

Inciso a) El sistema operativo android posee forma de pila y consiste de 4 capas:

- 1) Capa de Kernel.
- 2) Capa de librerías nativas.
- 3) Capa del framework de las aplicaciones.
- 4) Capa de aplicaciones.

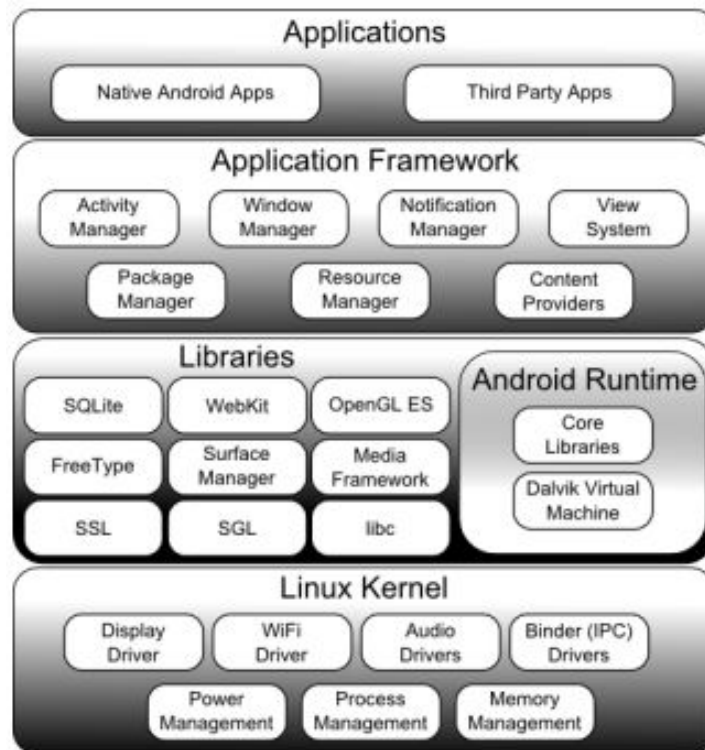


Figura 1: Arquitectura del Sistema Operativo Android

- 1) **Capa de Kernel:** provee funcionalidades básicas para el sistema operativo como manejo de procesos, manejo de memoria, de dispositivos (como la cámara), manejo del display, del teclado, etc... El kernel es de un sistema operativo Linux 2.6 mejorado para algunas tareas específicas como manejo de memoria procesos, energía e intercomunicación entre procesos. Linux es considerado una muy buena opción para realizar operaciones básicas como operaciones de redes y manejo de drivers.

Las mejoras agregadas al kernel son:

- a) **Drivers de alarma:** Utilizado para que el kernel realice ciertas tareas cuando ocurren eventos específicos.
- b) **Binder:** Se utiliza para controlar la intercomunicación entre procesos mas eficientemente en términos de velocidad y memoria. Utiliza pasaje de objetos por referencia entre procesos para que los mismos puedan guardarse solo en un segmento de memoria compartida entre los procesos.
- c) **Manejo de energía.**
- d) **Low Memory killer:** Es manejado por el "Out Of Memory handler". Cuando al dispositivo le falta memoria selecciona un proceso que esté corriendo en el background basándose en las directivas del usuario y lo termina.

- e) **Kernel Debugger:** Utilizado para poder controlar que los cambios que se hacen al kernel (debido a su naturaleza Open Source, cualquier puede hacer cambios) funcionen correctamente.
 - f) **Logger:** simplifica el proceso de troubleshooting
 - g) **Ashmem** (Android Shared Memory): facilita el uso de memoria compartida y al haber faltante de memoria puede elegir que parte de dicha memoria se puede liberar.
- 2) **Capa de librerías nativas:** Esta capa le permite al dispositivo manejar distintos tipos de datos específicos del hardware. Esta dividida en dos sub capas:
- a) **Librerías de Android:** Todas las librerías de esta capa están escritas en C++ y utilizadas a través de una interfaz de Java, se encargan de las tareas más “pesadas”. Algunas librerías de esta capa son: Libc, SQLite, Media Framework, Surface Flinger, WebKit.
 - b) **Runtime de Android:** Esta sub capa esta compuesta por la Dalvik Virtual Machine (DVM) y algunas librerías de Java. Esta máquina virtual cumple un proposito similar al de la Java Virtual Machine en Java que permite escribir código en un dispositivo y permitir usarlo en otros sin necesidad de reescribir el código. DVM está hecha para sistemas pequeños con poca memoria RAM y procesadores lentos.
- 3) **Capa del framework de las aplicaciones:** provee Interfaces para Programar Aplicaciones (APIs) y servicios de alto nivel usando clases en Java, Los programadores pueden acceder al framework de las APIs y utilizar componentes de las mismas para aprovechar la reutilización de su código. Cada componente tiene un ciclo de vida y propósito distinto que describe como el mismo será creado y destruido. Los componentes de las APIs pueden ser:
- a) Activity
 - b) Services
 - c) Content Providers
 - d) Package Manager
 - e) Window Manager
 - f) Hardware Services
 - g) Telephony Service
 - h) Location Service
- 4) **Capa de aplicaciones:** Contiene las aplicaciones nativas que vienen preinstaladas en cada dispositivo android como: Phone dialer, SMS Client, Navegador Web, contact Manager, etc.

