

Android ?....De qué me hablan

Android ?.... Are you talking about*

Carlos Alberto Vanegas**

Fecha de recepción: junio de 2013

Fecha de aprobación: agosto de 2013

Resumen

Actualmente el uso de dispositivos móviles se ha incrementado, especialmente los teléfonos inteligentes a los cuales se les han proporcionado funciones como pantallas de alta resolución, mayor potencia de procesamiento, mayor capacidad de almacenamiento, navegación por Internet, reproducción de multimedia, realizar operaciones bancarias, GPS, conexión a redes sociales, Wi-fi, entre otras. Estos dispositivos corren bajo un sistema operativo determinado que hace posible administrar las diferentes aplicaciones con capacidades avanzadas. Android es un sistema operativo abierto para dispositivos móviles adquirido por Google y por el grupo *Open Handset Alliance* que representa una alternativa tecnológica capaz de administrar todos los recursos de manera eficiente y sin interrupciones, de tal manera que el usuario puede mantener una comunicación estable haciendo uso de los recursos que el hardware le suministra. En este artículo se analiza el sistema operativo Android, su arquitectura, las versiones existentes, su máquina virtual Dalvik, así como el desarrollo de un sencillo ejemplo de una aplicación Android. of traditional cell phones to great deal.

Palabras claves: aplicación, código abierto, dispositivo, programación, sistema operativo.

* Artículo de investigación

** Ingeniero de Sistemas, Universidad Incca de Colombia; Especialista en Ingeniería de Software, Universidad Distrital Francisco José de Caldas; Magíster en Ingeniería de Sistemas, Universidad Nacional de Colombia. Docente investigador del grupo CompuParalela adscrito a la Facultad Tecnológica de la Universidad Distrital Francisco José de Caldas. Contacto: cavanegas@udistrital.edu.co.

Abstract

Today the use of mobile devices has increased, especially those called smartphones which have been provided with useful features like high resolution screens, more processing power, more storage, internet browsing, multimedia playback, banking operations, GPS, download games, social networking, Wi-Fi, Bluetooth, watch TV, listen to radio, etc.. These devices are under a particular operating system that makes it possible to manage different applications with advanced capabilities. Android is an open operating system for mobile devices purchased by Google and Open Handset Alliance group that represents a technological alternative capable of managing all resources efficiently and without interruption, so that the user can maintain stable communication using resources that the hardware supplied. This article seeks to know a little about the Android operating system, its architecture, the existing versions, the Dalvik virtual machine, as well as the development of a simple example of an Android application.

Keywords: application, device, open source, operating system, programming.

1. Introducción

Android es un sistema operativo basado en *GNU/Linux* de código abierto bajo licencia Apache, el cual permite la creación principalmente de aplicaciones para dispositivos móviles –teléfonos inteligentes, tablets, reproductores MP3, notebook, y otros– desarrollado por Google [1] y actualmente liderado por el grupo *Open Handset Alliance* [2], en el cual se agrupan varias compañías del sector, entre las cuales se encuentran: Google, Samsung, HTC, Dell, Intel, Qualcomm, Motorola, LG, Telefónica, T-Mobile, Nvidia. Android permite desarrollar aplicaciones usando código Java y C –se compila en código nativo *ARM*, familia de microprocesadores RISC diseñados por la empresa ARM Holdings– [3]. Las características de la plataforma Android son [4]:

- Framework de aplicaciones: permite la sustitución y reutilización de los componentes.
- Máquina virtual Dalvik: software que ejecuta las aplicaciones en los teléfonos móviles.
- Navegador inteligente: basado en el navegador *WebKit* –plataforma para aplicaciones que funciona como base para diferentes navegadores –.
- Optimizador de gráficas: librerías de graficas 2D, 3D sobre especificaciones OpenGL 1.0.
- SQLite: base de datos para almacenamiento de datos estructurados.
- Multimedia: soporta los formatos comunes de audio, video e imágenes MPEG4, MP3, JPG, GIF, JPEG.
- Telefonía GSM: dependiente del hardware.
- Bluetooth, EDGE, 3G y Wifi: dependiente del hardware.

