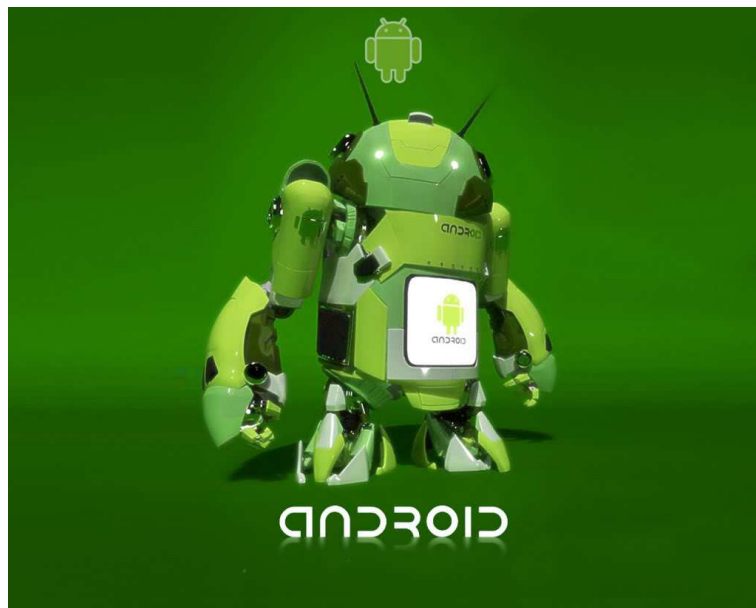


# “Sistema Operativo Android”

Autor: Aylén Mogena Soler

October 4, 2013



## 1 Resumen:

Mediante esta investigación abordaremos los principales aspectos del Sistema Operativo Android el cual está basado en Linux, además esta diseñado principalmente para dispositivos móviles con pantalla táctil, ejemplos de estas son: los teléfonos inteligentes, tabletas, ordenadores portátiles, netbooks , auriculares, reloj de pulsera, entre otros.

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un marco de trabajo (framework) Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

El mismo presenta diversas características como son: diseño de diapositivas, almacenamiento, conectividad, mensajería, navegador web, soporte de java, soporte multimedia, soporte para streaming, soporte para hardware adicional, entorno de desarrollo, google play, multi-táctil, bluetooth, videollamada, multitarea, características basadas en voz y finalmente tethering.

Android es considerado como uno de los modelos de negocio más exitosos, pues su desarrollo estratégico contempla los factores que más se tienen en cuenta dentro de las herramientas y metodologías desarrollados por expertos en negocios. Este sistema operativo se ha convertido en un modelo a seguir por desarrolladores de tendencias y negocios de alto impacto.

Android is considered as one of the most successful business models, for its strategic development includes the factors that are taken into account within the tools and methodologies developed by business experts. This operating system has become a model for developers and senior business trends impact.

## 2 Abstract:

Through this research we will address key aspects of the Android operating system which it is based on Linux, and is designed primarily for touch screen mobile devices, examples of these are: intelligentas phones, tablets, laptops, netbooks, headphones, wrist watch , among others.

The Android system announcement was performed November 5, 2007 along with the creation of the Open Handset Alliance, a consortium of 78 companies for hardware, software and telecommunications dedicated to developing open standards for mobile devices. Google released most of the Android code under the Apache license, a license free and open source.

The structure of the Android operating system consists of applications running on a Java framework for object-oriented applications on the core Java libraries on a Dalvik virtual machine with runtime compilation. The operating system consists of 12 million lines of code including 3 million lines of XML, 2.8 million lines of language C, 2.1 million lines of Java, and 1.75 million lines of C++.

It presents various features such as: slide design, storage, connectivity, messaging, web browser, java support, multimedia support, streaming support, support for additional hardware environment, development environment, google play, multi-touch screen, bluetooth, video calling, multitasking, voice-based features and then tethering.

