lista de requisitos del Documento de Definición de Compatibilidad (**CDD**). Un dispositivo compatible con Android debe cumplir con una lista de requisitos de Software del Vendedor (**VSR**), así como pruebas como las del Conjunto de Pruebas del Vendedor (**VTS**) y el Conjunto de Pruebas de Compatibilidad (**CTS**). Para más información sobre la compatibilidad con Android, adjunto la documentación oficial de Android al final de este documento.

### 3.3. Arquitectura de la pila de software de AOSP

La pila de software para **AOSP** contiene las siguientes capas:

1. **Aplicación Android:** Una aplicación creada únicamente usando la API de Android. La Google Play Store se utiliza ampliamente para encontrar y descargar aplicaciones Android, aunque existen muchas otras alternativas. En algunos casos, un fabricante de dispositivos puede querer preinstalar una aplicación Android para soportar la funcionalidad básica del dispositivo. Si te interesa desarrollar aplicaciones Android, consulta [developers.android.com](https://developers.android.com).

2. **Aplicación privilegiada:** Una aplicación creada utilizando una combinación de la **API** de Android y APIs del sistema. Estas aplicaciones deben ser preinstaladas como aplicaciones privilegiadas en un dispositivo.
3. **Aplicación del fabricante del dispositivo:** Una aplicación creada utilizando una combinación de la **API** de Android, la **API** del sistema y acceso directo a la implementación del marco de Android. Dado que un fabricante de dispositivos puede acceder directamente a APIs inestables dentro del marco de Android, estas aplicaciones deben ser preinstaladas en el dispositivo y solo pueden actualizarse cuando se actualiza el software del sistema del dispositivo.
4. **API del sistema:** La **API** del sistema representa APIs de Android disponibles solo para socios y OEMs para su inclusión en aplicaciones agrupadas. Estas APIs están marcadas como `@SystemApi` en el código fuente.
5. **API de Android** La **API** de Android es la **API** pública disponible para desarrolladores de aplicaciones Android de terceros. Para más información sobre la **API** de Android, consulta la referencia de la **API** de Android.
6. **Marco de Android:** Un grupo de clases Java, interfaces y otro código precompilado sobre el que se construyen las aplicaciones. Partes del marco son accesibles públicamente a través del uso de la **API** de Android. Otras partes del marco están disponibles solo para **OEMs** mediante el uso de las APIs del sistema. El código del marco de Android se ejecuta dentro del proceso de una aplicación.
7. **Servicios del sistema:** Los servicios del sistema son componentes modulares y específicos, como `system_server`, `SurfaceFlinger` y `MediaService`. La funcionalidad expuesta por la API del marco de Android se comunica con los servicios del sistema para acceder al hardware subyacente.
8. **Tiempo de ejecución de Android (ART):** Un entorno de ejecución de Java proporcionado



por **AOSP**. **ART** realiza la traducción del bytecode de la aplicación en instrucciones específicas del procesador que son ejecutadas por el entorno de ejecución del dispositivo.

**9. Capa de abstracción de hardware (HAL):** Una **HAL** (Hardware Abstraction Layer) es una capa de abstracción con una interfaz estándar para que los proveedores de hardware la implementen. Las **HAL** permiten que Android sea agnóstico respecto a las implementaciones de controladores de bajo nivel. Usar una **HAL** permite implementar funcionalidad sin afectar o modificar el sistema de nivel superior. Para más información, consulta la descripción general de **HAL**.

**10. Demonios y bibliotecas nativas:** Los demonios nativos en esta capa incluyen **init**, **healthd**, **logd** y **storaged**. Estos demonios interactúan directamente con el kernel u otras interfaces y no dependen de una implementación **HAL** basada en espacio de usuario.

Las bibliotecas nativas en esta capa incluyen **libc**, **liblog**, **libutils**, **libbinder** y **libselinux**. Estas bibliotecas nativas interactúan directamente con el kernel u otras interfaces y no dependen de una implementación HAL basada en espacio de usuario.

**11. Kernel:** El kernel es la parte central de cualquier sistema operativo y se comunica con el hardware subyacente de un dispositivo. Siempre que sea posible, el kernel de **AOSP** se divide en módulos agnósticos al hardware y módulos específicos del proveedor. Para una descripción, incluidas definiciones, de los componentes del kernel de **AOSP**, consulta la descripción general del kernel.

