

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA ELÉCTRICA



Aplicación de comunicación móvil mediante teléfonos celulares a través de Wifi

PRESENTADO POR:

MARIO ERNESTO LÓPEZ LARA

PARA OPTAR AL TITULO DE:

INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, NOVIEMBRE 2011

UNIVERSIDAD DE EL SALVADOR

RECTOR :

ING. MARIO ROBERTO NIETO LOVO

SECRETARIA GENERAL :

DRA. ANA LETICIA ZAVALETA DE AMAYA

FACULTAD DE INGENIERIA Y ARQUITECTURA

DECANO :

ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL

SECRETARIO :

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERIA ELÉCTRICA

**DIRECTOR
(INTERINO) :**

ING. JOSÉ WILBER CALDERÓN URRUTIA

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO ELECTRICISTA

Título :

**Aplicación de comunicación móvil mediante
teléfonos celulares a través de Wifi**

Presentado por :

MARIO ERNESTO LÓPEZ LARA

Trabajo de Graduación Aprobado por:

Docente Director :

Dr. CARLOS EUGENIO MARTÍNEZ CRUZ

San Salvador, noviembre 2011

Trabajo de Graduación Aprobado por:

Docente Director :

DR. CARLOS EUGENIO MARTÍNEZ CRUZ

DEDICATORIA.

El presente trabajo de graduación se lo dedico a Dios y a la Virgencita María, quienes me han permitido alcanzar éste logro en mi vida.

También se lo dedico a mi familia, especialmente a mi madre María Yolanda quien durante toda mi vida como estudiante ha sido mi inspiración para no darme por vencido y lograr todo lo que me he propuesto. Se lo dedico a mi papá José Miguel y a mi hermano Rogelio que también han sido un enorme apoyo para llegar hasta ésta instancia. A mis abuelos, tíos, primos y los demás miembros de mi familia que me brindan su cariño.

A Ana Ruth, persona muy importante en mi vida que con sus palabras y apoyo incondicional, merece reconocimiento en este logro que he obtenido.

Para finalizar a todas aquellas personas que han estado a mí alrededor durante mi formación académica: docentes, compañeros y amigos.

MARIO LÓPEZ.

AGRADECIMIENTOS.

Doy gracias a Dios por brindarme la bendición de alcanzar un sueño más y por darme la sabiduría necesaria para llegar hasta esta etapa.

Gracias a mi madre María Yolanda, excelente mamá, que me ha brindado todo lo que he necesitado durante el transcurso de mi carrera. Quien con sus consejos siempre me ha mostrado el camino correcto. A mi papá José Miguel que me ha brindado su ayuda en momentos muy críticos. Primero Dios muy pronto podré recompensar todos sus esfuerzos.

Gracias a mi hermano Rogelio López, por su apoyo y ayuda en momentos importantes, por ser mi compañero durante toda mi vida. Estoy orgulloso de él y sé que alcanzará todas las metas que se proponga.

Gracias a mi amiga y mejor compañera Ana Ruth, por todos los momentos que estuvo a mi lado estudiando, escuchándome y aconsejándome. Por todo el apoyo que recibí de su parte para todo el desarrollo del presente trabajo de graduación. Primero Dios alcancemos las metas que nos propongamos y podamos seguir cosechando frutos como profesionales.

Gracias al Dr. Carlos Martínez, por su apoyo y guiarme durante el desarrollo de éste trabajo de graduación. Persona a la cual le tengo un gran estima y admiración por ser un excelente docente, una buena persona y un gran amigo.

Gracias a Carlos Franconi por su ayuda con el préstamo de un dispositivo móvil, ya que me facilito continuar con el desarrollo de éste trabajo.

Gracias a mis amigos y compañeros con los que compartimos bastantes momentos de estudios y muchas alegrías. Gracias porque con pequeñas acciones y consejos han colaborado para que pueda lograr este objetivo.

MARIO LÓPEZ.

ACRONIMOS

API:	Application Programming Interface.
IPL:	Initial Program Loader.
SPL:	Second Program Loader.
RAM:	Random Access Memory.
SMS:	Short Message Service.
ROM:	Read Only Memory.
GPS:	Global Positioning System.
SD:	Secure Digital.
MTD:	Memory Technology Device.
FC:	Force Close.
ANR:	Application Not Responding.
JIT:	Just In Time.
OTA:	Over The Air.
GPRS:	General Packet Radio Service.
EDGE:	Enhanced Data rates for GSM of Evolution.
GSM:	Group Special Mobile.
UMTS:	Universal Mobile Telecommunications System.
HSDPA:	High Speed Downlink Packet Access.
WIFI:	Wireless Fidelity.
IDE:	Integrated Development Environment.
JVM:	Java Virtual Machine.
JRE:	Java Runtime Environment.
JDK:	Java Developer Kit.
SDK:	Software Development Kit.
AVD:	Android Virtual Device.
USB:	Universal Serial Bus.
ADT:	Android Development Tool.
NDK:	Native Development Tool.
GNU:	GNU is Not Unix
ADB:	Android Debug Bridge.
DDMS:	Dalvik Debug Monitor Server.
BATMAN:	Better Approach To Mobile Adhoc Networking.
LAN:	Local Area Network.
IP :	Internet Protocol.
SIP:	Session Initiation Protocol.
PBX:	Private Branch eXchange.
FXS:	Foreign Exchange Station.
TTL :	Time To Life.
SSID:	Service Set IDentifier.
MP01:	Mesh Potato versión 01.
VoIP:	Voice over Internet Protocol.
MAC:	Media Access Control.
ARP:	Address Resolution Protocol.
DNA :	Distributed Numbering Architecture.
SID :	Subscriber ID.
DID:	Direct Inward-Dialled.

Contenido

Capítulo 1: Sistema Operativo Android	13
1.1 Inicios de Android.....	13
1.2 Android en la actualidad.	14
1.3 Actualizaciones.....	16
1.4 Nivel API.	17
1.5 Arquitectura de Android.	18
1.6 Funcionamiento de Android.....	19
1.6.1 Secuencias de arranque (Boot normal).....	19
1.6.2 Modos alternativos de arranque.....	20
1.6.3 Tipos de reset	21
1.6.4 Tipos de memoria.....	21
1.6.5 Estructura de sistema de ficheros YAFFS2	23
1.6.6 OTA Update (del inglés Over The Air)	25
1.6.7 Coberturas móviles	25
1.7 Teléfonos con Android.	25
1.8 Conclusión.	27
Capítulo 2: Entorno de desarrollo	28
2.1 Eclipse.....	28
2.1.1 Instalación de eclipse	29
2.2 SDK Android (kit de desarrollo de software).....	30
2.2.1 Instalar plataformas y otros componentes	30
2.3 Plugin ADT (Android Development Tool)	32
2.3.1 Descarga e instalación del ADT	32
2.3.2 Configuración del ADT.....	32
2.4 Creación de un AVD (Dispositivo Virtual Android).....	33
2.5 NDK (Native Development Tool).	35
2.5.1 Descargar e instalar Cygwin 1.7 o mayor	36
2.5.2. Descarga e instalación de GNU Make 3.81	36
2.6 Composición de un proyecto Android en eclipse.	37
2.6.1 Creación de un proyecto en eclipse	40
2.7 Estructura de una aplicación en Android.	42