### 3 Introducción:

Google adquirió Android Inc. en julio del 2005 que no era más que una pequeña compañía de Palo Alto, California fundada en el 2003. Algunos de sus cofundadores se fueron a trabajar a Google como: Andy Rubin (cofundador de Danger), Rich Miner (co-fundador de Wildfire Communications, Inc.), Nick Sears (alguna vez VP en T-Mobile) y Chris White (quien encabezó el diseño y el desarrollo de la interfaz en WebTV). En esos tiempos las funciones de este sistema eran muy poco conocido.

En Google, el equipo liderado por Rubin desarrolló una plataforma para dispositivos móviles basada en el núcleo Linux que fue promocionado a fabricantes de dispositivos y operadores con la promesa de proveer un sistema flexible y actualizable.

Inicialmente fue desarrollado por una firma comprada por Google en el 2005 la Android. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.

Posee una inmensa comunidad de desarrolladores escribiendo aplicaciones para extender las funcionalidades de los dispositivos, actualmente ha alcanzado una cifra de 1.000.000 de aplicaciones de las cuales dos tercios son gratuitas disponibles para la tienda de aplicaciones oficial de Android. Los programas están escritos en el lenguaje de programación Java.

Android ha vivido numerosas actualizaciones desde su liberación inicial. Estas actualizaciones al sistema operativo base típicamente arreglan bugs y agregan nuevas funciones. Generalmente cada actualización del sistema operativo Android es desarrollada bajo un nombre en código de un elemento relacionado con postres en orden alfabético.

La variada aparición de versiones, que en muchos casos no llegan a funcionar correctamente en el hardware para versiones previas, hacen que sea considerado uno de los elementos promotores de la obsolescencia programada. El mismo ha sido criticado en varias ocasiones por la fragmentación que sufren sus terminales al no ser soportado con actualizaciones constantes por los distintos fabricantes, tras un comunicado de Google el cual anunció que los fabricantes de comprometerían a aplicar actualizaciones al menos 16 meses desde la salida al mercado, pero esto nunca se concretó.

#### 3.1 Objetivos:

El presente trabajo tiene como objetivo hacer un resumen acerca del sistema operativo Android para de esta manera dar a conocer a las personas que tienen poco conocimiento acerca de este sistema características principales como son: su surgimiento y evolución, herramientas de desarrollo de esta aplicación, además de sus componentes y arquitectura.

## 4 Desarrollo:

### 4.1 Arquitectura de una aplicación Android.

Para desarrollar aplicaciones en Android es importante conocer cómo está estructurado. El mismo está formado por varias capas que facilitan al desarrollador la creación de aplicaciones, además esta estructura permite acceder a las capas más bajas mediante el uso de librerías, lo cual le permite al desarrollador no tener que programar a bajo nivel las funcionalidades requeridas para que una aplicación haga uso de los componentes de hardware de los teléfonos. Cada una de las capas utiliza elementos de la capa inferior para realizar sus funciones, es por ello que a este tipo de arquitectura se le conoce también como pila.



#### **Aplicaciones:**

En la última capa se incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, las nativas (programadas en C o C++) y las administradas (programadas en Java), las que vienen preinstaladas en el dispositivo y aquellas que el usuario ha instalado.

En esta capa encontramos también la aplicación principal del sistema, porque es la que permite ejecutar otras aplicaciones mediante una lista y mostrando diferentes escritorios donde se pueden colocar accesos directos a aplicaciones o incluso widgets (reproductores), que son también aplicaciones de esta capa.

#### **Framework (Marco de trabajo) de aplicaciones:**

Está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik. Siguiendo el diagrama encontramos:

**Activity Manager (Administrador de actividades):**

Es el encargado de administrar la pila de actividades de nuestra aplicación así como su ciclo de vida.

**Windows Manager (Administrador de ventanas):**

Es la encargada de organizar lo que se mostrará en pantalla.

**Content Provider (Proveedor de contenido):**

Esta librería crea una capa que encapsula los datos que se compartirán entre aplicaciones para tener control sobre cómo se accede a la información.