- Soporte de hardware adicional: cámaras, GPS, brújulas, acelerómetros, pantallas táctiles.
- Entorno de desarrollo: emuladores, herramientas para depuración, memorias y perfiles de rendimiento, conector para el entorno de desarrollo basado en Java Eclipse.

2. Arquitectura Android

La arquitectura de Android [4], [5] está formada por cuatro *capas* o *niveles* que le permiten al programador la creación de aplicaciones. Su distribución ayuda a acceder a las diferentes capas por medio de librerías y cada capa utiliza los elementos de la capa inferior para realizar sus funciones, por eso, su arquitectura es tipo *pila*. La arquitectura del sistema operativo Android se puede apreciar en la siguiente gráfica:

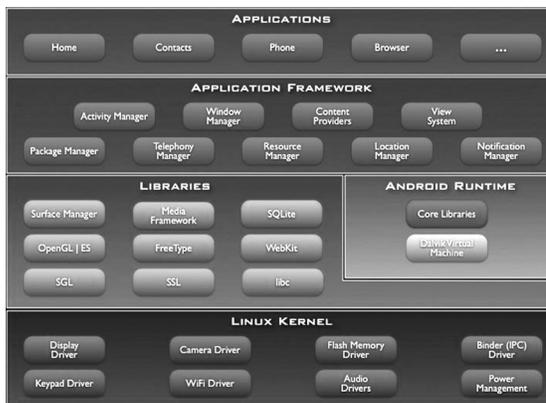


Figura 1. Arquitectura Android

Fuente: disponible en [4].

2.1 Capa Linux kernel

El núcleo de Android está basado en la versión 2.6, adaptado a las características de hardware para dispositivos móviles. El kernel es una capa de abstracción del hardware (*hardware abstraction layer*) que permite que las aplicaciones accedan a través de drivers (controladores), y también asume la admi-

nistración de los recursos del teléfono y del sistema operativo. En el núcleo se encuentran los siguientes elementos:

- Display driver: controlador de pantalla.
- Keypad driver: controlador de teclado.
- Camera driver: controlador de cámara.
- Wifi driver: controlador de conexión inalámbrica.
- Flash memory driver: controlador de memoria rápida.
- Audio drivers: controladores de audio.
- Binder (IPC) driver: controlador de cobertura.
- Power management: administrador de energía.

2.2 Capa librerías (libraries) [6]

Son bibliotecas nativas de Android escritas en C/ C++ y son las encargadas de realizar la comunicación entre la capa de abstracción de hardware con las API (*Application Programming Interface* - Interfaz de Programación de Aplicaciones) y las aplicaciones. Las principales librerías son:

- Surface manager: encargado de la gestión de las ventanas gráficas.
- OpenGL/ES: librerías para soportar gráficas 3D.
- SGL: librerías gráfica 2D.
- Media framework: bibliotecas para manejo de multimedia.
- Freetype: permite manejar diferentes fuentes de mapas de bits.
- Ssl: capa de seguridad Android.
- SQLite: base de datos relacional.
- Webkit: plataforma de aplicaciones para navegadores.
- Libc: librerías de C.

2.3 Entorno de ejecución Android (Android runtime)

Aunque no se considera una capa, también está formado por librerías. Las principales librerías son:

- Máquina virtual Dalvik (*Android SDK*): es el componente principal y para su funcionalidad requiere poca memoria, se diseñó para ejecutar simultáneamente varias instancias de la máquina virtual. Aunque el Android SDK es una variación del SDK de Java y sus aplicaciones se codifican con Java, no son compatibles con el bytecode (código independiente del hardware generado por el compilador Java y ejecutado por su intérprete) de Java, porque al compilarse las aplicaciones Android los ejecutables que se generan son de extensión (.dex).
- Núcleo de librerías (*Core libraries*): contiene más clases básicas para el manejo del sistema.

