

Descubriendo la anatomía de una aplicación sobre Android

Benedik Jaseth Delgado Jaén¹; M. Vargas-Lombardo^{†2}

¹Licenciatura en Ingeniería en Sistemas y Computación
Facultad de Sistemas Computacionales
Centro Regional Universitario Tecnológico de Azuero
La Villa de Los Santos, Panamá
Universidad Tecnológica de Panamá
benedik.delgado@utp.ac.pa

²Grupo de Investigación de Salud Electrónica y Supercomputación (GISES)
Centro de Investigación, Desarrollo e Innovación en Tecnologías de la Información y las comunicaciones (CIDITIC)
Universidad Tecnológica de Panamá, Panamá
miguel.vargas@utp.ac.pa

(Recibido/received: 14-Junio-2012; aceptado/accepted: 03-Noviembre-2012)

RESUMEN

La plataforma de trabajo Android incluye, un sistema operativo, un middleware y un conjunto de aplicaciones clave. Sin embargo, la incorporación de servicios de acceso integrados a internet en estos dispositivos móviles, aumenta la exposición a los daños causados por varios tipos de malware. Este documento muestra las más importantes partes que componen la anatomía de una aplicación sobre el sistema operativo Android, el comportamiento y función de cada una de ellas y la importancia que tiene la estructura interna de Android la que, lo ha convertido en una plataforma confiable, segura y flexible frente a las necesidades del mercado actual.

Palabras Clave: Android; sistema operativo; archivo manifiesto; aplicaciones; máquina virtual.

ABSTRACT

The Android framework includes an operating system, middleware and a set of key applications. The incorporation of integrated access services to the Internet on such mobile devices, however, increases their exposure to damages inflicted by various types of malware. This paper shows the most important component parts of the anatomy of an application on the Android operating system, behavior and function of each and the importance of the internal structure of Android, which has become a platform, reliable, secure and flexible in response to current market needs.

Keywords: Android; Operating System; **Archivo manifiesto**; Applications; Virtual machine.

[†] Autor para la correspondencia

INTRODUCCIÓN

En el documento que a continuación se presenta, se han abordado los puntos más importantes para conocer un poco acerca de la metodología de desarrollo y el funcionamiento del sistema operativo Android. Se detallan, entre otros puntos, de interés, su anatomía o componentes principales, la forma en la que se encuentra estructurado, los mecanismos de seguridad que incluye y las múltiples posibilidades de personalización con las que se cuenta. A través de este documento el lector tendrá la oportunidad de conocer las innovaciones del mundo móvil y del alcance que brinda el “Pequeño Gran Gigante” Android.

Actualmente, las ciencias de la computación proponen una programación de sistemas, basados en la funcionalidad y manejo de datos de alta eficiencia, sin ser el sistema Android la excepción. Las aplicaciones para este sistema, requieren que el desarrollador conozca a fondo el ciclo de vida del proyecto, la terminología, el contexto, las actividades a desarrollar y los distintos bloques involucrados en todo el proceso de diseño, creación, implementación y mantenimiento de la aplicación.

Al ser Android, un tema de debate y múltiples cuestionamientos en nuestros días, surge la pregunta, ¿Qué es Android.?

Android es una solución completa de software de código libre para teléfonos y dispositivos móviles. Es un paquete que engloba un sistema operativo, un runtime de ejecución basado en Java, una serie de librerías de bajo y medio nivel y un conjunto inicial de aplicaciones destinadas al usuario final. Se distribuye bajo una licencia Apache versión 2 (ASL2), una licencia libre permisiva que permite la integración con soluciones de código propietario [1].

Este trabajo se organiza de la siguiente manera: en la sección II se describen la importancia y función de cada una de las partes que componen la anatomía de una aplicación sobre Android. En la siguiente sección, se detalla uno de los componentes más importantes de la estructura de Android, la máquina virtual de Dalvik, a través de la cual cobran vida todas las aplicaciones ejecutadas en este sistema operativo. Seguidamente, hablamos sobre las aplicaciones en Android, cómo están estructuradas cada una de ellas y el funcionamiento de cada uno de sus componentes. Continuando, en la sección V, damos un vistazo relevante a los Servicios en este sistema operativo, los cuales proveen la funcionalidad a las aplicaciones que en este entorno

ejecutemos. Finalmente, en la sección VI, abordamos el tema de la interfaz que ofrece Android, las partes que la conforman y los elementos que se encuentran disponibles y que ofrecen un elevado grado de personalización.

