

## Análisis de Comparación de Android y GNU / Linux

### [ Comparison Analysis Android and GNU / Linux ]

*Gary Reyes Zambrano*

Facultad de Ciencias Matemáticas y Físicas,  
Universidad de Guayaquil,  
Guayaquil, Ecuador

---

Copyright © 2016 ISSR Journals. This is an open access article distributed under the ***Creative Commons Attribution License***, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**ABSTRACT:** With the advent of smartphones and applications that provide users originate research efforts to learn the basic software that allows the iteration between the device and the user, this is called the Android operating system for mobile devices. This paper explains the main features, architecture and components of the operating system, analyzing their kernel, making the final make a comparison with GNU / Linux and application models, drivers, kernel extensions, as a result of this investigation.

**KEYWORDS:** Analysis, Core, mobile device, operating system.

**RESUMEN:** Con la aparición de los teléfonos inteligentes y los aplicativos que estos proporcionan a sus usuarios, se originan esfuerzos de investigación para conocer el software básico que permite la iteración entre el dispositivo y el usuario, uno de los que predominan el mercado es el llamado sistema operativo para dispositivos móviles Android. Este trabajo permite conocer las principales características, arquitectura y componentes de éste sistema operativo, analizando su kernel, logrando al final realizar una comparativa con GNU/Linux y sus modelos de aplicaciones, drivers, extensiones del kernel, como resultado de ésta investigación.

**PALABRAS CLAVES:** Análisis, Núcleo, dispositivo móvil, sistema operativo.

#### 1 INTRODUCCIÓN A ANDROID

Android, es una pila de software para dispositivos móviles que incluye un sistema operativo[1], desarrollado por Google y Open Handset Alliance [2], sin embargo la empresa pionera en su creación fue Android Inc., Andy Rubin, Rich Miner, Nick Sears y Chris White iniciaron ésta empresa cuya actividad se centraba en el desarrollo de software para teléfonos móviles [3], en el año 2005 fue adquirida por Google [4]. En el año 2007 un consorcio de empresas conocido como Open Handset Alliance anunciaron la primera plataforma para móviles construida sobre el kernel del sistema operativo Linux 2.6 [5]. Dentro del consorcio de empresas existen fabricantes de terminales móviles (Samsung, Lg, Htc y Motorola), fabricantes de componentes (Texas Instruments, Intel, Nvidia, etc.), fabricantes de software (Pv, Ebay, Esmertec, ect.), operadores de todo el mundo (T-Mobile, Telefónica, Italia Telecom, China Mobile, etc.) y también empresas de comercialización, lo que hace hacer ver un gran equipo detrás del proyecto [6]. Conjuntamente con éste consorcio de 78 compañías, dedicadas al desarrollo de estándares abiertos para dispositivos móviles, Google liberó la mayoría de código de Android bajo licencia Apache, una licencia libre y de código abierto [5]. En Octubre del 2008 es publicado el proyecto open source Android distribuido principalmente con licencia Apache 2.0, en éste mismo año la versión de Android 1.0 se instalaba en el modelo G1 de HTC Dream, un móvil deslizable hacia el costado con teclado QWERTY y una gran pantalla sensible al tacto [3].

**1.1 EVOLUCIÓN DE VERSIONES ANDROID**

Cada nueva versión, actualiza el sistema operativo para arreglar bugs o incorporar nuevas funcionalidades, cada actualización del sistema operativo Android es desarrollada bajo un nombre en código de un elemento relacionado con postres como: CupCake, Donut, Froyo, Ginebra, etc [5].

En la tabla 1 se encuentran las principales versiones de Android y sus características más relevantes.