2.8	Compilación e instalación de una aplicación.	44
2.8.1	Corriendo aplicación en el emulador	44
2.8.2	Corriendo aplicación en dispositivo móvil	44
2.9	Instalación de los drivers USB.	45
2.10	ADB (Android Debug Bridge).....	45
2.10.1	Detectar dispositivo o emulador	46
2.10.2	Instalando una aplicación.....	46
2.10.3	Otros comandos con ADB	47
2.11	Herramienta DDMS.	48
2.12	Conclusión.	50
Capítulo 3: La aplicación Batphone.....		51
3.1	Serval Project.....	51
3.2	Batphone.	52
3.3	Configuración preliminar en dispositivos móviles.	52
3.3.1	Configuración preliminar en el HTC Dream.....	52
3.3.2	Configuración preliminar en el htc comet (Huawei Ideos U8150).....	55
3.3.3	Configuración preliminar en el Huawei M835	56
3.4	Serval-batphone-helper.	56
3.4.1	Código base: Wifi Tether	56
3.4.2	Estructura del .apk del serval-bapthone-helper	57
3.5	Instalación de serval-batphone-helper en dispositivos.	60
3.5.1	Instalación de serval-batphone-helper a htc dream g1	60
3.5.2	Instalación de serval-batphone-helper a htc comet y metropcs	62
3.5.3	Configuración de la aplicación serval batphone.	62
3.5.4	Entorno de la aplicación batphone	63
3.6	Serval-Patched-Sipdroid.....	65
3.6.1	Sipdroid	65
3.6.2	Estructura del .apk del serval-patched-sipdroid	66
3.7	Instalación de serval-patched-sipdroid.	67
3.7.1	Configuración de la aplicación serval-patched-sipdroid.	67
3.8	SERVAL-0-05.....	68
3.8.1	Configuración del Serval Batphone 0-05.....	68

3.8.2 Menú Ajuste	69
3.9 Conclusión	69
Capítulo 4: Batphone en red mesh.	70
4.1 Red mesh.....	70
4.1.1 Configuración de la Red	71
4.1.2 Configuración de un router desde un dispositivo móvil	72
4.2 Realizar llamadas.....	72
4.3 Monitoreo de batphones.	73
4.4 DNA Arquitectura de Distribución Numérica.	75
4.5 Conclusión	77
ANEXO A	79
A. Versiones de Android.....	79
ANEXO B	83
B. Identificadores USB de los fabricantes.	83
BIBLIOGRAFIA	84

Lista de figuras

Figura 1. Gráfica de cantidad de Smartphone en uso para cada OS, basada en resultados publicados por ComScore.	14
Figura 2. Grafico de versiones de Android vendidas en móviles hasta 15 de junio del año 2011.....	16
Figura 3. Arquitectura de Android.	18
Figura 4. Secuencia de arranque del Huawei Ideos U8150 (a) IPL , (b) SPL	19
Figura 5. Modos de arranque en HTC Dream, (a) modo fastboot, (b) modo recovery.....	20
Figura 6. Tipos de reset	21
Figura 7. Android SDK Manager	32
Figura 8. Emulador corriendo con plataforma Android 2.1	33
Figura 9. Estructura de un proyecto en Eclipse.....	37
Figura 10. Creación de nuevo proyecto en Eclipse	41
Figura 11. Entorno del DDMS.....	48
Figura 12. Maquinas virtuales para diversas aplicaciones	49
Figura 13. Contenido de la carpeta src de la aplicación Serval-Batphone-Helper	57
Figura 14. Contenido de las carpetas jni, libs y native de la aplicación Serval-Batphone.....	58
Figura 15. Entorno de Serval Batphone (a) Setup, (b) Show Log.	64
Figura 16. Contenido de carpeta src de aplicación serval patched sipdroid.....	66
Figura 17. Red Mesh.....	70
Figura 18. (a) Llamada realizada al Huawei U8150 según tabla 8, (b) Llamada entrante.....	73
Figura 19. Aplicación Batphone ejecutándose.....	75
Figura 20. Resolución de una simple llamada usando Serval DNA	76

Lista de tablas

Tabla 1. Datos obtenidos hasta el 15 de junio del 2011	16
Tabla 2. Nivel API para las diferentes plataformas	17
Tabla 3, Características de los dispositivos móviles a configurar como batphone	27
Tabla 4. Características de los microprocesadores de los dispositivos móviles	27
Tabla 5. Opciones con las que puede arrancar el emulador.	35
Tabla 6. Opciones de ADB.	47
Tabla 7. Permisos de acceso a hardware de dispositivos	60
Tabla 8. Parámetros de configuración en Bapthone Serval Helper	64
Tabla 9. Versiones de la plataforma Android	82
Tabla 10. Identificadores de USB	83

Capítulo 1: Sistema Operativo Android.

Este capítulo inicia presentando una breve historia del sistema operativo Android, desde su origen hasta la actualidad, y el papel que tiene en el presente mercado mundial de teléfonos móviles.

Se estudia la evolución del proyecto Android desde las versiones betas hasta las actuales, mostrando un resumen de las características que identificaban a cada versión. Se describe el significado del término API y como éste se relaciona con cada versión de Android.

Luego, se analiza de manera breve la arquitectura del sistema operativo. Es decir las cuatro capas que lo conforman. Para comprender como el usuario a través de una aplicación se relaciona con el hardware del terminal.

Se explica de manera breve la relación con el software y hardware; aprovechando para definir algunos conceptos necesarios. Finalizando con la descripción física de cada uno de los tres modelos a utilizar para el presente proyecto.

1.1 Inicios de Android.

En junio de 2005 Google compró Android Inc., una pequeña empresa cuya finalidad era el desarrollo de aplicaciones para dispositivos móviles ubicada en Palo Alto, California.

En ese momento empezó la especulación de que Google estaría desarrollando un teléfono móvil. En un primer momento lo que hacía atractivo este dispositivo no era el nivel tecnológico, facilidad de uso, ni se sabía que sería de código abierto, sino que existía la posibilidad de que el dispositivo y/o el costo del servicio fueran subsidiados por publicidades en el móvil.

Android es un sistema operativo para dispositivos móviles basada en el kernel de Linux. Se informó que Google había alineado ya una serie de fabricantes de hardware y software y se señaló a los operadores que estaba abierto a diversos grados de cooperación por su parte.

El lanzamiento de Android se realizó el 5 de noviembre de 2007. Además se anunció 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. Entre las compañías que forman el consorcio están: Texas Instruments, Broadcom Corporation, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, Intel, LG, Marvell Technology Group, Motorola, y T-Mobile [1].

Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto, estrenando su primer producto Android en una plataforma para dispositivos móviles construidas en la versión 2.6 del kernel de Linux [2].

El 9 de diciembre de 2008, se anunció que 14 nuevos miembros se unirían al proyecto Android, incluyendo PacketVideo, ARM Holdings, Atheros Communications, Asustek, Garmin, Softbank, Sony Ericsson, Toshiba y Vodafone [3].

1.2 Android en la actualidad.

En el mercado mundial existe una contienda de teléfonos móviles inteligentes, con diferentes sistemas operativos y cada uno tiene sus ventajas y desventajas.

Los teléfonos con sistema operativo Android están ganando muchos seguidores en el mercado mundial. Según un estudio realizado por una compañía de marketing en internet (ComScore), el sistema operativo de Google ha superado los demás OS en el mercado en menos de un año.