COMPONENTES DE LA ANATOMÍA DE UNA APLICACIÓN ANDROID

La anatomía de una aplicación Android está compuesta por sus actividades, las cuales, por lo general, son pantallas que permiten la implementación de las clases, los intent e intentfilters que son clases que permiten el desplazamiento entre pantallas, el Broadcast Intent Receiver, que es usado como un disparador de eventos frente a determinadas situaciones, los servicios, que son códigos que se ejecutan en segundo plano, sin ninguna interfaz de usuario y el content provider que, como su nombre lo indica, es el encargado de la gestión de datos.

A continuación se detalla cada una de estos componentes:

Activity: habitualmente es una pantalla de la aplicación. Cada actividad es implementada como una clase que extiende a la clase base. La clase mostrará una interfaz de usuario compuesta por Views que responderá a los distintos eventos que se produzcan. La mayoría de las aplicaciones están compuestas por varias pantallas, por lo que éstas, a su vez estarán compuestas por varias actividades. Las Activities deben estar declaradas en el archivo Android Manifest de la siguiente manera:

```
<manifest ... >
<application ... >
<activity android:name=".ExampleActivity" />
...
</application ... >
...
</manifest >
```

Intent e IntentFilters: son clases especiales para moverse de una pantalla a otra. Describe lo que una aplicación quiere hacer. Lo más importante de esta estructura es la acción y los datos para llevarla a cabo. La navegación entre pantallas se logra mediante la resolución de Intents. Este tipo de aplicaciones usan un mecanismo asíncrono de mensajería mediante el cual, se logra relacionar cada solicitud del sistema a una actividad adecuada.

Cuando la aplicación genera esa Intención (muchas están pre-definidas), Android busca la más adecuada para manejar la misma. Si no está registrada, el usuario debe definirla en el fichero AndroidManifest.xml para que se lance la actividad correspondiente. Este fichero es requerido por todas las aplicaciones [2]. Contiene los valores globales del paquete que se está usando, incluyendo los componentes de aplicación, la implementación de las clases para cada componente, qué tipo de datos puede manejar y cuándo pueden ser lanzados.

Broadcast Intent Receiver: se puede usar cuando se requiere que se ejecute algo como reacción a un evento externo. No usan la interfaz de usuario. Usan Notification Manager para alertar de que algo ha pasado [3].

Service: es un código que se ejecuta sin ninguna interfaz de usuario, en segundo plano, para permitir así la ejecución de otras actividades. Es importante notar que es posible conectarse a un servicio mediante la interfaz que la actividad pone a disposición del usuario [4].

Content Provider: son aplicaciones que establecen sus datos en ficheros, bases de datos, etc. Tienen sentido si se quiere compartir datos con otras aplicaciones. Es un servicio que da a las aplicaciones, capacidad de comunicación bilateral con otras aplicaciones de forma interna.

La figura 1, muestra la estructura interna del sistema Android. Se hace énfasis en que este sistema está estructurado de forma jerárquica; donde el nivel más alto lo ocupan las aplicaciones, seguido del marco de las aplicaciones o framework applications (administrador de actividades, administrador de paquetes, administrador de recursos, etc), las librerías y en el nivel más bajo, pero no menos importante, un kernel Linux que sirve como base de la pila de software y se encarga de las funciones más básicas del sistema: gestión de drivers, seguridad, comunicaciones, etc.

DALVIK, LA MÁQUINA VIRTUAL DE ANDROID.

Android contiene la máquina virtual Dalvik, diseñada por Dan Bornstein, con contribuciones de otros ingenieros de Google. Una máquina virtual es un software que emula a un ordenador y puede ejecutar programas como si fuera un ordenador real.