**Tabla 1. Evolución de versiones del sistema operativo Android [5]**

<b>Versión</b>	<b>Fecha de liberación</b>	<b>Características relevantes</b>
Android 1.1 Banana Bread	9 de febrero 2009	Guarda archivos adjuntos en mensajes, mejoras en las llamadas a manos libres.
Android 1.5 Cupcake	30 de abril de 2009	Graba y reproduce videos en Camcorder, soporte para Bluetooth A2DP AVRCP, sube videos a Youtube desde el teléfono
Android 1.6 Donut	15 de septiembre de 2009	Experiencia mejorada en Android Market, actualización de soporte para CDMA/EVDO, 802.1x, VPN y text-to-speech
Android 2.0/2.1 Eclair	26 de octubre de 2009	Mejoras en Google Maps 3.1.2, Bluetooth 2.1, Zoom digital, SDK 2.0.1 y SDK 2.1
Android 2.2 Froyo	20 de mayo de 2010	Funcionalidad de Wi-Fi hotspot y tethering por Browser, marcación por voz y compartir contactos por Bluetooth, Soporte para Adobe Flash 10.1
Android 2.3 GingerBread	6 de diciembre de 2010	SDK 2.3, soporte nativo para telefonía VoIP SIP, soporte para Near Field Communication, teclado multi-táctil rediseñado
Android 3.0 HoneyComb	22 de febrero de 2011	Soporte para video chat mediante Google Talk, escritorio 3D con widgets rediseñados, mayor soporte para redes Wi-Fi
Android 4.0 Ice Cream Sandwich	12 de octubre de 2011	Unifica el uso de teléfonos, tablets, televisores, netbooks, etc., Interfaz limpia y moderna con nueva fuente Roboto.
Android 4.1.2 Jelly Bean	30 de junio de 2012	Involucra el Proyecto Butter, el que usa anticipación táctil, incorpora triple buffer
Android 4.4 KitKat	31 de octubre 2013	Posibilidad de impresión mediante Wi-Fi, WebViews basadas en el motor de Chromium
Android 5.0 Lollipop	3 de noviembre 2014	Soporte para CPUs de 64 bits, OpenGL ES 3.1 y Adroid Extensión Pack (AEP) en configuraciones GPU mejoradas
Android 6.0 Marshmallow	5 de octubre 2015	Soporte oficial para tarjetas SD y USB, Compatibilidad con lápices bluetooth, Mejoras en el apartado de memoria RAM

Se observa que al inicio las liberaciones de versiones tenían una alta frecuencia, a medida que pasaron los años la frecuencia fue disminuyendo, hoy en día en promedio se realiza una liberación anual, lo que supone una alta estabilidad del Software.

**1.2 CARACTERÍSTICAS DE ANDROID Y COMPARACIÓN CON OTROS SISTEMAS OPERATIVOS**

Así también, se observa en la tabla 2 una comparación de características de rendimiento de Android con otros sistemas operativos de dispositivos móviles.

**Tabla 2. Características de Android vs. Otros S.O Móviles[1].**

Feature	Android	Symbian	Win Mobile
Portability	1	0.5	0
Reliability	1	1	1
Connectivity	1	1	1
Open system	1	0.5	0.5
Kernel size	0.5	1	0
Standards	1	0.5	0.5
Special features	1	0	0.5
Total	6.5	4.5	3.5

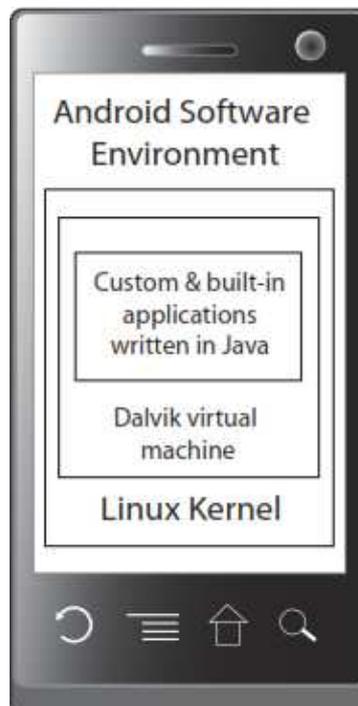
Las conclusiones de éste estudio, que midió la portabilidad, confiabilidad, conectividad, sistema abierto, tamaño del kernel, estándares y características especiales, son que Android es superior en comparación con los sistemas de su competencia y actúa como una plataforma emergente de software para dispositivos móviles, convirtiéndose en líder en plataformas móviles[1].