2.4 Capa marco de trabajo de aplicaciones (application framework)

Está conformada por clases y servicios que sirven para obtener las funciones básicas de los móviles y programar aplicaciones. Sus componentes permiten acceder a los recursos de las capas *Libraries* y *Linux Kernel*. Se encuentran los siguientes elementos:

- Activity manager: se encarga de administrar todas las actividades y el ciclo de vida de una aplicación.
- Windows manager: se encarga de organizar lo que se visualizará en pantalla.
- Content provider: encapsula los datos que se compartirán entre las aplicaciones.
- Views: son elementos que permiten la construcción de interfaces de usuario, como cuadros de texto, botones, listas, entre otros.
- Notification manager: notifica al usuario cuando se requiere de su atención.
- Package manager: permite obtener información de los paquetes instalados en el dispositivo e instalar nuevos paquetes.
- Telephony manager: permite realizar llamadas o enviar y recibir SMS (servicio de

mensaje corto)/MMS (servicio de mensajería multimedia).

- Resource manager: permite gestionar todos los elementos pertenecientes a una aplicación.
- Location manager: permite establecer mediante GPS o redes la posición geográfica de un dispositivo Android.

2.5 Capa aplicaciones (applications)

Es la capa de aplicaciones del dispositivo, donde se encuentra la interfaz de usuario, las aplicaciones programadas con Java, las aplicaciones preinstaladas y las instaladas por el usuario. En ella se encuentran los siguientes elementos:

- Home: inicio.
- Contacts: contactos.
- Browse: navegador.
- Phone: teléfono.
- Otras aplicaciones.

3. Versiones Android

El sistema operativo Android [7], [8] ha evolucionado vertiginosamente desde su primera versión 1.0 hasta la actual 4.0. Generalmente en cada actualización se solucionan errores de versiones anteriores y se le agregan nuevas funciones. Por otro lado, a cada versión de Android se le denomina con un nombre en código de un puesto en orden alfabético y un nivel de API [9], el cual es un número entero que identifica la nueva versión. El nuevo nivel API es compatible con versiones anteriores, es decir, se agregan nuevas funcionalidades, se actualizan las API más obsoletas, pero no se quitan, y en raras ocasiones estas se modifican o se eliminan. A continuación se relacionan las diferentes versiones de Android:

- Versión 1.0 (Nivel API 1): primera versión liberada el 23 de septiembre de 2008.
- Versión 1.1 (Nivel API 2): versión liberada el 9 de febrero de 2009.