**Views (Vistas):**

En este caso las vistas son los elementos que nos ayudarán a construir las interfaces de usuario: botones, cuadros de texto, listas y hasta elementos más avanzados como un navegador web o un visor de Google Maps.

**Notification Manager (Administrador de notificaciones):**

Engloba los servicios para notificar al usuario cuando algo requiera su atención mostrando alertas en la barra de estado. Esta biblioteca también permite jugar con sonidos, activar el vibrador o utilizar los LEDs del teléfono en caso de tenerlos.

**Package Manager (Administrador de paquetes):**

Esta biblioteca permite obtener información sobre los paquetes instalados en el dispositivo Android, además de gestionar la instalación de nuevos paquetes. Con paquete nos referimos a la forma en que se distribuyen las aplicaciones Android, estos contienen el archivo .apk, que a su vez incluyen los archivos .dex con todos los recursos y archivos adicionales que necesite la aplicación, para facilitar su descarga e instalación.

**Telephony Manager (Administrador de telefonía):**

Con esta librería podremos realizar llamadas o enviar y recibir SMS/MMS, aunque no permite reemplazar o eliminar la actividad que se muestra cuando una llamada está en curso.

**Location Manager (Administrador de localización):**

Permite determinar la posición geográfica del dispositivo Android mediante GPS o redes disponibles y trabajar con mapas.

**Sensor Manager (Administrador de sensores):**

Permite manipular los elementos de hardware del teléfono como el acelerómetro, giroscopio, sensor de luminosidad, sensor de campo magnético, brújula, sensor de presión, sensor de proximidad, sensor de temperatura, etc.

**Cámara:**

Con esta librería podemos hacer uso de la(s) cámara(s) del dispositivo para tomar fotografías o para grabar vídeo.

**Multimedia:**

Permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.

### **Android Runtime (Entorno de Ejecución):**

El entorno de ejecución de Android no se considera una capa en sí mismo, dado que también está formado por librerías. Aquí encontramos las librerías con la funcionalidades habituales de Java así como otras específicas de Android.

El componente principal del entorno de ejecución de Android es la máquina virtual Dalvik. Las aplicaciones se codifican en Java y son compiladas en un formato específico para que esta máquina virtual las ejecute. La ventaja de esto es que las aplicaciones se compilan una única vez y de esta forma estarán listas para distribuirse con la total garantía de que podrán ejecutarse en cualquier dispositivo Android que disponga de la versión mínima del sistema operativo que requiera la aplicación.

Cabe aclarar que Dalvik es una variación de la máquina virtual de Java, por lo que no es compatible con el bytecode Java. Java se usa únicamente como lenguaje de programación, y los ejecutables que se generan con el SDK de Android tienen la extensión .dex que es específico para Dalvik, y por ello no podemos correr aplicaciones Java en Android ni viceversa.

### **Librerías:**

La siguiente capa que se sitúa justo sobre el kernel (núcleo) la componen las bibliotecas nativas de Android, también llamadas librerías. Están escritas en C o C++ y compiladas para la arquitectura hardware específica del teléfono. Estas normalmente están hechas por el fabricante, quien también se encarga de instalarlas en el dispositivo antes de ponerlo a la venta. El objetivo de las librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan a cabo de la forma “más eficiente”.

Entre las librerías incluidas habitualmente encontramos OpenGL (motor gráfico), Bibliotecas multimedia (formatos de audio, imagen y video), Webkit (navegador), SSL (cifrado de comunicaciones), FreeType (fuentes de texto), SQLite (base de datos), entre otras.

### **Kernel (núcleo) de Linux:**

El núcleo del sistema operativo Android está basado en el kernel (núcleo) de Linux versión 2.6, similar al que puede incluir cualquier distribución de Linux, como Ubuntu, solo que adaptado a las características del hardware en el que se ejecutará Android, es decir, para dispositivos móviles.