La constante liberación de nuevas versiones, las funcionalidades que se incorporan y su categoría de superior a su competencia, son claros indicios que la arquitectura y plataforma Android pueden ser consideradas como robustas y promueven su estudio, investigación y desarrollo.

### 1.3 LA PLATAFORMA ANDROID

La plataforma Android es hecha teniendo en cuenta diversos conjuntos de usuarios que utilizan las capacidades disponibles en Android en niveles diferentes. El interés que despierta en la industria móvil e industrias de diferentes arquitecturas de hardware se debe a dos aspectos: su naturaleza de código abierto y su modelo arquitectónico[1].

Android es un entorno de software integrado para dispositivos móviles, no es una plataforma de hardware. Aprovecha su núcleo de Linux para la interfaz con el hardware, por lo que se puede ejecutar en diferentes dispositivos de múltiples fabricantes[4]. Las aplicaciones son desarrolladas en Java, formando el entorno de software que se observa en la figura 1.



**Figura 1. Entorno de software Android[4]**

Android incluye un sistema operativo basado en el kernel de Linux, una rica interfaz de usuario, diversas aplicaciones de usuario final, librerías de código, los entornos de aplicaciones, soporte multimedia, etc. Los componente subyacentes del sistema operativo están escritos en C o C++, las aplicaciones de usuario se construyen para Android en Java, incluso las aplicaciones incorporadas están escritas en Java. Una de las características de la plataforma Android es que no hay diferencia entre las aplicaciones y las aplicaciones incorporadas que se crean con el SDK (kit de desarrollo de software de Android)[4].

El término plataforma se refiere a Android en sí, el entorno de software, incluyendo su código binario, bibliotecas de código y cadena de herramientas. Los emuladores de Android disponibles en el SDK son simples componentes de la plataforma Android[4].

Como se aprecia, Android es una plataforma cuya construcción base utiliza el núcleo Linux 2.6 que es la capa que realiza la abstracción del hardware y el resto del sistema, ofrece a la capa de software los diversos servicios existentes como la seguridad, la gestión de procesos y memoria, la pila de red y el modelo de drivers. Por encima del núcleo de Android se han diseñado una serie de capas que complementan un entorno de desarrollo[7].

Es importante para éste estudio, conocer estas capas, sus características y principales funcionalidades, esto es la descripción de lo que se conoce como arquitectura de Android.

## 2 COMPONENTES DE ANDROID (ARQUITECTURA)

La arquitectura de Android se la describe en capas, como se observa en la figura 2, ésta arquitectura está conformada por la capa "Linux Kernel", la capa "Android Runtime", la capa "Libraries", la capa "Application framework" y la capa "Applications".

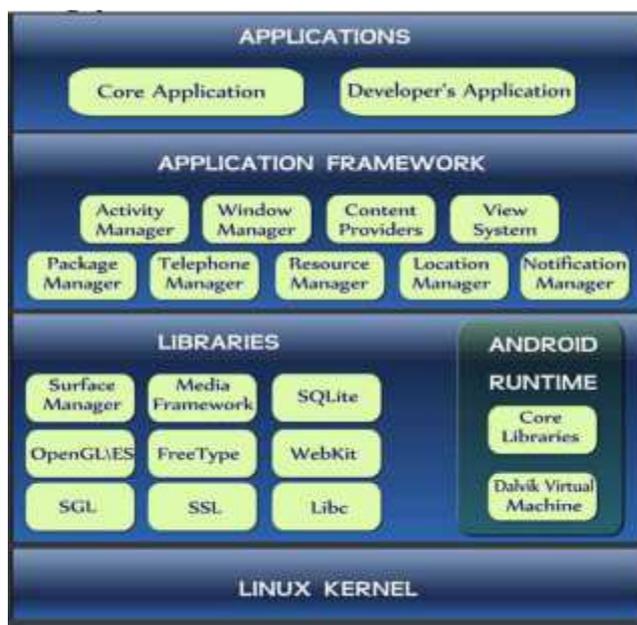


Figura 2. Arquitectura Android [8]

### 2.1 KERNEL LINUX

En el nivel más bajo se encuentra el núcleo de Linux. Android utiliza Linux para sus controladores de dispositivos, gestión de memoria, gestión de procesos, y la creación de redes[9].