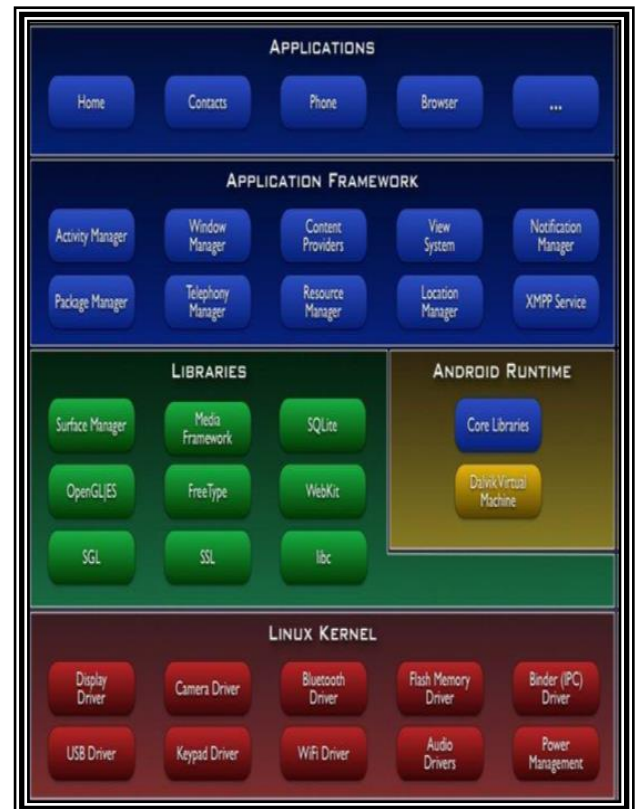


Figura 1. Arquitectura Android [5]

Dalvik es un ejemplo de las necesidades de eficacia, el deseo de un entorno de programación completo e incluso el enfrentamiento de ciertas restricciones de propiedad intelectual, dando todo ello como resultado, la innovación. El entorno Java de Android proporciona una completa plataforma de aplicaciones y resulta muy accesible debido a la popularidad del propio lenguaje Java [6].

Además, el rendimiento de las aplicaciones, en especial en entornos de memoria reducida como son los teléfonos, es imprescindible para el mercado de la telefonía móvil, aunque no sea el único problema[7].

Dalvik está optimizada para requerir poca memoria y está diseñada para que permita ejecutar varias instancias de la máquina virtual simultáneamente, delegando el control y la gestión de memoria al sistema operativo subyacente.[8] Se debe resaltar, que a diferencia de una máquina virtual Java, la cual se basa en registros, Dalvik está completamente basada en una estructura de pilas[9]

En la plataforma Android, el código Java es compilado en archivos de clase, pero después de los archivos clase se generan los archivos “DX”, con extensión “.dex”, conocidos como archivos ejecutables Dalvik [10]. A diferencia de un archivo clase, el cual contiene una sola clase, un archivo “DX” contiene varias clases que se ejecutan en la máquina virtual Dalvik.

forman parte de éstas. En términos informáticos, se puede decir que el contexto es la clase utilizada para administrar detalles específicos dentro de la configuración y que, es a través de éste, que se puede realizar instancias de las múltiples actividades que forman parte de la aplicación. Haciendo uso del contexto de la aplicación, se debe acceder a partes sensitivas de las aplicaciones como lo son: recuperación de paquetes