- Versión 1.5 (Nivel API 3, Cupcake): versión liberada en abril de 2009, se destacan entre otras las siguientes características:
 - Captura y reproducción de videos.
 - Permite subir videos de Youtube en forma directa.
 - Teclado con predicción de texto.
 - Nuevos widgets (reproductores).
 - Pantallas animadas.
- Versión 1.6 (Nivel API 4, Donut): liberada en septiembre de 2009, se destacan entre otras las siguientes características:
 - Navegación paso a paso por Google Map.
 - Mejoras en la interfaz de la cámara y el video.
 - Búsquedas por voz.
 - Soporte para pantallas con resoluciones WVGA.
- Versión 2.0/2.1 (Nivel API 5/7, Eclair): liberada entre diciembre de 2009 y enero 2010, se destacan entre otras las siguientes características:
 - Zoom digital de la cámara.
 - Navegador con soporte de video y flash.
 - Interfaz de usuario en el navegador que soporta HTML5.
 - Bluetooth 2.1.
 - Soporte nativo de facebook.
- Versión 2.2 (Nivel API 8, Froyo): liberada en mayo de 2010, se destacan entre otras las siguientes características:
 - Optimización de la memoria.
 - Funcionalidad Wifi, hotspot (puntos de acceso) y tethering (inmovilización) por USB.
 - Soporta correo electrónico de Microsoft Exchange.
 - Nuevos widgets (reproductores).
 - Soporta contraseñas numéricas y alfanuméricas.
 - Cambio entre múltiples idiomas de teclado y sus diccionarios.
- Versión 2.3 (Nivel API 9, Gingerbread): liberada en diciembre de 2010, se destacan entre otras las siguientes características:
 - Soporte telefonía VoIP (voz sobre IP), SIP (protocolo de iniciación de sesiones).
 - Reproducción de videos WebM/VP8 (gestión de correo electrónico desde una Web).
 - Teclado multitáctil rediseñado.
 - Funcionalidad de copiar, cortar y pegar.
 - Soporte para más sensores (giroscopios, barómetros).
 - Administración de la energía mejorada.
- Versión 3.0/3.1/3.2 (Nivel API 11/12/13, Honeycomb): liberada en mayo de 2011, se destacan entre otras las siguientes características:
 - Sistema operativo optimizado para tablets.
 - Escritorio 3D con widgets rediseñados.
 - Funciones de multitarea.
 - Nuevos widgets (reproductores).
 - Soporte videochat mediante Google Talk.
 - Soporte para diversos periféricos y accesorios con conexión a USB.
 - Sincronización de favoritos con Google Chrome.
 - Autorrelleno de formularios.
- Versión 4.0 (Nivel API 14, ice cream sandwich): liberada en octubre de 2011, se destacan entre otras las siguientes características:
 - Diseñado para smartphones, tablets, notebooks.
 - Opción de utilizar botones virtuales en la interfaz de usuario.
 - Corrector de texto rediseñado.
 - Widgets redimensionables.
 - Reconocimiento facial.
 - Características para compartir contenido entre teléfonos (Android Beam).

- Soporte para uso de lápiz táctil (Stylus).

4. Máquina virtual Dalvik (Software Development Kit Android-SDK Android)

El SDK Android es una máquina virtual llamada *Dalvik* [10],[11],[12] que proporciona herramientas y API para desarrollar aplicaciones en la plataforma Android. Está basada en registros y ejecuta clases compiladas por intermedio del compilador de Java que posteriormente es convertida a formato nativo Dalvik utilizando la herramienta *dx*. En este kit de desarrollo se encuentran herramientas para entornos de desarrollo y depuración, conjuntos de bibliotecas o librerías, emuladores de teléfonos móviles, documentación, ejemplos, conexión a Eclipse IDE para Java.

El SDK Android funciona con los sistemas operativos *Windows Xp* o superior, *Mac OS X 10.4.8* o superior, *Ubuntu 6.06* o superior y en otras distribuciones de *Linux* (Open Handset Alliance solo da soporte a Ubuntu), necesita *Python 2.2* y *Java 1.5* o superior y se puede integrar al entorno de desarrollo de *Eclipse IDE para Java*. Como su arquitectura se basa en el *kernel Linux* es posible delegar las tareas relacionadas con la gestión de hilos y memoria de bajo nivel.

El SDK ejecuta los archivos en formato *Dalvik Executable (*.dex)*, el cual es un formato optimizado para un almacenamiento eficiente. De esta manera es posible que el código sea compilado a un bytecode independiente de la máquina en la que se va a ejecutar, por lo que la máquina virtual interpreta dicho bytecode cuando se ejecuta una aplicación. El uso de SDK Android permite la reducción del tamaño de los programas porque busca

toda la información duplicada en las diferentes clases y las reutiliza, esto ayuda a liberar el espacio en memoria de objetos que no se estén utilizando y mantener siempre libre la máxima memoria posible.