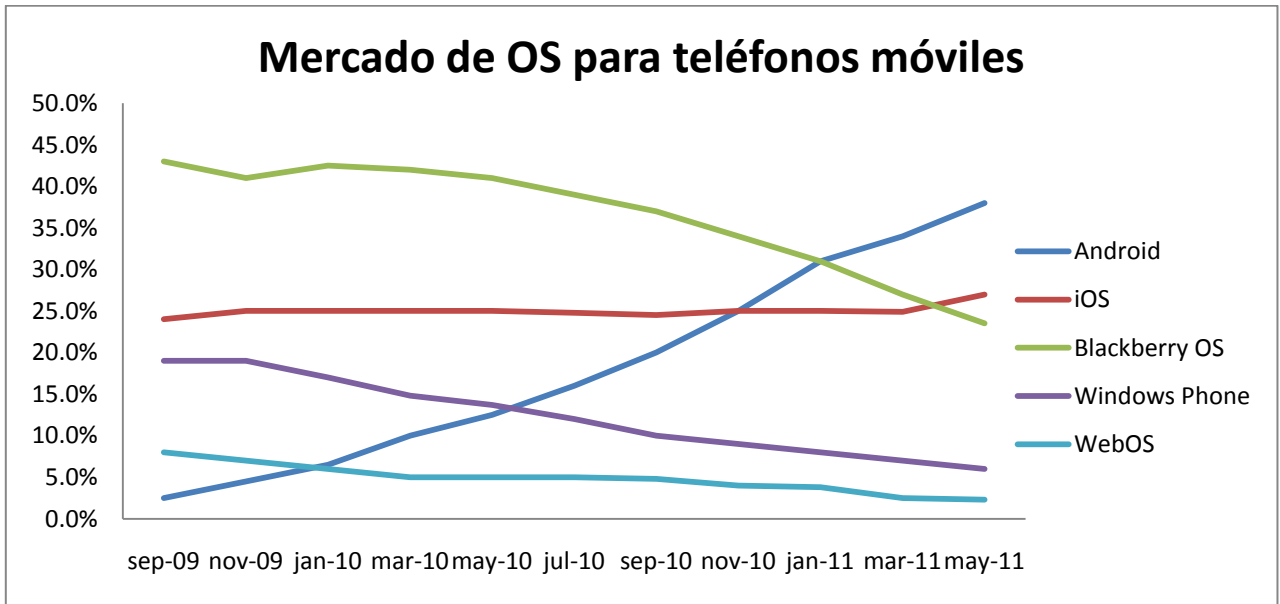


Figura 1. Gráfica de cantidad de Smartphone en uso para cada OS, basada en resultados publicados por ComScore [4].

Al analizar la Figura 1, se observa que desde septiembre del 2009 Android presenta un crecimiento casi lineal. A partir de abril de 2011 este ya tiene una ventaja aproximada del 10 % sobre el iOS y el Blackberry OS. Con la velocidad con la cual el Blackberry OS pierde seguidores parece que iOS y Android representarían todo el mercado. De hecho, si Android sigue creciendo así se cumplirá la predicción de que poseerá el 50% del mercado para el 2012 e incluso antes.

A continuación se presenta un breve resumen de cada uno de los sistemas operativos rivales de Android en el mercado mundial.

- **iOS** : Anteriormente denominado iPhone OS, es un sistema operativo móvil de Apple desarrollado originalmente para el iPhone, siendo después usado en el iPod Touch e iPad. Es un derivado de Mac OS X.

La interfaz de usuario de iOS se basa en el concepto de manipulación mediante gestos multitáctil. Los elementos de la interfaz se componen por deslizadores, interruptores y botones. La respuesta es inmediata y se provee de una interfaz fluida. La interacción con el sistema operativo se realiza mediante gestos como deslizar, y tocar. Acelerómetros y Giroscopios internos son utilizados por algunas aplicaciones para responder a movimientos y gestos, como sacudir el aparato (en campos de texto es usado para deshacer y rehacer) o rotarlo (se suele usar para cambiar de posición vertical a modo paisaje) [5].

- **Blackberry OS:** El BlackBerry OS es un sistema operativo móvil desarrollado por Research In Motion para sus dispositivos BlackBerry. El sistema permite multitarea y tiene soporte para diferentes métodos de entrada adoptados por RIM para su uso en computadoras de mano, particularmente la trackwheel, trackball, touchpad y pantallas táctiles.

Su desarrollo se remonta a la aparición de los primeros portátiles en 1999. Estos dispositivos permiten el acceso a correo electrónico, navegación web y sincronización con programas como Microsoft Exchange o Lotus Notes aparte de poder hacer las funciones usuales de un teléfono móvil.

Al igual que en el OS Symbian desarrolladores independientes también pueden crear programas para BlackBerry pero en el caso de querer tener acceso a ciertas funcionalidades restringidas necesitan ser firmados digitalmente para poder ser asociados a una cuenta de desarrollador de RIM [6].

- **Windows Phone :** Windows Phone, anteriormente llamado Windows Mobile es un sistema operativo móvil compacto desarrollado por Microsoft, y diseñado para su uso en teléfonos inteligentes (Smartphones) y otros dispositivos móviles. Windows Phone hace parte de los sistemas operativos con interfaz natural de usuario.

Se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows. Está diseñado para ser similar a las versiones de escritorio de Windows estéticamente. Además, existe una gran oferta de software de terceros disponible para Windows Mobile, la cual se puede adquirir a través de Windows Marketplace for Mobile.

Originalmente apareció bajo el nombre de Pocket PC, como una ramificación de desarrollo de Windows CE para equipos móviles con capacidades limitadas. Windows Mobile ha evolucionado y cambiado de nombre varias veces durante su desarrollo, siendo la última versión la llamada Windows Phone 7, anunciada el 15 de febrero del 2010 y sujeta a disponibilidad a finales de 2010 [7].

- **webOS:** HP webOS es un sistema operativo multitarea para sistemas basado en Linux. Fue desarrollado por Palm, Inc., ahora propiedad de Hewlett-Packard Company. Fue presentado al público el 8 de enero de 2009 junto con el Palm Pre. El Palm Pre y webOS fueron lanzados el 6 de junio de 2009. Un segundo dispositivo utilizando este sistema, el Palm Pixi, fue lanzado el 15 de noviembre de 2009.

La interfaz gráfica de usuario de webOS fue diseñada para dispositivos con pantalla táctil. Incluye un conjunto de aplicaciones para el manejo de la información personal y usa tecnologías web como HTML5, JavaScript y CSS. Palm asegura que el diseño alrededor de estas tecnologías fue intencionado, para evitar así el aprendizaje de un nuevo lenguaje de programación por parte de los desarrolladores. El Palm Pre, lanzado el 6 de junio de 2009, fue el primer dispositivo en utilizar esta plataforma. Los dispositivos que actualmente utilizan webOS son el Palm Pre y el Palm Pixi, así como sus versiones "Plus" en Verizon Wireless, AT&T y algunos operadores internacionales [8].