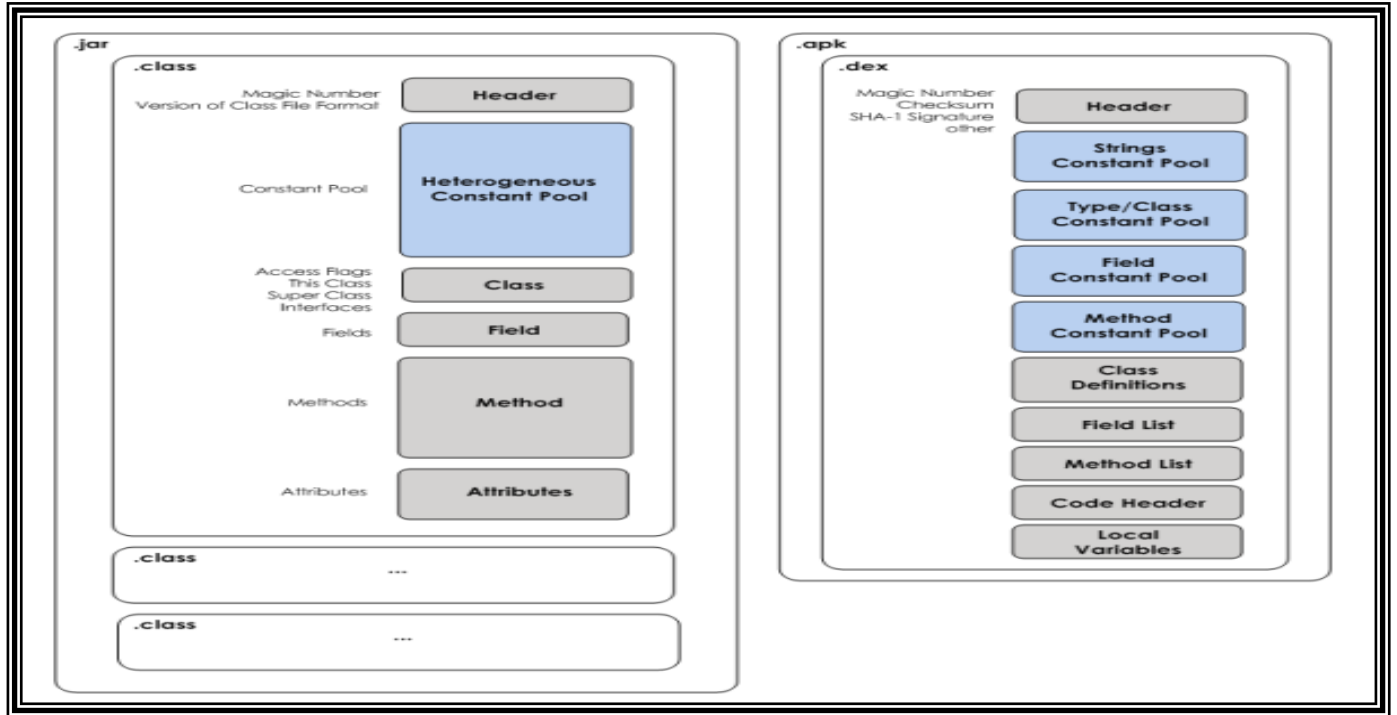


Figura 2. Comparativa de la estructura interna de un archivo .class y un archivo .dex (D. Ehringer)

LAS APLICACIONES EN ANDROID

Están programadas en Java, pero no corriendo sobre Java ME, sino sobre Dalvik, donde los códigos fuente se compilan a ficheros de bytecode .dex. Toda aplicación de este tipo hará uso de las distintas APIs expuestas por Android; de forma que los componentes encargados de realizar cada tarea puedan ser manipulados o reemplazados sin problemas, asegurando la máxima flexibilidad [11].

A continuación se explican las principales características, con las que cuenta una aplicación sobre Android.

Contexto de las aplicaciones Android:

A nivel superior dentro de las aplicaciones, el contexto es la referencia central para todas las funciones que

en la comunicación de servicios, solicitud de servicios de forma directa (ejemplo, servicios de localización), manejo privado de archivos y directorios, entre otros.

Framework Applications

Todos los desarrolladores de aplicaciones Android tienen acceso total al código fuente usado en las aplicaciones base. Esto ha sido diseñado de esta forma, para que no se generen cientos de componentes de aplicaciones distintas (que respondan a la misma acción) dando la posibilidad de que los programas sean modificados o reemplazados por cualquier usuario, sin tener que empezar a programar sus aplicaciones desde su origen [12].

Las librerías

Android incluye en su sistema un set de librerías C/C++, que son expuestas a todos los desarrolladores a través

del Framework de las aplicaciones Android, librería C del sistema, librerías de medios, librerías de gráficos, 3D, SQLite, etc [13].