5. Crear una aplicación Android utilizando Eclipse IDE para Java

Aunque para crear aplicaciones Android se pueden utilizar varios programas autónomos o Web como, *App Inventor* (disponible en <http://www.appinventorbeta.com/learn/setup/>), *iBuildApp* (disponible en <http://ibuildapp.com/>), *Feed.nu* (disponible en <http://feed.nu/>) y *Apps GevSer* (disponible en <http://www.appsgeyser.com/>). En este apartado se presentará la creación de una aplicación Android en Windows utilizando Eclipse IDE para Java, el cual es recomendado por Android para crear *Apps* (aplicaciones). Para esto es necesario tener en cuenta lo siguiente:

- Tener instalado el Java Development Kit # (JDK#) en la máquina de trabajo. Este kit es necesario crear programas en Java (es recomendable la versión 5 o superior), se puede descargar de la página Web: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
- Descargar e instalar el SDK Android para la plataforma Windows desde la página oficial <http://developer.android.com/sdk/index.html>, en los link *android-sdk r16-windows.zip* o *installer r16-windows.exe* (recomendable).
- Descargar el archivo .zip Eclipse IDE para Java Developers desde la página Web: <http://www.eclipse.org/downloads/>. Se deberá descomprimir el archivo en la carpeta deseada y dar clic sobre el archivo *eclipse.exe* para ingresar al entorno de desarrollo de Eclipse.

- Instalar el plug-in (conector) de Android para Eclipse. Para realizar este paso se puede apoyar con los tutoriales que se encuentran en las páginas: <http://instartius.com/blog/?p=289> y <http://www.weterede.com/2009/06/instalacion-sdk-de-android-en-eclipse/>.

Si se tiene instalado el software mencionado anteriormente, el siguiente paso para desarrollar una aplicación Android que visualice el mensaje *Mi primer programa con Android*, desde el entorno de Eclipse IDE, se deberá crear un nuevo proyecto llamado *PrimerProgramaAndroid*. Para esto se deben realizar los siguientes pasos:

- a) Dentro del entorno de Eclipse seleccione *File > New > Project*, como se observa en la figura:

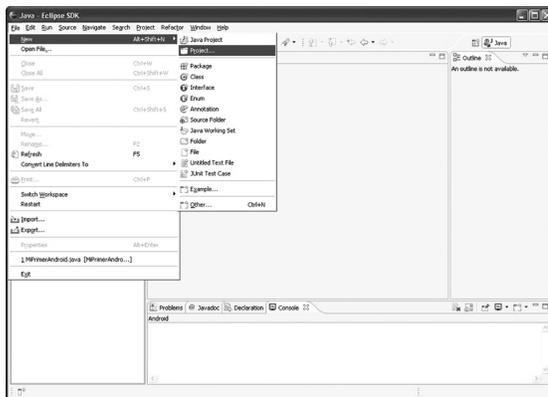


Figura 2. Entorno de desarrollo Eclipse

Fuente: elaboración propia.

Al pulsar la opción *Project*, se obtendrá la siguiente figura:

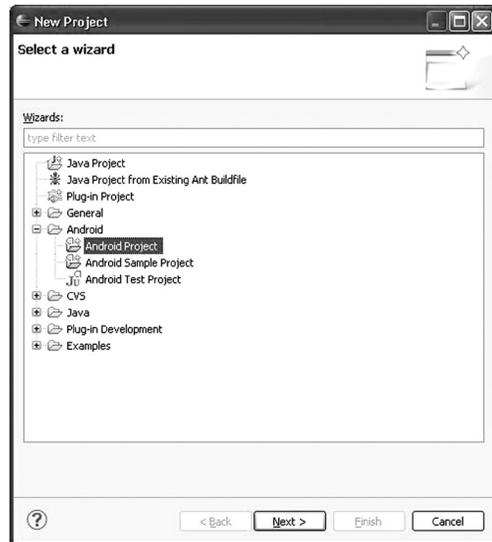


Figura 3. Asistente para crear un proyecto Android con Eclipse

Fuente: elaboración propia.