1.3 Actualizaciones.

Las actualizaciones del sistema operativo base consisten en arreglos de fallas y la incorporación de nuevas funciones. Generalmente cada actualización del sistema operativo Android es desarrollada bajo un nombre en código de un elemento relacionado con postres. En el anexo A se encuentra detalladas estas versiones con sus respectivas características

En la Tabla 1, se muestra un porcentaje de las versiones de Android que se encuentra en los diversos móviles. La estimación fue realizada en el mes de marzo para el año 2011, según ComScore [9]. En la figura 2 se muestra la gráfica de los datos mostrados en la Tabla 1

Plataforma	Nivel de API	Cuota de mercado
Android 3.1	12	0.3%
Android 3.0	11	0.3%
Android 2.3.3	10	8.1%
Android 2.3	9	1.1%
Android 2.2	8	64.6%
Android 2.1	7	21.2%
Android 1.6	4	2.5%
Android 1.5	3	1.9%

Tabla 1. Datos obtenidos hasta el 15 de junio del 2011

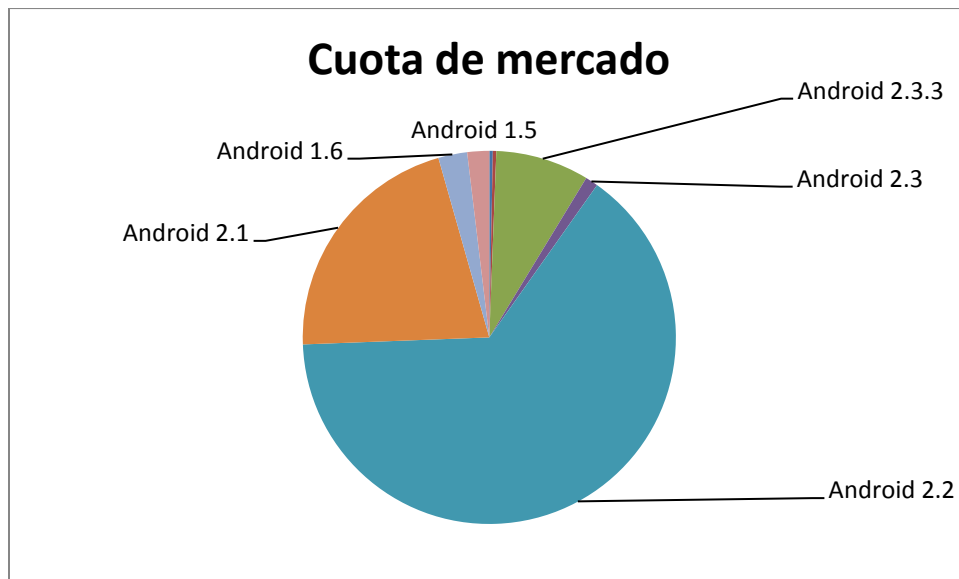


Figura 2. Gráfico de versiones de Android vendidas en móviles hasta 15 de junio del año 2011

1.4 Nivel API.

Al desarrollar una aplicación para Android, es útil entender lo que es el identificador del nivel API (del inglés Application Programming Interface). Una breve introducción es decir que este identifica si la aplicación que se va a crear tiene compatibilidad con el dispositivo en cual se va a instalar. A continuación se brinda información sobre el nivel API y su importancia en las aplicaciones creadas.

El nivel API es un identificador (número entero) exclusivo para el framework ofrecido por cada versión del sistema operativo Android. Es decir, cada versión tiene una única API que la identifica, según la Tabla 2.

Versión de la plataforma	Nivel API
Android 1.0	1
Android 1.1	2
Android 1.5	3
Android 1.6	4
Android 2.0	5
Android 2.01	6
Android 2.1	7
Android 2.2	8
Android 2.3	9
Android 2.3.3	10
Android 2.3.4	10
Android 3.0	11
Android 3.1	12

Tabla 2. Nivel API para las diferentes plataformas.

Las plataformas Android ofrecen un framework que las aplicaciones utilizan para interactuar con la base del sistema Android. El framework API se compone de:

- Un grupo básico de paquetes y clases
- Un grupo de elementos XML y atributos para declarar un archivo Manifest
- Un grupo de elementos XML y atributos para declarar y acceder a recursos.
- Un conjunto de permisos que las aplicaciones pueden solicitar, entre estos el acceso root que se requiere para el presente trabajo.

Por lo tanto la actualización de una versión de Android, actualiza el framework API. Cada vez que se publica una nueva versión de OS Android el indicador de nivel API incrementa.

Otro dato importante es que las API de mayor nivel pueden ejecutar aplicaciones que fueron diseñadas para un nivel menor de API. Por lo tanto al actualizar una versión de Android se agregan nuevas funciones al framework de su API y de esta forma, puede ejecutar aplicaciones de menor nivel API.

1.5 Arquitectura de Android.

A continuación se presentan las cuatro capas de las que está formada el sistema y la dependencia entre ellas (Figura 3). Cada una de las capas se encapsula y provee una interfaz de comunicación suficiente para que cada capa cumpla con su función específica.

En la capa de aplicaciones que es el nivel superior se encuentra todas las aplicaciones creadas para la plataforma Android, entre estas deben estar: cliente de email, calendario, programa de SMS, mapas, navegador, contactos. Todas escritas en el lenguaje de programación JAVA [10].

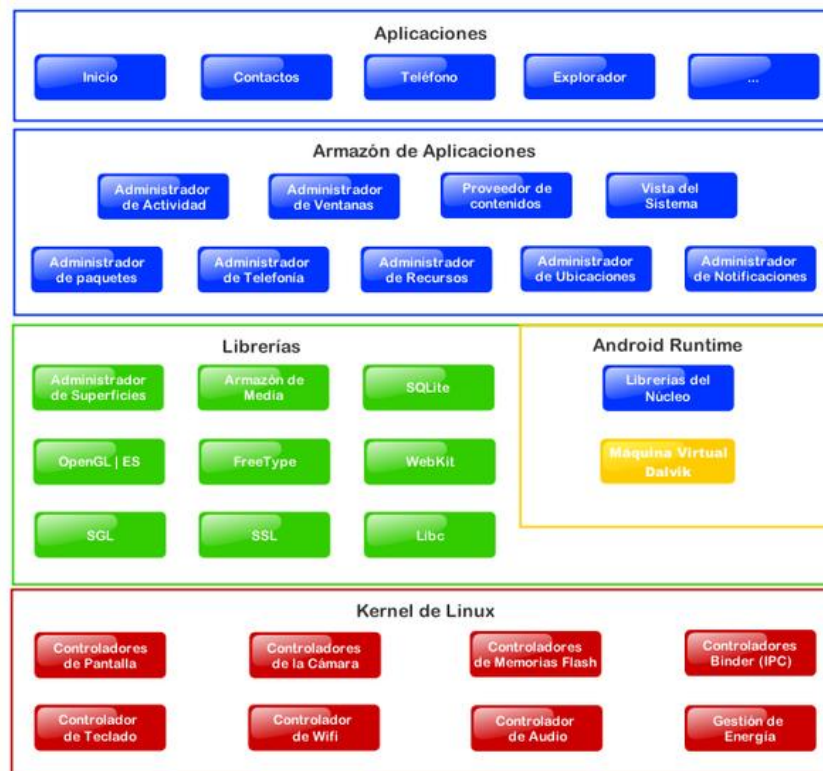


Figura 3. Arquitectura de Android. [11]

La capa de Armazón de aplicaciones (Framework) se encarga de gestionar las distintas llamadas que realizan las aplicaciones [10].