El núcleo actúa como una capa de abstracción entre el hardware y el resto de las capas de la arquitectura. El desarrollador no accede directamente a esta capa, sino que debe utilizar las librerías

disponibles en capas superiores. De esta forma también nos evitamos el hecho de quebrarnos la cabeza para conocer las características precisas de cada teléfono. Si necesitamos hacer uso de la cámara, el sistema operativo se encarga de utilizar la que incluya el teléfono, sea cual sea. Para cada elemento de hardware del teléfono existe un controlador (o driver) dentro del kernel (núcleo) que permite utilizarlo desde el software.

El kernel (núcleo) también se encarga de gestionar los diferentes recursos del teléfono (energía, memoria, etc.) y del sistema operativo en sí: procesos, elementos de comunicación (networking), etc.

## 4.2 Componentes de una aplicación Android.

A continuación vamos a hacer una breve descripción de algunos de los elementos más importantes para una aplicación Android como son:

### **Vista (View):**

Las vistas son los elementos que componen la interfaz de usuario de una aplicación. Todas las vistas van a ser objetos descendientes de la clase View (Vista), por lo que, pueden ser definidas utilizando código Java. Sin embargo, lo habitual va a ser definir las vistas utilizando un fichero XML y dejar que el sistema cree los objetos por nosotros a partir de este fichero.

### **Disposición (Layout):**

Es un conjunto de vistas agrupadas de una determinada forma. Vamos a disponer de diferentes tipos de Layouts para organizar las vistas de forma lineal, en cuadrícula o indicando la posición absoluta de cada vista. Los Layouts (las Disposiciones) también son objetos descendientes de la clase View (Vista). Igual que las vistas los Layouts (las Disposiciones) pueden ser definidos en código, aunque la forma habitual de definirlos es utilizando código XML.

### **Actividad (Activity):**

Una aplicación va a estar formada por un conjunto de elementos básicos de visualización, coloquialmente conocidos como pantallas de la aplicación. En Android cada uno de estos elementos, o pantallas, se conoce como actividad. Su función principal es la creación del interfaz de usuario. Una aplicación suelen necesitar varias actividades para crear el interfaz de usuario. Las diferentes actividades creadas serán independientes entre sí, aunque todas trabajarán para un objetivo común. Toda actividad ha de pertenecer a una clase descendiente de Activity.



**Servicio (Service):**

Un servicio es un proceso que se ejecuta “detrás”, sin la necesidad de una interacción con el usuario. Es algo parecido a un demonio en Unix o a un servicio en Windows. En Android disponemos de dos tipos de servicios: servicios locales, que pueden ser utilizados por aplicaciones del mismo terminal y servicios remotos, que pueden ser utilizados desde otros terminales.

**Intención (Intent):**

Una intención representa la voluntad de realizar alguna acción; como realizar una llamada de teléfono, visualizar una página web. Se utiliza cada vez que queramos: lanzar una actividad, lanzar un servicio, lanzar un anuncio de tipo broadcast Comunicarnos con un servicio

Los componentes lanzados pueden ser internos o externos a nuestra aplicación. También utilizaremos las intenciones para el intercambio de información entre estos componentes.

En muchas ocasiones una intención no será inicializada por la aplicación, si no por el sistema, por ejemplo, cuando pedimos visualizar una página web. En otras ocasiones será necesario que la aplicación inicialice su propia intención. Para ello se creará un objeto de la clase Intent.

**Receptor de anuncios (Broadcast receiver):**

Un receptor de anuncios recibe y reacciona ante anuncios de tipo broadcast. Existen muchos originados por el sistema; como por ejemplo Batería baja, llamada entrante,... Aunque, las aplicaciones también puede lanzar un anuncio broadcast. No tienen interfaz de usuario, aunque pueden iniciar una actividad para atender a un anuncio.

**Proveedores de Contenido (Content Provider):**

La compartición de información entre teléfonos móviles resulta un tema vital. Android define un mecanismo estándar para que las aplicaciones puedan compartir datos sin necesidad de comprometer la seguridad del sistema de ficheros. Con este mecanismo podremos acceder a datos de otras aplicaciones, como la lista de contactos, o proporcionar datos a otras aplicaciones.