- ✓ System C library: Librería standard de C (libc) optimizada para usarla en dispositivos móviles.
- ✓ Media Libraries: Librería que proporciona funciones para la reproducción y grabación de audio y vídeo en diferentes formatos entre ellos: MPEG4, H.264, MP3, AAC, AMR, JPG, PNG.
- ✓ Surface Manager: Gestiona el acceso al sistema de visualización y realiza la composición de capas 2D y 3D de diversas aplicaciones.
- ✓ LibWebCore: Un motor de navegación web para potenciar el navegador web o vistas webs incluidas en aplicaciones.
- ✓ SGL: Motor para gráficos 2D.
- ✓ 3D Libraries: Una implementación de las APIs de OpenGL ES 1.0, la cual permite el uso de aceleración 3D por hardware y la optimización para 3D por software.
- ✓ FreeType: Renderizado de mapas de bits y fuentes de texto.
- ✓ SQLite: Proporciona un motor de bases de datos relacional potente y ligero, disponible para todas las aplicaciones[14].

Runtime Android

Incluye un set de librerías base que proveen la mayor parte de las funcionalidades disponibles en las librerías base del lenguaje de programación Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Esta máquina virtual ha sido diseñada, de forma que un dispositivo pueda correr múltiples máquinas virtuales con un elevado nivel de eficiencia[15].

Permiso en las aplicaciones

Dentro de los sistemas operativos móviles más comercializados de los últimos años, (RIM de Blackberry, Android de Google, Symbian de Nokia), ninguna de las aplicaciones, al ser instaladas, traen habilitados por defecto, los permisos necesarios para su correcto funcionamiento. Es responsabilidad directa del usuario, establecer dichos permisos al momento de la instalación, estableciendo este, una equidad entre los privilegios otorgados a la aplicación y su alcance dentro del sistema[16].

Recursos de una aplicación Android

Todas las aplicaciones android están formadas de dos elementos: código de instrucciones (funcionalidad) y los datos (recursos)[17]. La funcionalidad está definida por el propio alcance de las aplicaciones; y los recursos incluyen, imágenes e iconos, archivos de audio y video y otros datos utilizados por la aplicación. Todos los recursos dentro de este tipo de sistemas son almacenados de forma separada dentro de una clase Java, con formato XML, mismo formato que tiene el archivo manifiesto antes mencionado.

LOS SERVICIOS, LA FUERZA INVISIBLE

Básicamente, un servicio no es más que una actividad sin interfaz de usuario. La interfaz de usuario solo muestra en primer plano, lo que el usuario realmente amerita ver, dejando en un segundo plano, centenares y hasta millares de procesos que están directamente relacionados con el funcionamiento de la aplicación[18]. Los servicios pueden ser utilizados con diferentes propósitos. Por lo general, son usados cuando no se necesita de una entrada por parte del usuario (un email, aplicaciones dentro de las redes sociales, verificación de actualizaciones del sistema operativo). Cabe señalar, que en las anteriores actividades existe cierto nivel de interacción por parte del usuario, pero son ejemplos válidos para la implementación de servicios. Al igual que las actividades, los servicios deben estar declarados en el archivo manifiesto de la siguiente manera:

```
<manifest ... >
...
<application ... >
<service android:name=".ExampleService" />
...
</application>
</manifest>
```

Estas son algunas de las circunstancias en las que es posible que desee poner en práctica o usar un servicio de Android:

- ✓ El correo electrónico o aplicación de red social puede implementar un servicio de forma rutinaria para comprobar si hay actualizaciones. (Nota: Hay otras implementaciones para el sondeo, pero esto es un uso común de servicios.)
- ✓ Una aplicación de fotos o de los medios de comunicación que mantiene los datos sincronizados en línea, podrían llevar a cabo un servicio para empaquetar y cargar contenido

nuevo en un segundo plano, cuando el dispositivo está inactivo.

- ✓ Una aplicación de edición de vídeo, puede descargar el procesamiento pesado para una cola en su servicio, con el fin de evitar que éste afecte el rendimiento global del sistema para tareas no esenciales.
- ✓ Una aplicación de noticias podría implementar un servicio de "pre-carga" mediante la descarga de contenidos de noticias de anticipación, cuando el usuario inicia la aplicación, para mejorar el rendimiento[11].