En la capa tres se encuentran las librerías que Android incluye en su base de datos. Estas son un set de librerías C/C++, que son expuestas a todos los desarrolladores a través del framework de las aplicaciones Android System C library, librerías de medios, librerías de gráficos, 3D, SQLite, etc [10]. También en este nivel se encuentra el Runtime, el cual está conformado por la máquina virtual Dalvik, que es un software que emula a una computadora y puede ejecutar programas como si fuera una computadora real.

Dalvik está optimizada para requerir poca memoria y está diseñada para permitir y ejecutar varias instancias de la máquina virtual simultáneamente, delegando en el sistema operativo el soporte de aislamiento de procesos, gestión de memoria.

La máquina virtual Dalvik ejecuta los ficheros .dex que se encuentran empaquetados en un .apk (Es el termino que se les asigna a todas las aplicaciones en Android, en otras palabras es la

compresión en formato .zip), esto lo realiza traduciendo los .class de JAVA a .dex usando la herramienta dx.

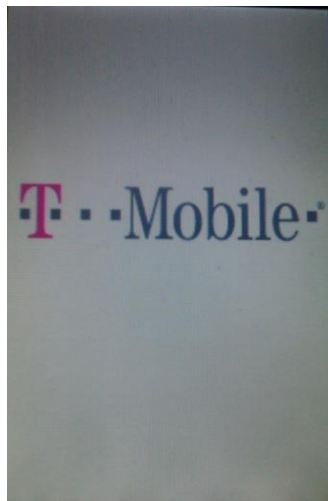
En conclusión las aplicaciones de Android se ejecutan en el interior de la máquina virtual y usa Linux como núcleo de servicios en la que se ejecuta la máquina virtual [10].

En la capa Kernel de Linux se gestiona el hardware. Android usa el Kernel de Linux 2.6 para comunicarse con la capa de Hardware, esto permite a Android ser compatible con muchos de los drivers creados para Linux facilitando de este modo ciertos desarrollos.

1.6 Funcionamiento de Android

1.6.1 Secuencias de arranque (Boot normal): Este proceso se origina al presionar el botón de encendido del dispositivo, llevándose a cabo los siguientes procesos, con el orden que se presentan:

- **IPL (del inglés Initial Program Loader):** cargador inicial de programa equivalente a la BIOS que inicia el SPL colocándolo en la RAM. En este momento normalmente aparece en la pantalla del dispositivo la marca y el modelo. Ver figura 4(a)



(a)



(b)

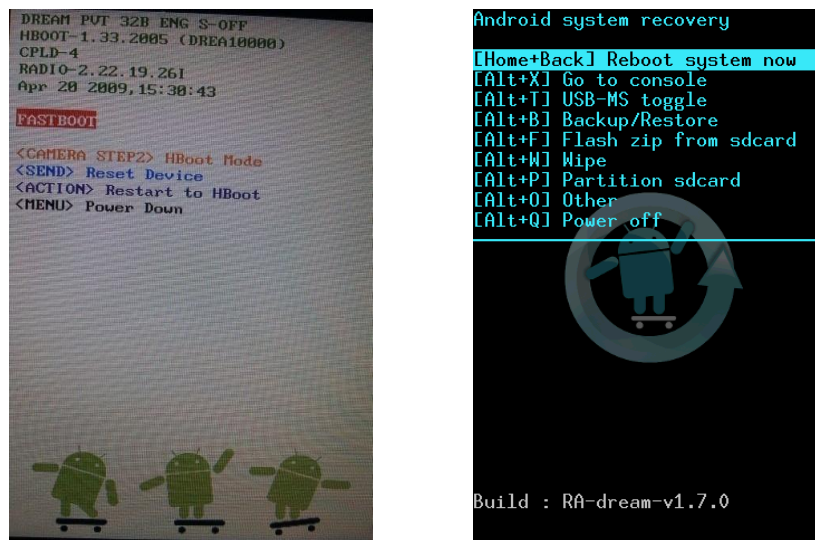
Figura 4. Secuencia de arranque del Huawei Ideos U8150 (a) IPL , (b) SPL.

- **SPL (del inglés Second Program Loader):** cargador secundario de programa o bootloader que se encarga de cargar la Radio/GSM y el sistema operativo.
- **RADIO:** Es un controlador de hardware, está relacionado con las comunicaciones vía GSM del dispositivo, con la cámara, el GPS.
- **ROM:** En este se encuentra instalado el sistema operativo Android, mientras se carga aparece "ANDROID" en la pantalla.

1.6.2 Modos alternativos de arranque: Además de iniciar el dispositivo de una manera normal, se pueden realizar otros tipos de arranques al presionar una combinación de teclas cuando el teléfono está apagado:

- **FastBoot Mode:** Se logra al combinar los botones cuando el teléfono está apagado “BACK + POWER ON”. Es un protocolo utilizado para actualizar el sistema de archivos flash en los dispositivos Android desde una computadora a través de USB. Este permite formatear imágenes de particiones no asignadas en la memoria MTD (del inglés Memory Technology Device). Esta opción esta deshabilitada mientras no se tenga acceso root.

Dentro del Fastboot, se puede obtener información del teléfono, como por ejemplo: versión de la radio, versión del SPL, etc. Según se puede apreciar en la figura 5 (a)



(a)

(b)

Figura 5. Modos de arranque en HTC Dream, (a) modo fastboot, (b) modo recovery

- **Recovery:** Ayuda a iniciar la carga del sistema operativo dentro de la RAM. Las versiones del recovery mode pueden cambiar para distintas versiones de teléfonos o versiones del sistema operativo. Permite hacer cambios en la ROM, hacer backup de ella o modificar propiedades de la tarjeta SD (del inglés Secure Digital, es una memoria externa que se utiliza para expandir el almacenamiento del dispositivo), reiniciar el teléfono a propiedades de fábrica. El ingreso al modo recovery se obtiene con la combinación de teclas “POWER + Volumen abajo”, cuando el teléfono este apagado. El menú de este se puede apreciar en la figura 5(b).

En el modo Recovery, se presentan varias opciones:

++ reboot system now [Home+Back]: Al presionar la combinación de teclas que se indica, se reinicia el teléfono.

++ apply sdcard: update.zip [Alt+S]: Permite cargar archivos desde la memoria SD. El archivo que se desea cargar, debe renombrarse como “update.zip”.

++ wipe data/factory RESET [Alt+W]: Permite dejar al móvil con los valores de fábrica, es decir un Hardreset.

- **Safe Mode Boot:** Este proceso permite limpiar un largo número de problemas y puede ser utilizado antes de un Hard Reset. Se obtiene al mantener presionado la tecla “HOME” al encender el teléfono.

1.6.3 Tipos de reset:

- **Soft Reset:** Se usa para que el terminal reinicie sin necesidad de perder la personalización del dispositivo, es decir, sin que se borren los datos de agenda, calendarios, configuraciones, aplicaciones y juegos que se tengan en el mismo. Se puede considerar reset de emergencia. La combinación de los botones “MENU + CALL + POWER ON” realiza el soft reset como lo muestra la Figura 6.
- **Hard Reset:** Reinicia el teléfono de forma tal que se borre todo el contenido que se ha añadido, es decir, datos de agenda, calendarios, configuraciones, aplicaciones, juegos, etc., quedando el teléfono con el software limpio como salió de fábrica permitiendo volver a personalizarlo como se tenía.

Los pasos a seguir en el teléfono para ejecutar un hard reset son: presionar la tecla menú luego configuración > privacidad > Restablecer datos de fábrica.