- b) Seleccionar el icono *Android Project* y pulsar el boton *Next*, para visualizar la siguiente pantalla:

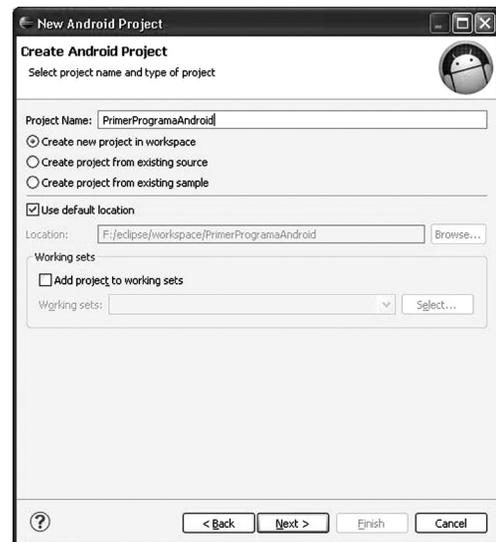


Figura 4. Creacion de un nuevo proyecto Android

Fuente: elaboración propia

- c) En el campo de texto al frente de *Project Name* escribir el nombre del nuevo proyecto, en este caso el proyecto se llamará *PrimerProgramaAndroid* y nuevamente

pulse el botón *Next*, para obtener la siguiente figura:

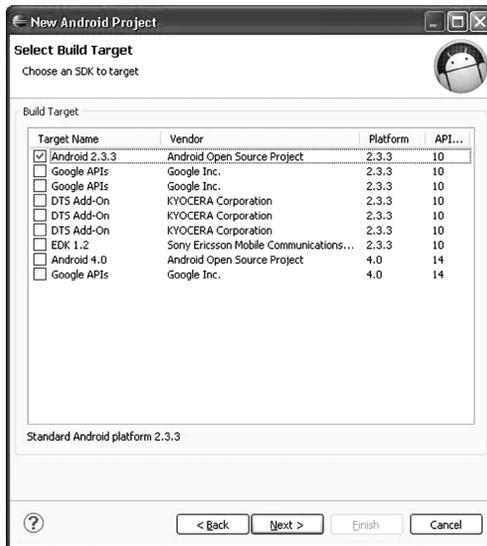


Figura 5. Selección de la plataforma Android

Fuente: elaboración propia.

d) Seleccione la version SDK de Android que previamente halla instalado, para el ejemplo se seleccionó la plataforma *Android 2.3.3*. Pulse el boton *Next* para mostrar la siguiente pantalla:

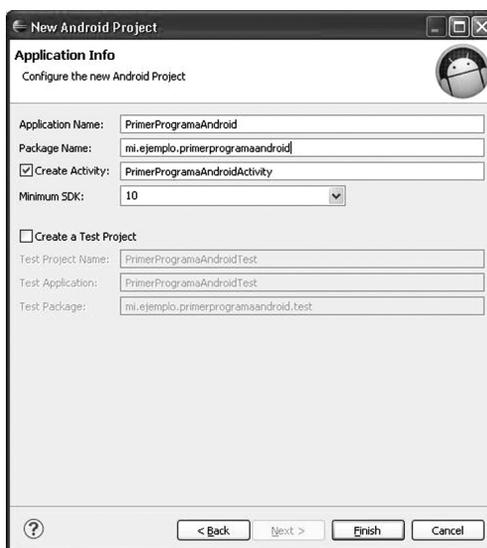


Figura 6. Información del proyecto

Fuente: elaboración propia.

e) En esta ventana se puede modificar todo lo concerniente a la información de la aplicación:

- **Application Name:** es el nombre de la aplicación que aparecerá en el dispositivo Android, tanto en la barra superior durante la ejecución, como en el icono que se instalará en el menú de programas.
- **Package Name:** será el nombre del paquete del espacio de nombres donde se almacenará la aplicación generada.
- **Create Activity:** es el nombre de la clase que será generada. Esta será una subclase de *Activity Android*, la cual es una clase que se puede ejecutar y modificar.
- **Minimum SDK:** se especifica el nivel del API Android que requiere la aplicación.