### 2.2 ANDROID RUNTIME

Incluye un conjunto de las librerías del núcleo que proporciona la mayor parte de la funcionalidad disponible en las bibliotecas del núcleo del lenguaje de programación Java. Todas las aplicaciones de Android se ejecutan en su propio proceso, con su propia instancia de la máquina virtual Dalvik. La Dalvik VM ejecuta archivos en el formato Dalvik ejecutable

(.dex), que está optimizado para la mínima cantidad de memoria. La DVM (por sus siglas en inglés) está basada en registros y ejecuta clases compiladas por el compilador del lenguaje Java que se han transformado en el formato .dex mediante la herramienta "dx" incluida[10].

### 2.3 LIBRARIES

Son las bibliotecas nativas de Android, que se encuentran escritas internamente en C / C++, pero pueden ser invocadas a través de interfaces Java. En ésta capa se encuentran la superficie Manager (para la composición de las ventanas), gráficos 2D y 3D, codecs de medios (mpeg – 4, h.264, mp3, etc.), la base de datos SQL (SQLite) y un motor de renderizado nativo (WebKit)[9].

### 2.4 APPLICATION FRAMEWORK

Los desarrolladores tienen pleno acceso a las mismas API del framework utilizado por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes, cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede entonces hacer usos de esas capacidades (sujeto al marco de reglas de seguridad). Este mismo mecanismo permite que componentes sean reemplazados por el usuario. Todas las aplicaciones trabajan en un conjunto de servicios que incluyen: vistas, proveedores de contenido, administrador de recursos, administrador de notificaciones, administrador de actividades[11].

### 2.5 APPLICATIONS

Esta capa incluye las aplicaciones incluidas por defecto de Android y aquellas que el usuario vaya añadiendo posteriormente, ya sean de terceras empresas o de su propio desarrollo. Todas estas aplicaciones utilizan los servicios, las API y librerías de los niveles/capas anteriores[12].

## 3 COMPARATIVA ANDROID VS. GNU/LINUX: MODELO DE APLICACIONES, MODELO DE DRIVERS, EXTENSIONES DEL KERNEL

### 3.1 EL KERNEL DE ANDROID

Android está construido un núcleo de Linux y una avanzada y optimizada Máquina Virtual para sus aplicaciones Java. Ambas tecnologías son cruciales para Android. El núcleo de Linux proporciona agilidad y portabilidad para las numerosas opciones de hardware. El entorno Java de Android es clave porque es accesible al gran número de desarrolladores en Java[4].

### 3.2 COMPARATIVA

El núcleo Linux en su versión 2.6 fue elegido debido a su demostrado modelo de controlador, sus controladores existentes, la gestión de memoria y procesos, redes de soporte junto con otros servicios del núcleo del sistema operativo. Además del núcleo de Linux se añadieron diversas bibliotecas a la plataforma con el fin de soportar una funcionalidad superior. Muchas de estas bibliotecas se originan a partir de proyectos de código abierto; sin embargo el equipo de Android ha creado su propia biblioteca de C, por ejemplo para resolver los conflictos de licencias. También desarrollaron su propio motor de Java runtime, optimizado para los limitados recursos disponibles en una plataforma móvil llamada Dalvik Virtual Machine. Por último el Marco de trabajo de aplicaciones era creado con el fin de ofrecer las bibliotecas del sistema de una manera concisa a las aplicaciones de usuario final[13].

El kernel de Linux soporta diferentes arquitecturas de destino. Sin embargo sólo dos son totalmente soportadas por Android en éstos momentos: x86 y ARM. La arquitectura x86 para Android es principalmente dirigida a dispositivos móviles de internet (MID), mientras que la arquitectura ARM es prevalente en teléfonos móviles. Estas arquitecturas se utilizan normalmente para diferentes sistemas informáticos. La x86 se utiliza para fines generales de computadoras de escritorio/portátiles/servidores mientras que ARM es ampliamente utilizada en dispositivos móviles, una comparación de estas 2 arquitecturas proporciona una visión fuerte de las diferencias fundamentales entre Linux y android.