Figura 6. Tipos de reset

1.6.4 Tipos de memoria.

- **RAM:** Es donde el sistema almacena las últimas aplicaciones o elementos que se están utilizando.

Android es un sistema operativo multitarea, lo que implica que puede haber varias aplicaciones y procesos ejecutándose al mismo tiempo. Debido a ello, a medida que se abren y cierran aplicaciones la interfaz se ralentiza. Esto se debe a que algunas aplicaciones no tienen una opción para cerrarse explícitamente y quedan corriendo en segundo plano hasta que Android decida cancelarlas para liberar RAM.

Mientras haya memoria RAM disponible, Android hará lo posible por mantener dichas aplicaciones y procesos en ella. Si en algún momento se inicia una aplicación y Android detecta que sólo hay disponible cierta cantidad de RAM, procederá a cancelar algún proceso o aplicación que esté residente en memoria.

El sistema operativo Android gestiona el uso de la memoria RAM utilizando seis niveles de procesos:

- ++ Nivel 1: Primer plano
- ++ Nivel 2: Visible
- ++ Nivel 3: Segundo plano
- ++ Nivel 4: Oculto
- ++ Nivel 5: Proveedor de contenidos
- ++ Nivel 6: Vacío

El nivel 1 es el más importante y las aplicaciones del nivel 6 son los primeros que se pueden eliminar. Además para cada uno de estos niveles hay un límite mínimo de memoria a partir del cual Android comenzará a eliminar procesos de dicho nivel. Por ejemplo, se puede especificar que las aplicaciones de nivel 6 se deben empezar a liberar cuando haya menos de 26MB de RAM libres. Los comandos a ejecutar para obtener el número de página de los distintos niveles en la consola del dispositivo son

```
$ cat /sys/module/lowmemorykiller/parameters/minfree
cat /sys/module/lowmemorykiller/parameters/minfree
1536,2048,4096,7168,8192,8192
$ cat /proc/meminfo
cat /proc/meminfo
MemTotal:          183060 kB
MemFree:           16216 kB
Buffers:            24 kB
Cached:            43248 kB
SwapCached:        0 kB
Active:            68484 kB
Inactive:          76404 kB
Active(anon):      44788 kB
Inactive(anon):    57720 kB
Active(file):      23696 kB
Inactive(file):    18684 kB
Unevictable:       532 kB
Mlocked:           0 kB
SwapTotal:         0 kB
```

En Android una página equivale a 4kB. Para conocer los límites de memoria que el sistema tiene configurado para cada nivel, desde una terminal, se ingresa el comando que aparece en la segunda línea de la ventana anterior, obteniendo los números de páginas 1536,2048,4096,7168,8192, y 8192 , para los respectivos niveles 1,2,3,4,5 y 6. Al realizar una regla de tres obtenemos;

1 Mega = 1024 kB , entonces para una página 4kB = 3.90625 kB

Para 1536 páginas = 6MB

Por lo tanto el resultado total es: 6MB, 8MB, 16MB, 28MB, 32MB y 32MB.

Para obtener detalles de la memoria RAM del móvil se consulta el archivo *meminfo* que se encuentra alojado en el directorio proc. Según los resultados mostrados en la ventana anterior el teléfono tiene una memoria total aproximadamente de 183 MB.

- **Tarjetas microSD**

Son utilizadas para expandir la cantidad de memoria del dispositivo. Estas se clasifican en clases según los tipos de velocidad. Estos tipos pueden ser: clase 2, clase 4, clase 6 y clase 10.

- **MTD (del inglés Memory Technology Device)**

Es la memoria interna del teléfono, esta cuenta con varias particiones. Es del tipo flash NAND.

En la memoria MTD existen 6 particiones, en cada una de ellas se almacenan archivos específicos desde datos del usuario hasta archivos propios del sistema. Existen espacios no asignados en memoria NAND, donde ya se encuentran instalados el SPL. En la partición MTD4 se aloja la maquina virtual Darvik.

Para verificar las particiones en Android se consulta el archivo mtd que se encuentra en el fichero proc, para ello hay que ingresar los siguientes comandos en la consola del dispositivo.

```
# cat /proc/mtd
cat /proc/mtd
dev:      size  erasesize  name
mtd0: 00040000 00020000 "misc"
mtd1: 00500000 00020000 "recovery"
mtd2: 00280000 00020000 "boot"
mtd3: 05a00000 00020000 "system"
mtd4: 01e00000 00020000 "cache"
mtd5: 059c0000 00020000 "userdata"
```

En estas particiones se encuentran:

- ++ mtd1 "recovery", en esta partición se aloja la secuencia de arranque Recovery.
- ++ mtd2 "boot", en ella se encuentra el kernel de Android.
- ++ mtd3 "system", tiene el sistema operativo Android (y aplicaciones de sistema) que se monta (en solo lectura) en modo normal.
- ++ mtd4 "cache", dentro de esta se encuentra la maquina virtual Darvik.
- ++ mtd5 "userdata", se monta (en lectura/escritura) en modo normal, en ella van las configuraciones de las aplicaciones y los datos del usuario.

1.6.5 Estructura de sistema de ficheros YAFFS2

Un sistema de ficheros es un método para el almacenamiento y organización de archivos y los datos que estos contienen. Los sistemas de archivos son usados en dispositivos de almacenamiento como discos duros y CD-ROM e involucran el mantenimiento de la localización física de los archivos.

Android utiliza el sistema de ficheros YAFFS2. Éste es el primer sistema de ficheros que fue diseñado específicamente para las memorias Flash NAND. Es muy robusto ante fallos de energía. Al ejecutar el comando `mount | grep mtd`, en la consola del dispositivo, se muestran las particiones mtd montadas.

```
# mount | grep mtd
mount | grep mtd
/dev/block/mtdblock3 on /system type yaffs2 (ro,relatime)
/dev/block/mtdblock5 on /data type yaffs2 (rw,nosuid,nodev,relatime)
/dev/block/mtdblock4 on /cache type yaffs2 (rw,nosuid,nodev,relatime)
```

Las particiones son del tipo yaffs2 y el almacenamiento `"/dev/block/mtdblock"` es un disco flash RAM de tipo mtd.

Al ubicarse en la raíz del sistema puede observarse los ficheros, con el comando `ls`. El contenido de algunos de estos ficheros se detallan a continuación.

- **/**: Raíz del sistema de archivos.
- **/dev**: Contiene ficheros del sistema representando los dispositivos que estén físicamente instalados en el ordenador. Estos son conocidos también como los drivers.
- **/etc**: Este directorio está reservado para los ficheros de configuración del sistema. En este directorio no debe aparecer ningún fichero binario (programas).
- **/proc** : Contiene ficheros especiales que reciben o envían información al kernel del sistema . Es recomendable no modificar el contenido de este directorio y sus ficheros.
- **/sbin** : Contiene programas que son únicamente accesibles al superusuario o root.
- **/system**: Alojada en la partición system de la memoria mtd donde se encuentra alojado el sistema operativo, es decir contiene la ROM. Forma parte de este fichero la carpeta media, que contiene los sonidos, los ringtones, todo lo relación a archivos de audio.
- **/data**: En este fichero se encuentra, muchos archivos y carpetas relacionados a las aplicaciones o las .apk; entre estos se tiene la carpeta tombstones que es el cementerio de aplicaciones dañadas, también se encuentra la carpeta app donde se almacenan las aplicaciones, la carpeta dalvik-cache que como su nombre lo indica , aloja la memoria cache de la maquina virtual.