## 4. En búsqueda de la comunidad AOSP

Esta sección discute la falta de una comunidad cohesiva en torno al Android Open Source Project (AOSP), a pesar de ser una de las iniciativas de código abierto más exitosas, con miles de desarrolladores involucrados.

#### 4.1. AOSP y su Comunidad:

AOSP tiene entre 10,000 y 100,000 desarrolladores, pero dicha comunidad es casi invisible. Esta situación se compara con la **paradoja de Fermi**: Muchos deben estar allí, pero no se ven.

#### 4.2. Aislamiento de Desarrolladores:

Los desarrolladores tienden a trabajar en aislamiento, con poca comunicación entre equipos, incluso dentro de la misma empresa. La falta de oportunidades para interactuar y compartir experiencias son obstáculos para contribuir al Open Source no propietario y generar alternativas a el duopolio actual de los SO's móviles.

#### 4.3. Beneficios de una Comunidad Más Fuerte:

El hecho de crear una comunidad más unida beneficiaría a todos los involucrados, facilitando el intercambio de trabajo y experiencias. Esto sugiere implícitamente que no se afectaría o amenazaría a los proveedores actuales de dispositivos móviles.

#### 4.4. Proyecto Yocto:

El Proyecto Yocto es una iniciativa de código abierto que proporciona herramientas y recursos para crear sistemas operativos Linux personalizados para dispositivos embebidos. Su objetivo principal es facilitar el desarrollo de distribuciones Linux adaptadas a las necesidades específicas de hardware y aplicaciones, permitiendo a los desarrolladores construir imágenes de software optimizadas y personalizables. A través de un conjunto de herramientas como BitBake y recetas, los usuarios pueden gestionar las dependencias de software, automatizar el proceso de construcción y asegurar la consistencia en diferentes plataformas. Además, el Proyecto Yocto es altamente modular, lo que permite integrar una amplia variedad de componentes y bibliotecas, lo que lo convierte en una opción popular para el desarrollo de productos embebidos en sectores como automoción, IoT y telecomunicaciones.

Yocto tiene una comunidad más visible y activa, con mejores canales de comunicación y eventos. A comparación de **AOSP** que carece de recursos similares y comunicación efectiva.

#### 4.5. Propuestas de Mejora:

##### Puntos Clave:

1. **Mejorar los canales de comunicación:** Por ejemplo, creando grupos en plataformas como Telegram, Matrix, IRC, entre otros.
2. **Fomentar sentido de comunidad y encuentros (meetups):** Tanto en línea como presenciales, es necesario desarrollar un sentido de comunidad.
3. **Necesidad de Liderazgo:** Se requiere encontrar voluntarios para roles de liderazgo dentro de la comunidad AOSP.
4. **Creación de una organización independiente en apoyo al AOSP:** Eventualmente, se sugiere la creación de una organización independiente para apoyar la comunidad AOSP.
5. **Rol de Google:** Google actúa como gatekeeper, limitando qué parches se aceptan en AOSP. La comunidad necesita más apertura en la aceptación de contribuciones.
6. **Recursos y Formación:** Proporcionar capacitación y recursos para desarrolladores, facilitando talleres, conferencias y documentación para mejorar las habilidades en el desarrollo de AOSP.
7. **Financiación y Patrocinios:** Buscar financiamiento a través de donaciones, patrocinios o colaboraciones con empresas del sector tecnológico para apoyar iniciativas, eventos y proyectos dentro de la comunidad.
8. **Gestión de Proyectos:** Actuar como una plataforma centralizada para coordinar esfuerzos en proyectos específicos dentro de AOSP, promoviendo la colaboración entre diferentes grupos

y desarrolladores.

9. **Promoción y Visibilidad:** Aumentar la visibilidad del AOSP a través de campañas de marketing, eventos y participación en conferencias, atrayendo a más desarrolladores y empresas a la comunidad.
10. **Soporte Técnico:** Ofrecer soporte y asesoría técnica a los desarrolladores, ayudando a resolver problemas y optimizar el uso de AOSP en diferentes dispositivos y aplicaciones.
11. **Mantenimiento de la Infraestructura:** Ayudar en la gestión y mantenimiento de la infraestructura necesaria para el desarrollo y la colaboración, como servidores y herramientas de gestión de versiones

## 5. Android en el Contexto de la Vigilancia

Android, como sistema operativo ampliamente utilizado en dispositivos móviles, ha generado preocupaciones significativas sobre la privacidad y la recolección de datos en el contexto de la vigilancia. Debido a su naturaleza conectada y a las múltiples aplicaciones que utilizan servicios en línea, Android puede facilitar la recopilación y el intercambio de datos personales, lo que ha llevado a un escrutinio creciente por parte de usuarios, activistas y reguladores.