Linux fue creado inicialmente en 1991 como un sistema operativo de escritorio de código abierto. El núcleo de Linux fue creado desde el sistema operativo Minix (una versión del sistema Unix académico) no era muy compatible con las nuevas características de 32 bits del Intel 80386. La familia del x86 más tarde llegó a dominar el mercado de escritorio y dio lugar a la mayoría de los ordenadores personales utilizando la arquitectura x86 de hoy. En consecuencia la mayor parte del desarrollo del kernel de Linux hoy en día se centra en ésta familia.

Los teléfonos inteligentes en comparación no tenían un claro líder del mercado Nokia, RIM, Apple, HTC y Samsung todos tienen una significativa fracción del mercado de los teléfonos inteligentes. A pesar de que ningún fabricante domina el segmento de los teléfonos inteligentes, todos ellos utilizan principalmente la arquitectura ARM, de hecho a partir de 1996 el 98% de todos los teléfonos móviles tenían al menos 1 microprocesador ARM en ellos. La gran popularidad de la plataforma ARM es en gran parte debido a su enfoque en las funciones de ahorro de energía. A diferencia de los sistemas de escritorio, los teléfonos móviles se basan en energía de la batería para el funcionamiento y por lo tanto el consumo de energía es un factor de diseño primario.

Entre las diferencias más fundamentales entre estas dos arquitecturas está la filosofía de diseño de instrucciones. La familia x86 es ante todo un CISC (conjunto de instrucciones complejas de Computadora) mientras que el diseño de ARM es una arquitectura RISC (conjunto de instrucciones Reducidos de computador). La memoria es un recurso escaso en dispositivos embebidos, debido a su tamaño, costo y restricciones de potencia. ARM para compensar esta preocupación, ofrece un segundo conjunto de instrucciones de 16 bits denominado "pulgar", que puede ser intercalado con instrucciones regulares ARM de 32 bits. Este conjunto de instrucciones adicionales puede reducir el tamaño del código hasta en un 30%, a expensas de algo de rendimiento.

La familia ARM también tiene un enfoque de diseño en todo el sistema de bajo consumo de energía. Por ejemplo, el ARM7100 consume sólo 72mW cuando se opera a 14MIPS, 33mW en modo inactivo y 33uW durante la espera. En comparación, el más pequeño de los procesadores x86 de Intel, Atom, es capaz de trabajar a aproximadamente 1W.

En resumen, la actual cuota del mercado del procesador ARM dentro del mercado de la telefonía móvil, se da por un énfasis en el pequeño tamaño de su código y su operación de bajo consumo de energía, principal razón para su uso generalizado en teléfonos Android. En contraste con la mayoría de pc de escritorio/portátiles/servidores que utilizan Linux.

Android está basado en Linux, pero no utiliza un kernel estándar de Linux. El Kernel mejorado de Android incluye un controlador de alarma, controlador de memoria compartida (ashmem), conductor aglomerante (entre procesos de la interfaz de comunicación), la administración de energía, Eliminador de memoria baja, Depurador del núcleo y el registrador. Estas mejoras al kernel han contribuido a la comunidad de código abierto bajo licencia Pública GNU (GPL)[13].

Otro de los elementos importantes del Kernel son los drivers, estos se ejecutan en modo núcleo y su fiabilidad es significativa para los sistemas operativos. Los drivers utilizan funciones de extensión del kernel para operar y administrar dispositivos. Sin embargo, en algunos casos, estas funciones pueden devolver errores que son mortales para los drivers, provocando fallos en la asignación de memoria y fallos en la configuración[14].

Entre las diferencias más fundamentales entre estas dos arquitecturas está la filosofía de diseño de instrucciones. La familia x86 es ante todo un CISC (conjunto de instrucciones complejas de Computadora) mientras que el diseño de ARM es una arquitectura RISC (conjunto de instrucciones Reducidos de computador). La memoria es un recurso escaso en dispositivos embebidos, debido a su tamaño, costo y restricciones de potencia. ARM para compensar esta preocupación, ofrece un segundo conjunto de instrucciones de 16 bits denominado "pulgar", que puede ser intercalado con instrucciones regulares ARM de 32 bits. Este conjunto de instrucciones adicionales puede reducir el tamaño del código hasta en un 30%, a expensas de algo de rendimiento.