Al pulsar el boton *Finish*, se obtendrá la siguiente figura:

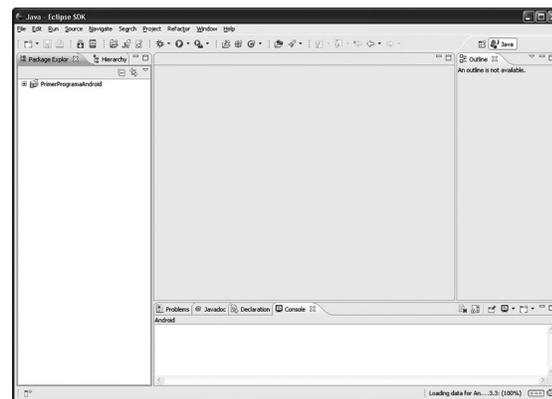


Figura 7. Proyecto Android finalizado

Fuente: elaboración propia.

Si se pulsa sobre el signo **+** al lado del nombre de la aplicación, se observarán los siguientes elementos del proyecto Android: el descriptor de la aplicación (*AndroidManifest.xml*), el código fuente y una serie de archivos con recursos.

f) Pulse sobre la carpeta *src*, luego sobre *mi.ejemplo.primerprogramaandroid* y por último sobre *PrimerProgramaAndroidActivity*, allí dé doble clic sobre el programa fuente *PrimerProgramaAndroidActivity* para obtener el código fuente generado, como se observa en la siguiente figura:

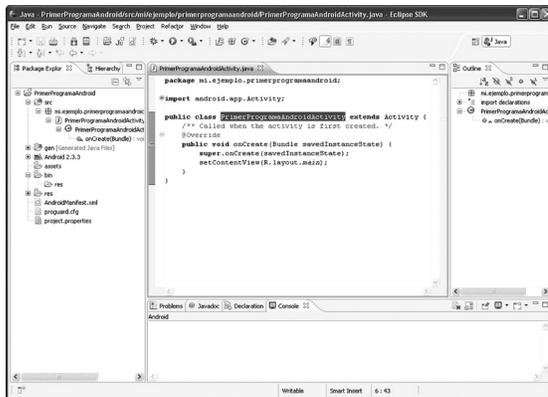


Figura 8. Código fuente del proyecto Android

Fuente: elaboración propia.

g) Modificar el código de la aplicación para mostrar en el dispositivo el mensaje deseado. Para ello se debe realizar lo siguiente:

- Colocar en comentario la instrucción *setContentView(R.layout.main)*.
- Agregar las siguientes instrucciones:
 - *import android.widget.TextView;* (se importa este paquete para poder utilizar el objeto *TextView*).
 - *TextView mitexto=new TextView(this);* (se crea un objeto llamado *mitexto* al cual se le asigna espacio de memoria de tipo *TextView*).
 - *mitexto.setText("Mi primer programa con Android");* (se le asigna al objeto *mitexto* por intermedio de la función *setText* un mensaje).
 - *setContextView(mitexto);* (utilizando la función *setContentView* se visualiza el contenido del objeto *mitexto* en el dispositivo).