b) Las características representativas de sistemas operativos basados en linux (como Android) son las siguientes:

- **Multitarea:** varios programas o aplicaciones (realmente procesos) pueden estar ejecutándose al mismo tiempo.
- **Multiplataforma:** corre en muchas CPUs (en este caso dispositivos móviles) distintas.
- Tiene **protección de la memoria entre procesos**, de manera que uno de ellos no pueda colgar el sistema.
- **Ejecución por demanda:** Linux sólo lee de la memoria principal aquellas partes de un programa que están siendo usadas actualmente.
- Política de copia en escritura para la compartición de páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse.
- Cuando alguno intenta escribir en esa memoria, la página se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.
- Memoria virtual usando paginación (sin intercambio de procesos completos) a disco: una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de añadir más áreas de intercambio sobre la marcha.
- La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y este puede a su vez ser reducido cuando se ejecuten grandes programas.
- Maquina virtuales que permite ejecución del sistema operativo en varios dispositivos
- **Multiusuario:** varios usuarios en la misma máquina al mismo tiempo.
- **Todo el código fuente está disponible**, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente.
- Puede ejecutarse/utilizarse en computadoras de pocos recursos

Inciso c) El artículo provee una descripción general del sistema operativo Android, sus herramientas y sus mejoras para favorecer su uso en dispositivos pequeños o de bajos recursos. Lo consideramos una buena primera aproximación a la estructura del Sistema Operativo Android y del manejo de dispositivos móviles en general. Podemos hacer una relación con los conceptos de threads, planificación y sincronización de procesos, manejo de memoria, inicio, componentes y herramientas de un sistema operativo vistos a lo largo de la materia y nos permitió conocer la forma en la cual podemos hacer una aplicación correcta de los mismos al momento de desarrollar o especificar un software para un dispositivo.

En una primera instancia se nos describe la arquitectura del sistema operativo android, en la cual podemos observar la elección del kernel del sistema operativo linux, el cual vimos en profundidad a lo largo de la materia y conlleva ventajas como desventajas que se describirán posteriormente. Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El kernel también actúa como una capa de abstracción entre el hardware y el resto de la pila de software (la arquitectura). Las mejoras realizadas sobre el kernel favorecen en su totalidad a las características de Linux profundizadas en la materia, todas conllevan un mejor uso de recursos, ampliación del paralelismo entre tareas y mejor uso de la memoria del dispositivo. En sí podemos observar que Android está fuertemente atado a recursos de alto nivel puesto que sus partes más fundamentales están implementadas en Java, C++ y C (lenguajes de alto nivel respecto de los contenidos vistos en esta materia)

Tuvimos una primera aproximación a los problemas de seguridad relacionados a los conceptos de la materia y como afectan en el sistema operativo descrito en el artículo, ventajas y desventajas del sistema operativo linux y sus mecanismos de memoria compartida en este ámbito, posibles problemas o brechas de seguridad al momento de no respetar los permisos del sistema o de asignarlos de manera errónea a las aplicaciones o procesos y la vulnerabilidad de nuestros datos debido a los inconvenientes antes mencionados relacionados con la selección del kernel, de las mejoras realizadas sobre el mismo y con el manejo libre que se le puede dar al usuario sobre el dispositivo y el sistema operativo lo cual puede abrir un gran debate en cuanto a la libertad que debería dársele al usuario o las restricciones que se aconsejaría imponer sobre el mismo.

En nuestra opinión el artículo nos presenta de forma intuitiva la arquitectura del sistema operativo Android y es una buena primera aproximación para continuar investigando respecto del tema y para profundizar sobre todos los conceptos vistos en las materias Sistemas Operativos, Arquitectura de Computadoras para Ingeniería y Organización de Computadoras. Nos permitió entender lo fundamental de un sistema operativo utilizado globalmente y nos motivó a indagar en mayor profundidad sobre el mismo y sobre los conceptos aprendidos (en principal sobre el uso de memoria y la optimización del uso de procesos) por nuestros propios medios.