La familia ARM también tiene un enfoque de diseño en todo el sistema de bajo consumo de energía. Por ejemplo, el ARM7100 consume sólo 72mW cuando se opera a 14MIPS, 33mW en modo inactivo y 33uW durante la espera. En comparación, el más pequeño de los procesadores x86 de Intel, Atom, es capaz de trabajar a aproximadamente 1W.

En resumen actual cuota de mercado del procesador ARM del mercado de la telefonía móvil combinado con un énfasis en el tamaño pequeño código y operación de bajo consumo es la principal razón de su uso generalizado en los teléfonos Android. Este es un fuerte contraste con la mayoría / ordenador portátil / servidor de escritorio sistemas Linux, lo que comúnmente se utilizan en lugar de los procesadores x86.

El NDK de Android es una herramienta complementaria configurada para el SDK de Android, destinado a permitir que los desarrolladores implementen y mejoren el rendimiento de sus aplicaciones a través código nativo. Aunque el marco Android está destinado simplemente para aplicaciones basadas en Java, el NDK provee las herramientas necesarias para desarrollar aplicaciones de Android con el uso de código nativo producido a través de C / C ++[15].

Para el desarrollo de una aplicación Android, Google ha creado una estructura cuyas componentes más importantes son:

- **Activity:** Es el componente principal encargada de mostrar al usuario la interfaz gráfica, es decir, Activity es el equivalente a una ventana, y es el medio de comunicación entre la aplicación y el usuario.

- **Listeners:** Son mensajes que producen notificaciones o cambios de estado, que al ser recibidos por actividades o servicios pueden levantar procesos.
- **Views:** Son los componentes de la interfaz de usuario, diferentes vistas pueden agruparse a través de grupos logrando una jerarquía, esto se logra a través de la disposición de los componentes a través de un archivo XML.
- **Service:** Son componentes que ejecutan operaciones en segundo plano y no tienen una interfaz de usuario.
- **Content Provider:** Para almacenar información en este SO se hace mediante bases de datos, pero cuando lo que se quiere es compartir la información para que otras aplicaciones la usen, se utilizan los content provider, esta clase tiene métodos estándares que hace que las aplicaciones puedan consultar, guardar, o modificar la información general de las aplicaciones.
- **Manifest:** El archivo AndroidManifest.xml es donde se configura la aplicación, se agregan actividades, asignan permisos, etc.
- **Broadcast Receivers:** Son componentes que responden a avisos y anuncios de difusión (broadcast). Estos avisos provienen del sistema (batería baja, una llamada entrante, etc.) y de aplicaciones (pasando avisos de una aplicación a otra) [16].

Estudios demuestran que el 99 % de las funcionalidades del kernel de Linux fueron reutilizados en Android y sólo el 0,7 % de archivos del Linux del kernel fueron modificados durante esta adaptación. El 95 % de los errores reportados en el kernel de Android están fijados por los desarrolladores de Linux. Los equipos de desarrollo deben considerar posible la adaptación de Linux a Android, ya que hay una buena probabilidad de que la mayor parte del esfuerzo de mantenimiento del sistema adaptado sea levantado por equipo de desarrollo de Linux[17].

En la mayoría de las distribuciones de Linux la biblioteca GNU C se utiliza para proporcionar las rutinas de biblioteca especificadas por la norma ISO C para los programas de lenguaje C. Muchos desarrolladores ven la biblioteca GNU C como inapropiada para las plataformas de memoria limitada, tales como sistemas embebidos. Por otra parte, esta biblioteca está disponible bajo la GNU Licencia Pública Menor (LGPL) y por lo tanto restringe la concesión de licencias de trabajos derivadas. Estas preocupaciones llevaron a los desarrolladores de Android a crear su propia biblioteca de C llamado "biónic"[13].