INTERFAZ ANDROID, LA VENTANA AL MUNDO

La mayoría de las aplicaciones Android requieren de una interfaz gráfica, que permita al usuario la interacción con la misma. La interfaz de usuario se construye utilizando las clases View y ViewGroup.

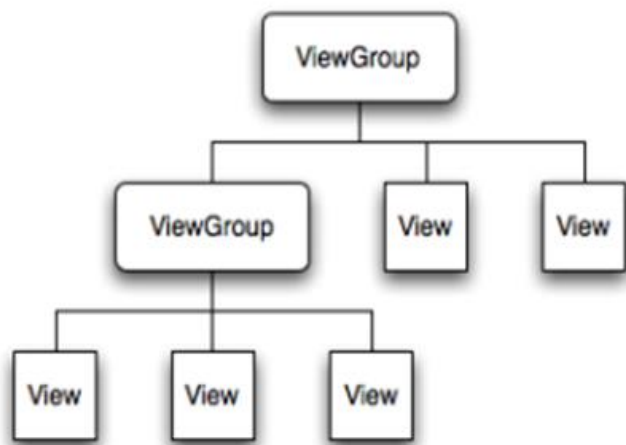


Figura 3. Árbol de nodos View o Viewgroups [19]

La clase View es la unidad básica de la interfaz de usuario en la plataforma Android, mientras que, la clase ViewGroup sirve como base para las subclases llamadas "layouts", que ofrecen distintos tipos de diseño para la interfaz, tales como lineal, relativo, tabular.

Las interfaces de estos sistemas están formadas por dos tipos de elementos: los que otorgan información al usuario y los que obtienen información del usuario. Los elementos que otorgan información al usuario son

aquellos que permiten visualizar, entre otras cosas, el funcionamiento de la aplicación y su configuración; mientras que los elementos que obtienen la información del usuario, son los que permiten al usuario dar entradas al sistema, para realizar los ajustes de personalización a la aplicación.

Un elemento notable, dentro de la interfaz de usuario de los sistemas android, son los llamados "widget". Un Widget es un objeto View que sirve como interfaz para la interacción del usuario con la aplicación [20]. Los "widget" pueden realizar funciones, tanto de proveer información al usuario, como obtener información del mismo.

CONCLUSIÓN

El desarrollo de este documento ha aportado conocimientos sobre varias tecnologías que en la actualidad están directamente relacionadas con el sistema operativo Android, pero sobretodo de cómo desarrollar una aplicación que las unifique a todas y tengan una función práctica. Además, a través de su estudio, se ha podido constatar un mercado emergente que cada día realiza más avances y terminará dotando a las personas de un nivel de comunicación muy elevado; pero sobretodo, de una accesibilidad inmediata. Por lo tanto, con este documento sentimos que estamos contribuyendo de alguna manera en este avance.

El vertiginoso, pero muy estable y seguro desarrollo de Android, ha sido, a nuestro parecer, muy interesante dado el gran abanico de posibilidades que ofrece, siendo imposible recogerlas todas en un solo documento o plasmarlas en una sola aplicación. Aun, a pesar de las múltiples posibilidades que ofrece este sistema operativo y que para esta oportunidad no se están mencionando, este documento es un buen punto de partida para familiarizarse con el entorno de dicho sistema en sí y las características y prestaciones fundamentales que el mismo ofrece.

Se pudo obtener una visión más amplia de los aspectos de seguridad y personalización, que han hecho de Android uno de los sistemas operativos más usados de los últimos años, haciendo competencia directa con sistemas como iOS de Apple o Rim de Blackberry, los que fácilmente doblan el tiempo de existencia en el mercado de la pequeña, pero brillante estrella de Google, Android.

Con el resultado obtenido, después de realizar este trabajo investigativo, se han cumplido las expectativas

iniciales de desarrollar un documento que muestre un vistazo inicial, pero completo, de lo que se espera sea el inicio de una nueva era en el mundo de la comunicación móvil y el desarrollo de aplicaciones que resuelvan desde la necesidad más básica, como hacer una llamada, hasta aplicaciones de monitoreo remoto a través de redes Wan.