### 5.1. Recolección de Datos

Los dispositivos Android recopilan una variedad de datos del usuario, que pueden incluir información de ubicación, historial de navegación, contactos, mensajes y hábitos de uso de aplicaciones. Esta recolección de datos se realiza a menudo para mejorar la experiencia del usuario y ofrecer servicios personalizados, como recomendaciones de aplicaciones o publicidad dirigida. Sin embargo, muchos usuarios no son plenamente conscientes de la cantidad de información que se está recopilando ni de cómo se utiliza.

Además, muchas aplicaciones de Android requieren permisos amplios que permiten el acceso a datos sensibles. Si bien los usuarios pueden optar por no otorgar ciertos permisos, muchas veces esto limita la funcionalidad de las aplicaciones, lo que puede llevar a la aceptación de permisos que comprometen la privacidad.

## 5.2. Vigilancia y Control

En algunos casos, las aplicaciones y servicios basados en Android han sido utilizados para fines de vigilancia. Gobiernos y agencias de seguridad pueden acceder a los datos recopilados por aplicaciones o incluso a información directa de dispositivos a través de herramientas de espionaje masivo. La capacidad de rastrear la ubicación de un dispositivo, por ejemplo, puede ser utilizada para monitorear movimientos y actividades de individuos, planteando serias preocupaciones sobre la invasión de la privacidad.

La integración de servicios como **Google Location Services** y **Google Maps** también ha suscitado preocupaciones sobre la vigilancia en tiempo real, ya que estos servicios pueden rastrear la ubicación de los usuarios continuamente. Aunque se ofrecen opciones para desactivar el seguimiento de ubicación, la complejidad de la configuración de privacidad en Android puede llevar a que muchos usuarios no ajusten adecuadamente sus preferencias.

## 5.3. Preocupaciones sobre la Privacidad

Las preocupaciones sobre la privacidad en Android han llevado a un creciente interés en la regulación y la protección de datos. En respuesta a escándalos de privacidad y el uso indebido de datos, se han implementado normativas, como el Reglamento General de Protección de Datos (GDPR) en Europa, que busca garantizar que los usuarios tengan mayor control sobre su información personal.

Además, las alternativas de sistemas operativos que priorizan la privacidad, como **LineageOS** o **GrapheneOS**, están ganando popularidad entre aquellos que desean evitar la recolección de datos y la vigilancia asociadas con las versiones estándar de Android. Estas alternativas ofrecen un enfoque

más controlado y seguro para el uso de dispositivos móviles.

## 6. Conclusiones

El uso de Android en cualquier contexto plantea importantes desafíos en términos de privacidad, protección de datos, licenciamiento y alternativas ante el actual duopolio de software propietario. La recopilación masiva de información y la posibilidad de vigilancia por parte de gobiernos y empresas han suscitado preocupaciones que requieren atención tanto de los usuarios como de los reguladores de realizar conciencia sobre estos problemas. Por lo que es crucial para que los usuarios tomen decisiones informadas sobre el desarrollo de SO's embebidos FOSS, su privacidad y en ayudar a construir una comunidad.

### 6.1. Bibliografía

- Android versions: A living history from 1.0 to 15. [Artículo] Consultado de: <https://www.computerworld.com/article/1714347/android-versions-a-living-history-from-1-0-to-today.html>
- Build Lineage OS From Source - Custom ROM Build Guide. [[Tutorial] Vídeo] Consultado de: <https://youtu.be/9DHAn7msdUk>
- Android 1.0: 5 funciones que marcaron el inicio de un sistema operativo. [Artículo] Consultado de: <https://www.pasionmovil.com/plataformas/android/5-funciones-de-android-1-0/>
- Historia del iPhone. [Wikipedia] Consultado de: [https://es.wikipedia.org/wiki/Historia\\_del\\_iPhone](https://es.wikipedia.org/wiki/Historia_del_iPhone)
- Pantalla táctil. [Wikipedia] Consultado de: [https://es.wikipedia.org/wiki/Pantalla\\_t%C3%A1ctil](https://es.wikipedia.org/wiki/Pantalla_t%C3%A1ctil)
- AOSP: Así al Android Open Source de Google, que queda como opción para Huawei. [Artículo]