Al finalizar se debería observar el código fuente como se aprecia en la siguiente figura:

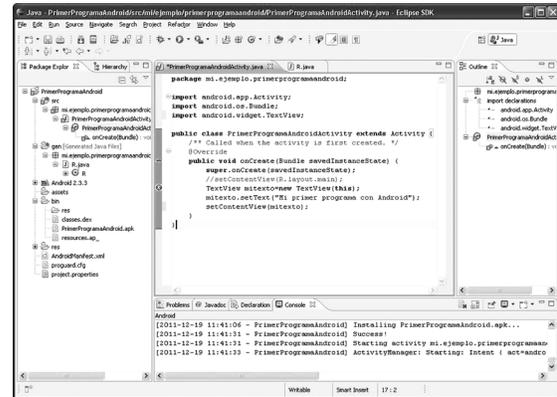


Figura 9. Código fuente del proyecto Android modificado

Fuente: elaboración propia.

h) Seleccione la opción *Run > Run* (Ctrl F11) o simplemente pulse el icono del entorno de Eclipse para ver la siguiente ventana:



Figura 10. Ventana para ejecutar la aplicación Android

Fuente: elaboración propia.

i) Seleccione la opción *Android Application* y pulse el botón *OK*, para visualizar el

dispositivo Android (la inicialización del emulador es algo lenta).

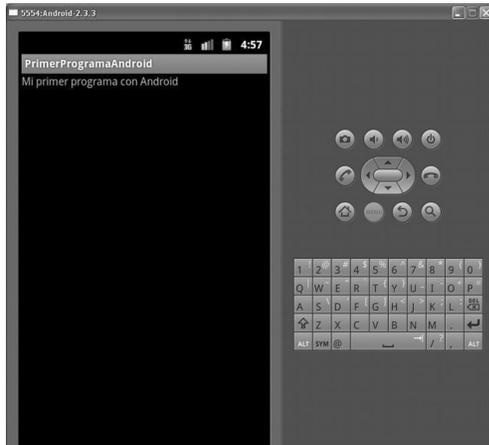


Figura 11. Aplicación Android ejecutada

Fuente: elaboración propia.

6. Conclusiones

El sistema operativo Android es de código abierto tanto para fabricantes de dispositivos móviles como para desarrolladores de aplicaciones.

Android permite administrar los recursos eficientemente y sin interrupciones. También permite desarrollar aplicaciones utilizando código Java y C.

La arquitectura de Android está formada por cuatro niveles o capas, a saber, *Linux kernel*, *libraries*, *application framework*, *applications*.

En la máquina virtual Dalvik de Android se encuentran herramientas para entorno de desarrollo y depuración, librerías, emuladores de teléfonos móviles, documentación, ejemplos y conexión a Eclipse IDE for Java.

Las aplicaciones Android reducen el tamaño de los programas, lo que permite liberar espacio de memoria.

Referencias

- [1] Google Projects for Android [en línea]. Disponible en <http://code.google.com/intl/es/android/>.
- [2] Open Handset Alliance [en línea]. Disponible en http://www.openhandsetalliance.com/android_overview.html.
- [3] Software Android [en línea]. Disponible en <http://upcommons.upc.edu/pfc/bitstream/2009.1/6752/1/memoria.pdf>.
- [4] What is Android [en línea]. Disponible en <http://developer.android.com/guide/basics/what-is-android.html>.
- [5] Arquitectura de Android [en línea]. Disponible <http://androideity.com/2011/07/04/arquitectura-de-android/>.
- [6] Software de comunicaciones - Arquitectura Android [en línea]. Disponible en <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>.
- [7] SDK updates [en línea]. Disponible en <http://android-developers.blogspot.com/search/label/SDK%20updates>.
- [8] Versiones Android [en línea]. Disponible en <http://es.wikipedia.org/wiki/Android>.
- [9] Api,s Level [en línea]. Disponible <http://developer.android.com/guide/appendix/api-levels.html>.
- [10] Android SDK, ya [en línea]. Disponible en <http://www.visualbeta.es/647/movil/android-sdk-ya-disponible/>.
- [11] Open Handset Alliance Releases Android SDK [en línea]. Disponible en http://www.openhandsetalliance.com/press_111207.html.
- [12] *La máquina virtual Dalvik* [en línea]. Disponible en <http://gdroid.com.mx/blog/2011/06/12/la-maquina-virtual-dalvik/>.