REFERENCIAS

- [1] P. F. Salas, "Implementación de una plataforma de telemonitorización de pacientes estándar, open source y ubicua basada en ISO/IEEE 11073-PHD sobre Android," Universidad de Zaragoza, 2010.
- [2] D. Barrera, et al., "A Methodology for Empirical Analysis of Permission-Based Security Models and its Application to Android," 2010.
- [3] B. C. Pujeu, "Technological overview of Android," Escola Politècnica Superior de Castelldefels, Universitat Politècnica de Catalunya, 2008.
- [4] I. A. F. Pacheco, "Diseño y desarrollo de un sistema de video-vigilancia basado en tecnología Android," Escuela Politècnica Superior, Universidad Carlos III de Madrid, 2010.
- [5] G. Inc. (2009). Android Market. Available: www.android.com
- [6] W. Enck, et al., "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," 2010.
- [7] H. M. Calleja and G. M. Postigo, "Aplicación de gestión y seguimiento via GPS sobre dispositivo móvil y plataforma Android," Departament de Ciència dels Materials i Enginyeria Metal·lúrgica, Universitat Politècnica de Catalunya, 2011.
- [8] J. P. Olesti, "Introducción a Android y sus aplicaciones," Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla, 2009.
- [9] J. F. S. Aquino, "Plataformas de Desenvolvimento para dispositivos Móveis," PROGRAMA DE PÓS GRADUAÇÃO EM INFORMÁTICA, PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO – PUC-RIO, 2007.
- [10] D. Ehringer, "The Dalvik Virtual Machine Architecture," 2010.
- [11] s. Conder and L. Darcey, "Android Wireless Application Development," 2011.
- [12] M. F. N. Burguera, "Creación de sistemas android personalizados," Universidad Politècnica de Catalunya, 2011.
- [13] M. Conti, et al., CRePE: Context-Related Policy Enforcement for Android, 2011.
- [14] A. J. S. García, "Desarrollo de una aplicación de cálculo de perfiles radioeléctricos para dispositivos móviles con sistema operativo Android," Escuela Técnica Superior de Ingeniería en Telecomunicaciones, Universidad Politècnica de Cartagena, 2011.
- [15] I. P. Rodríguez, "Estudio de la plataforma de software Android para el desarrollo de una aplicación social," Escuela Politècnica Superior de Castelldefels, Universidad Politècnica de Catalunya, 2009.
- [16] A. P. Felt, et al., "Android Permissions Demystified," 2011.
- [17] F. Ableson, Unlocking Android, 2009.
- [18] A. Developers, "Android Services," ed, 2009.
- [19] M. M. Minguijón, "Java Training: Diseño e implementación de una aplicación de aprendizaje Java a través de un dispositivo móvil Android," Escuela Politècnica Superior, Universidad Carlos III de Madrid, 2010.
- [20] C. S. Guzmán, "Desarrollo de una Aplicación de Podcast sobre Android," Escuela Politècnica Superior, Universidad Carlos III de Madrid, 2011.

Benedik Jaseth Delgado Jaén obtuvo su título de Ingeniero de Sistemas y Computación en la Universidad Tecnología de Panamá, sede de Azuero en 2011. Sus principales líneas de investigación son: Cibermedicina, Telemedicina, Grid Computing y Cloud Computing.

Miguel Vargas-Lombardo obtuvo su título de Ingeniero de Sistemas Computacionales en la Universidad Tecnológica de Panamá en 1998, a partir de ese mismo año fue profesor a tiempo parcial en el área de Arquitectura y Redes de Computadoras. Actualmente es profesor a tiempo completo en la Universidad Tecnológica de Panamá, sede de Coelé, Penonomé, Panamá; además es doctor por la Universidad Politècnica de Madrid, España en el área de Investigación para el Desarrollo de Sistema Software Complejo. Sus principales líneas de investigación son: Cibermedicina, Telemedicina, Grid Computing, y Cloud Computing