Android utiliza el sistema de archivos flash YAFFS, la primera NAND flash de Linux optimizado archivo sistema. Para los dispositivos móviles, los discos duros son demasiado grandes en tamaño, demasiado frágil y consumen demasiada energía de ser útil. Por el contrario, la memoria flash proporciona un tiempo de acceso de lectura rápida y una mejor cinética Resistencia a golpes que los discos duros. Existen fundamentalmente dos tipos diferentes de memoria flash en función de su técnica de construcción: NOR y NAND. Tampoco es de baja densidad, ofrece escrituras lentas lee y rápido. NAND es de bajo costo, alta densidad y ofrece las escrituras rápidas y lee lento. Incrustado Los sistemas están utilizando cada vez flash NAND y NOR para el almacenamiento de código y ejecución. Los sistemas de archivos de memoria flash tienen que hacer frente a estas limitaciones con el fin de proporcionar un robusto sistema de archivos.

Un sistema estándar de Linux normalmente no utiliza la memoria flash, pero en lugar de disco magnético unidades. Estas unidades tienen un formato común de forma predeterminada para utilizar la última versión del archivo Ext sistema. Al escribir estas líneas, Ext3 está actualmente en uso generalizado y se presenta aquí para la comparación con el sistema de archivos flash, YAFFS, en uso en Android[13].

#### 4 CONCLUSIONES

Se evidencia en el presente trabajo que Android utiliza el kernel de Linux, sin embargo existen muchas funcionalidades aún no explotadas, lo que abre un área de investigación y desarrollo, sin embargo aquellas funcionalidades a ser estudiadas deben ser las que consuman recursos mínimos como los que posee Android en un dispositivo móvil. Con respecto al Modelo de aplicaciones que nos brinda Androide es muy eficaz y ágil, sin embargo puede explotar algunas funciones de C nativas que utiliza Linux.

#### REFERENCIAS

- [1] N. Gandhewar and R. Sheikh, "Google Android : An Emerging Software Platform For Mobile Devices," no. 12, pp. 12–17, 2010.
- [2] C. Nimodia and H. R. Deshmukh, "ANDROID OPERATING SYSTEM," vol. 3, no. 1, p. 4015, 2012.
- [3] G. Herraiz Antón, "Android," 2012.

- [4] W. F. Ableson, R. Sen, C. King, and C. E. Ortiz, "ANDROID IN ACTION, THIRD EDITION," 2011.
- [5] Y. J. Molina Rivera, J. Sandoval Cardona, and S. A. Toledo Franco, "Sistema Operativo Android: Características y funcionalidad para dispositivos móviles," 2012.
- [6] J. Balaguero Peña, "Estudio de plataforma Android," 2008.
- [7] E. Castillo Montaña, "Cliente Moodle para Android," 2014.
- [8] C. Wang, "The research of Android System Architecture and application programming," pp. 785–790, 2011.
- [9] A. Vicente and J. Mozos, "A study of vulnerabilities on Android systems," no. September, 2013.
- [10] S. Singapati, "INTER PROCESS COMMUNICATION IN ANDROID," no. March, 2012.
- [11] A. Shanker, "Android Porting Concepts," pp. 129–133, 2010.
- [12] J. Aranaz Tudela, "Desarrollo de aplicaciones para dispositivos móviles sobre la plataforma android de google," 2009.
- [13] F. Maker and Y. Chan, "A Survey on Android vs . Linux," pp. 1–10, 2012.
- [14] H. Liu, Y. Wang, L. Jiang, and S. Hu, "PF-Miner: A New Paired Functions Mining Method for Android Kernel in Error Paths," *2014 IEEE 38th Annu. Comput. Softw. Appl. Conf.*, pp. 33–42, Jul. 2014.
- [15] "Native Pthread on Android Platform using Android NDK," no. November 2014, 2016.
- [16] M. A. Bustos, N. B. Perez, and M. Berón, "Plataformas para el desarrollo de aplicaciones móviles," 2012.
- [17] F. Khomh, H. Yuan, and Y. Zou, "Adapting Linux for Mobile Platforms : An Empirical Study of Android," pp. 629–632, 2012.