**4.3 Herramientas de desarrollo.****Android SDK ( Software Development Kit ):**

Incluye un conjunto de herramientas de desarrollo que comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales. Las plataformas de desarrollo soportadas incluyen Linux, Max OS X 10.4.9 o posterior, y Windows XP o posterior. La plataforma integral de desarrollo IDE (Integrated Development Environment) soportada oficialmente es Eclipse junto con el complemento ADT ( Android Development Tools plugin ), aunque también puede utilizarse un editor de texto para escribir ficheros Java y Xml y utilizar comandos en un terminal ( se necesitan los paquetes JDK, Java Development Kit y Apache Ant ) para crear y depurar aplicaciones. Además, pueden controlarse dispositivos Android que estén conectados.

Las Actualizaciones del SDK están coordinadas con el desarrollo general de Android. El SDK soporta también versiones antiguas de Android, por si los programadores necesitan instalar aplicaciones en dispositivos ya obsoletos o más antiguos. Las herramientas de desarrollo son componentes descargables, de modo que una vez instalada la última versión, pueden instalarse versiones anteriores y hacer pruebas de compatibilidad.

Una aplicación Android está compuesta por un conjunto de ficheros empaquetados en formato .apk y guardada en el directorio /data/app del sistema operativo Android ( este directorio necesita permisos de superusuario , root, por razones de seguridad ). Un paquete APK incluye ficheros .dex ( ejecutables Dalvik, un código intermedio compilado ), recursos, etc.

### **Native Development Kit. Desarrollo Nativo:**

El NDK permite instalar bibliotecas escritas en C y otros lenguajes, una vez compiladas para ARM o código x86 nativo. Los programas Java corriendo en la máquina virtual Dalvik ( Dalvik VM ) pueden llamar a clases nativas por medio de la función `System.loadLibrary`, que forma parte de las clases estándar Java en Android.

Se pueden compilar e instalar aplicaciones completas utilizando las herramientas de desarrollo habituales. El depurador ADB proporciona un shell root en el Simulador de Android que permite cargar y ejecutar código nativo ARM o x86. Este código puede compilarse con GCC en un ordenador normal. La ejecución de código nativo es difícil porque Android utiliza una biblioteca de C propia (libc, llamada Bionic). Se accede al dispositivo gráfico como un framebuffer disponible en /dev/graphics/fb0. La biblioteca gráfica que utiliza Android para controlar el acceso a este dispositivo se llama Skia Graphics Library (SGL), disponible con licencia de código abierto. Skia tiene implementaciones en win32 y Unix, permitiendo el desarrollo cruzado de aplicaciones, y es el motor de gráficos que soporta al navegador web Google Chrome.

### **Android Open Accessory Development Kit:**

La plataforma de Android 3.1 ( portado también a Android 2.3.4 ) introduce soporte para Android Open Accessory, que permite interactuar a dispositivos USB externos ( accesorios USB Android ) interactuar con el dispositivo en un modo especial llamado "accessory". Cuando un dispositivo Android está en modo "accessory" el dispositivo externo actúa como hub usb ( proporciona alimentación y enumera los dispositivos ) y el dispositivo Android actúa como dispositivo USB. Los accesorios Android USB están diseñados específicamente para conectarse a dispositivos Android y utilizan un protocolo simple ( Android accessory protocol ) que les permite detectar dispositivos Android que soportan modo "accessory".

### **App Inventor para Android:**

Google anunció en julio de 2010 la disponibilidad de App Inventor para Android, que es un entorno de desarrollo visual Web, para programadores noveles, basado en la biblioteca Open Blocks Java, del MIT. Este entorno proporciona acceso a funciones GPS, acelerómetro y datos de orientación, funciones de teléfono, mensajes de texto, conversión habla a texto, datos de contacto, almacenamiento permanente, y servicios Web, incluyendo inicialmente Amazon y Twitter. Hal Abelson, director de proyecto en el MIT, dijo: "Sólo hemos podido hacerlo porque la arquitectura Android es tan abierta".