Consultado de: <https://www.xataka.com/aplicaciones/aosp-asi-al-android-open-source-google-que->

- HarmonyOS: Todo lo que necesitas saber sobre el sistema operativo de Huawei. [Artículo] Consultado de: <https://www.lavanguardia.com/andro4all/otros-android/harmonyos-todo-lo-que-necesit>
- Qué es AOSP: Así funciona el Android sin Google. [Artículo] Consultado de: <https://www.lavanguardia.com/andro4all/android/que-es-aosp-asi-funciona-el-android-sin-google>
- Instalación de GrapheneOS. [Guía] Consultado de: <https://grapheneos.org/install/>
- ¿Por qué usar GrapheneOS en lugar de LineageOS? [Artículo] Consultado de: <https://threecats.com.au/why-use-grapheneos-rather-than-lineageos-degoogled-custom-alternative-android-os>
- Arquitectura del kernel de Android. [Documentación] Consultado de: <https://source.android.com/docs/core/architecture/kernel>
- Documentación sobre Android. [Sitio oficial] Consultado de: <https://source.android.com>
- Android (sistema operativo). [Wikipedia] Consultado de: [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- Arquitectura ARM. [Artículo de LinkedIn] Consultado de: <https://www.linkedin.com/pulse/arm-architecture-midhun-yogesh-%E3%83%9F%E3%83%8D%E3%82%B7%E3%83%A5-dxlic>
- Android 1.0: 5 funciones que marcaron el inicio de un sistema operativo. [Artículo] Consultado de: <https://www.pasionmovil.com/plataformas/android/5-funciones-de-android-1-0/>
- Características de Android. [Documentación] Consultado de: <https://androidos.readthedocs.io/en/latest/data/caracteristicas/>
- Android en China. [Artículo] Consultado de: <https://www.examinechina.com/android-in-china/>
- Sistemas operativos móviles más comunes en China. [Estadística] Consultado de: <https://www.>

[statista.com/forecasts/1371013/most-common-smartphone-operating-systems-in-china](https://statista.com/forecasts/1371013/most-common-smartphone-operating-systems-in-china)

- Características de Android. [Documentación] Consultado de: <https://androidos.readthedocs.io/en/latest/data/caracteristicas/>
- Desglose del mercado de Android en China. [Artículo] Consultado de: <https://www.goclickchina.com/blog/breakdown-of-china-android-market/>
- Todo lo que necesitas saber sobre las principales tiendas de aplicaciones de Android en China. [Artículo] Consultado de: <https://www.nativex.com/en/blog/all-you-need-to-know-about-china-top-app-stores/>
- Discusión en Reddit sobre Android o iOS para llevar a China. [Foro] Consultado de: [https://www.reddit.com/r/chinalife/comments/1augzvj/android\\_or\\_ios\\_to\\_bring\\_to\\_china/?rdt=58701](https://www.reddit.com/r/chinalife/comments/1augzvj/android_or_ios_to_bring_to_china/?rdt=58701)
- Herramientas de privacidad y seguridad en Android en China. [Artículo] Consultado de: <https://www.comparitech.com/privacy-security-tools/blockedinchina/android/>
- ¿Qué significa Android y de dónde viene el nombre del sistema operativo de Google? [Artículo] Consultado de: <https://www.infobae.com/tecnologia/2024/01/22/que-significa-android-y-de-donde-viene-el-nombre/>
- Curiosidades: ¿Por qué Android se llama Android? [Artículo] Consultado de: <https://www.unocero.com/smartphones/curiosidades-por-que-android-se-llama-android/>
- Video sobre Android. [YouTube] Consultado de: <https://youtu.be/BkP6FTy0a4Y>
- La relación entre Linux y Android explicada. [Artículo de LinkedIn] Consultado de: <https://www.linkedin.com/pulse/linux-android-relationship-explained-nachwa-haddad>
- Rant sobre OxygenOS y Cyanogen. [Foro] Consultado de: <https://devrant.com/rants/910656/to-fellow-oneplus-users-if-youre-using-oxygenos-maybe-even-the-preinstalled-cyan>



- Awesome Android Custom ROMs. [Repositorio en GitHub] Consultado de: <https://github.com/devadigax/awesome-android-custom-rom>