El fichero data también contiene las carpetas que almacén el ANR y el FC. Estos son dos mecanismos que Android posee para que el sistema no se bloquee o estropee ante una aplicación defectuosa o un evento temporal descontrolado.

Un ANR (del inglés Application Not Respondig) ocurre cuando las aplicaciones realizan una gran cantidad y variedad de tareas, dejando la aplicación en un estado de espera, saturando la eficiencia del procesador. Durante este período, si el sistema detecta que se está tocando la pantalla sin recibir respuesta alguna de la aplicación en ejecución, o simplemente detecta que la aplicación lleva demasiado tiempo en dicho estado, mostrará un mensaje advirtiendo que la aplicación no responde apropiadamente, dejándonos dos opciones: "Forzar cierre" o "Esperar".

Un FC (del inglés Force Close), es cuando el sistema detecta un error no controlado por el programador en una aplicación muestra una ventana con una única opción: "Forzar cierre".

Formando parte de este fichero se encuentra el JIT (del inglés Just In Time). Desarrollo exclusivo de los ingenieros de Google. No es más que un compilador para el Dalvik, permitiendo que el software se ejecute más rápido debido a que la compilación lo hace en código máquina nativo. Con lo cual se consigue mejorar bastante el rendimiento de la maquina virtual que es la encargada de mover el software del teléfono.

1.6.6 OTA Update (del inglés Over The Air)

Otra de las ventajas que presentan los móviles con el sistema operativo Android es la instalación de aplicaciones sobre el aire. Esto brinda la comodidad de instalar una aplicación con solo aceptar desde el teléfono y luego de que se ha realizado la descarga, automáticamente se realiza la instalación. Un ejemplo de este servicio es al utilizar el market de Android. El market de Android es una tienda de software en línea desarrollada por Google para los dispositivos Android. Es una aplicación llamada "Mercado" que está preinstalada en la mayoría de los dispositivos Android y que permite a los usuarios buscar y descargar aplicaciones publicadas por desarrolladores terceros [12].

1.6.7 Coberturas móviles

Los teléfonos que se utilizan en este trabajo, tienen la opción de trabajar bajo cualquiera de cuatro coberturas móviles, estas son:

- **GPRS:** "General Packet Radio Service", es una tecnología digital de telefonía móvil la cual proporciona velocidades de transferencia de datos superiores a las generadas por la tecnología GSM. Tiene un ancho de banda de 80kbps.
- **EDGE:** "Enhanced Data rates for GSM of Evolution", es una tecnología de la telefonía celular móvil que actúa como puente entre las redes 2G (segunda generación) y 3G (tercera generación). Para esa opción se tiene un ancho de banda de 236 kbps.
- **3G (UMTS):** 3G "Universal Mobile Telecommunications System" es la abreviación de tercera generación de transmisión de voz y datos a través de telefonía móvil. Los servicios asociados con la tercera generación proporcionan la posibilidad de transferir tanto voz y datos (una llamada telefónica o una video llamada). Aunque esta tecnología estaba orientada a la telefonía móvil, desde hace unos años las operadoras de telefonía móvil ofrecen servicios exclusivos de conexión a Internet mediante módem usb, sin necesidad de adquirir un teléfono móvil, por lo que cualquier computadora puede disponer de acceso a Internet. El ancho de banda en esta opción es de 2Mbps.
- **3G+(HSPA):** "High Speed Downlink Packet Access". Su ancho de banda es de 14Mbps.

1.7 Teléfonos con Android.

Para el presente proyecto, se trabajó con tres teléfonos que poseen Android, uno con la versión Eclair (Android 2.1): el HTC Dream; y dos con la versión Froyo (Android 2.2) siendo el Huawei Ideos U8150 y el Huawei M835.

A continuación se muestran las especificaciones de cada uno de los celulares, ver Tabla 3

CARACTERÍSTICAS	HTC DREAM	HUAWEI U8150	HUAWEI M835
Año de fabricación	2008	2010	2011
Red 2G	Quad Band (GSM 850/900/1800/1900 MHz)	Quad Band (GSM 850/900/1800/1900 MHz)	Quad Band (GSM 850/900/1800/1900 MHz)
Red 3G	HSDPA 1900/2100	HSDPA 1900/2100	HSDPA 1900/2100
Dimensiones	117x55x17 mm	104 x 54.8 x 13.5 mm	105 X 55.8 X 13.8 mm
Peso	158 gramos	100 gramos	110 gramos
Pantalla	Táctil, 3.2 pulgadas	Táctil, 2.8 pulgadas	Táctil, 2.8 pulgadas
Procesador	Qualcomm MSM7201A	Qualcomm MSM7225	Qualcomm MSM7225
Memoria ROM	256 MB	512 MB	512 MB
Memoria RAM	192 MB	256 MB	256 MB
Teclado	Qwerty deslizante de 5 filas	Táctil	Táctil
Conectividad	Bluetooth® 2.0 con ERD Wi-Fi®: IEEE 802.11b/g HTC ExtUSB™	Bluetooth® 2.1 con ENR Wi-Fi®: IEEE 802.11b/g	Bluetooth® 2.1 Wi-Fi®: IEEE 802.11b/g
Cámara	3.2 mega píxeles con auto focus	3.2 mega píxeles con enfoque automático y flash	3.2 mega píxeles con enfoque automático y flash
Audio	Micrófono y altavoz incorporados Formatos de timbres: * AAC, AAC+, AMR-NB, MIDI, MP3, WMA, WMV * 40 polifónicos y estándar formato MIDI 0 and 1 (SMF)/SP MIDI	Formatos de timbres: * AAC, AAC+, AMR-NB, MIDI, MP3, WMA, WMV * 40 polifónicos y estándar formato MIDI 0 and 1 (SMF)/SP MIDI	Formatos de timbres: * AAC, AAC+, AMR-NB, MIDI, MP3, WMA, WMV * 40 polifónicos y estándar formato MIDI 0 and 1 (SMF)/SP MIDI
Batería	Lithium-ion recargable Capacidad: 1150 mAh	Batería recargable de iones de litio o de polímeros de iones de litio. Capacidad: 1200 mAh	Batería recargable de iones de litio o de polímeros de iones de litio. Capacidad: 1200 mAh
Ranura de expansión	Tarjeta de memoria microSD™ (SD 2.0 compatible).	Tarjeta de memoria microSD™ (compatible con SD 2.0),admite hasta 32 GB.	Tarjeta de memoria microSD™ (compatible con SD 2.0),admite hasta 32 GB.
Version de Android	2.1 Eclair	2.2 Froyo	2.2 Froyo

Tabla 3. Características de los dispositivos móviles a configurar como batphone

El primer teléfono móvil en usar el sistema operativo Android, de Google fue el HTC Dream o T-Mobile G1. Recibe el nombre de HTC debido al fabricante y T-Mobile por la operadora que lo distribuyó [13]. El otro teléfono a utilizar el Huawei IDEOS es un teléfono celular de bajas pretensiones y económico, destinado a llevar Android a las masas [14]. La versión de Android que se incluye en este móvil es el Froyo, para su configuración solamente se activaron los permisos root.

Al observar las especificaciones técnicas de los tres modelos, puede notarse que el procesador de los móviles es de la familia Qualcomm. En la Tabla 4 se puede mostrar las características para estos dos procesadores.