### **HyperNext Android Creator (HAC):**

Es un sistema de desarrollo de programas dirigido a programadores que empiezan, permitiéndoles crear sus propias aplicaciones sin necesitar conocimientos de Java y del SDK de Android. Está basado en HyperCard, que gestiona el software como una pila de tarjetas en la que sólo una de ellas es visible en un momento dado y por tanto encaja bien en aplicaciones para teléfonos móviles, con una sola ventana disponible a la vez. El lenguaje principal de desarrollo se llama simplemente HyperNext y está relacionado con el lenguaje de HyperCards HyperTalk. HyperNext es un intérprete de un lenguaje similar al Inglés y tiene muchas funciones para crear aplicaciones Android. Soporta un subconjunto creciente del SDK de Android incluyendo sus propias versiones de controles gráficos de interfaz de usuario ( GUIs ) y ejecuta automáticamente su propio servicio, de forma que las aplicaciones pueden continuar ejecutándose y procesando información, sin estar en el frontal del usuario.

### **El proyecto Simple:**

El objetivo de Simple es ser un lenguaje fácil de aprender para la plataforma Android. Simple es un dialecto de BASIC para Android. Sirve tanto para programadores profesionales como aficionados permitiendo escribir rápidamente aplicaciones que utilizan los componentes de Android.

Parecido a Visual Basic 6 de Microsoft, los programas Simple consisten en definiciones de formularios ( que contienen componentes ) y código ( con la lógica del programa ). La interacción entre ellos se hace por medio de eventos lanzados por los componentes. La lógica del programa consiste en gestores de eventos, que ejecutan código dependiendo del evento.

El proyecto Simple no tiene mucha actividad. La última actualización de código se realizó en agosto de 2009. Existe un producto comercial parecido llamado Basic4android, inspirado en Visual Basic 6 y Microsoft Visual Studio. Este proyecto si tiene actividad y hay una comunidad sólida de programadores.

## **5 Conclusiones:**

Como hemos podido conocer Android nos proporciona un entorno sumamente poderoso para que podamos programar aplicaciones que hagan cualquier tipo de funcionalidad. Nada dentro de Android es inaccesible y podemos jugar siempre con las aplicaciones de nuestro teléfono para optimizar cualquier tarea.

El potencial de Android se sitúa en el control total que se le da al usuario para que haga de su teléfono un dispositivo a su medida.

## **6 Bibliografía:**

### **References**

- [1] Android Ayuda: Portal de Android con Apps, tutoriales y noticias. [online]. [Accessed 4 October 2013]. Available from: <http://androidayuda.com/>
- [2] Arquitectura de Android | Androideity. [online]. [Accessed 4 October 2013]. Available from: <http://androideity.com/2011/07/04/arquitectura-de-android/>
- [3] Componentes de una aplicación - Android: Programación Aplicaciones para móviles. [online]. [Accessed 3 October 2013]. Available from: <http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/149-componentes-de-una-aplicacion>

- [4] El androide libre: Aplicaciones Android, juegos, noticias sobre el SO móvil de Google, Nexus, Samsung, root y mucho más. [online]. [Accessed 4 October 2013]. Available from: <http://www.elandroidelibre.com/>
- [5] Historia de Android: La Evolución a lo largo de sus versiones - Android Zone. [online]. [Accessed 4 October 2013]. Available from: <http://androidzone.org/2013/05/historia-de-android-la-evolucion-a-lo-largo-de-sus-versiones/>
- [6] Introducción a las Herramientas de Desarrollo para Android. [online]. [Accessed 4 October 2013]. Available from: <http://www.slideshare.net/kix2902/introduccion-a-las-herramientas-de-desarrollo-para-android-16183107>
- [7] Los componentes de toda aplicación Android. | Android Chichayo. [online]. [Accessed 4 October 2013]. Available from: <http://android.cix.pe/lecciones/los-componentes-de-toda-aplicacion-android/>