CARACTERISTICA	QUALCOMM MSM7210A	QUALCOMM MSM7225
Año de lanzamiento	2008	2007
Predecesor	32bit Qualcomm MSM7200A	32bit Qualcomm MSM7225
Estructura del CPU	RISC	RISC
Bus primario de datos (RAM)	32 bit	32 bit
Grupos de instrucciones soportadas	ARMv6	ARMv6
Núcleo del CPU	ARM1136EJ-S	ARM1136EJ-S
Frecuencia del reloj	528 MHz	528 MHz
Tecnología del semiconductor	CMOS	CMOS

Tabla 4. Características de los microprocesadores de los dispositivos móviles

1.8 Conclusión.

Al comprender las características de hardware para cada dispositivo se entenderá con mayor facilidad cada uno de los cambios que se deben realizar en cada uno de ellos para poder lograr la configuración de batphone y poder formar parte de una red inalámbrica de tipo Mesh.

Android al estar basado en el kernel Linux presenta la facilidad de trabajar con los ficheros de los dispositivos en un bajo nivel y de esta manera configurar los archivos necesarios.

Finalmente el presentar la comparación de cada uno, existen dos características en cada uno de ellos que hay que destacar y que son indispensables para poder configurarse como un batphone. Estas características fundamentales son el Wifi y que el procesador soporte instrucciones ARM, de no ser así estos no podrían configurarse como se desea.

Capítulo 2: Entorno de desarrollo

En el presente capítulo se explicará todo el software relacionado para crear una aplicación en Android. Así como los complementos que se utilizan para poder simular un teléfono móvil en la computadora.

También se analizará la estructura de las aplicaciones en Android y como éstas pueden contener lenguajes nativos. De igual manera se presenta la estructura de las aplicaciones en el entorno de desarrollo Eclipse.

2.1 Eclipse

Para comenzar a crear aplicaciones es necesario decidir en qué sistema operativo se trabajará, y luego tomar la decisión sobre que entorno de desarrollo se utilizará. Se puede usar con el que se esté más relacionado, ya sea NetBeans o cualquier versión de Eclipse. En este documento se explica el uso de entorno de desarrollo Eclipse.

La rápida expansión de Java en los años 90 fue la causa para que se originara la fundación Eclipse. Ésta es un fundación sin fines de lucro que creó el entorno de desarrollo de código abierto, que lleva el mismo nombre "Eclipse". A este consorcio se han unido progresivamente importantes empresas de desarrollo de software a nivel mundial: Oracle, SuSe, HP, Ericsson, Novell, entre otras. Hay dos ausencias significativas: Microsoft y Sun Microsystems. Microsoft ha sido excluida por su posición de monopolio del mercado y Sun Microsystem cuenta con su propio IDE y principal competencia de Eclipse: NetBeans. De hecho, el nombre de Eclipse fue elegido porque el objetivo era crear un IDE capaz de "eclipsar a Visual Studio" (Microsoft) así como "eclipsar el sol" (Sun Microsystem) [15].

La plataforma Eclipse consiste en un Entorno de Desarrollo Integrado (IDE, Integrated Development Environment) de código abierto y multiplataforma. Un IDE es un programa compuesto por un conjunto de herramientas útiles para un desarrollador de software. Como elementos básicos, un IDE cuenta con en un editor de código, un compilador/intérprete y un depurador. Eclipse sirve como IDE Java y cuenta con numerosas herramientas de desarrollo de software. Aunque está orientado a la programación java, que es el lenguaje que tiene mayor jerarquía en las aplicaciones Android, también da soporte a otros lenguajes de programación, como son C/C++, Fortran, PHP o Python. Inclusive puede combinar dos tipos de lenguaje, añadiendo librerías nativas a la aplicación.

La última versión estable de Eclipse se encuentra disponible para los sistemas operativos Windows, Linux, Solaris, AIX, HP-UX y Mac OSX. Estas versiones pueden descargarse de la página oficial de Eclipse [16]:

<http://www.eclipse.org/downloads/>

Cabe aclarar que durante el desarrollo este documento la instalación de Eclipse se realizó en los sistemas operativos: Windows y Linux, aunque se dará por entendido de ahora en adelante que todo lo que se presente va orientado a Linux específicamente en Ubuntu versión 10.10; sin embargo se mostrará los pasos que hay que seguir en Windows cuando se considere que el proceso tiene un alto grado de diferencia con el sistema operativo Linux.

Todas las versiones de Eclipse necesitan tener instalado en el sistema una máquina virtual Java (JVM), preferiblemente JRE (Java Runtime Environment) o JDK (Java Developer Kit) de Sun. Cada versión de la máquina virtual Java depende del sistema operativo en el que se va a instalar Eclipse.

Las versiones para Windows pueden descargarse de la siguiente dirección web:

<http://www.oracle.com/technetwork/java/javase/downloads/java-ee-sdk-6u3-jdk-7-downloads-439818.html>

Mientras que la descarga de la máquina virtual de Java para Ubuntu se realiza desde la terminal ingresando el siguiente comando:

```
apt-get install sun-java6-jdk
```

Eclipse presenta varias versiones, cada una con correcciones de las versiones anteriores, entre estas se encuentran:

- Eclipse Ganymede (versión 3.4)
- Eclipse Galileo (versión 3.5)
- Eclipse Helios (versión 3.6)
- Eclipse Indigo (versión 3.7)

Para concluir esta breve introducción del entorno Eclipse, se comentan algunas ventajas que tiene sobre los demás IDE, las cuales son: la disposición de un Editor de texto con resaltado de sintaxis, un compilador en tiempo real y una de las más importantes es la opción de añadir extensiones (plugins) para extender la funcionalidad.

2.1.1 Instalación de eclipse

A continuación se presentan los pasos para seguir la instalación del entorno de desarrollo en Ubuntu, la instalación en Windows es similar.

Se inicia con la descarga del instalador de Eclipse de la siguiente dirección web:

<http://www.eclipse.org/downloads/>

Se debe seleccionar, primero la versión de Eclipse, de preferencia la versión Galileo, luego escoger Eclipse IDE for Java Developers , seguido de la plataforma en la que se programará y finalmente dependiendo de la computadora escoger si es 32 o 64 bits.

Es necesario crear una carpeta donde se guarde el instalador. Cuando se finalice la descarga, se descomprime el archivo y luego se busca el ejecutable con el nombre eclipse. Se hace click sobre él para iniciar. También, puede iniciarse desde la terminal ubicándose en la carpeta donde se encuentra eclipse e ingresar el comando “./eclipse”

Al iniciar por primera vez Eclipse aparecerá una ventana que exige que se ingrese la ruta de la carpeta que se definirá como el espacio de trabajo. Esta carpeta es donde se colocaran todos los proyectos que se creen en Eclipse es donde se guardaran los ejecutables de cada proyecto, estos ejecutables son las aplicaciones que se cargarán al celular o al emulador (archivos con extensión .apk)

De preferencia el espacio de trabajo, se debe encontrar en la misma carpeta donde se extrajo el instalador para consolidar una sola carpeta donde se ubiquen todos los archivos y carpetas que tengan que ver con la programación en Eclipse.

2.2 SDK Android (kit de desarrollo de software)

El SDK de Android proporciona las herramientas y las API necesarias para empezar a desarrollar aplicaciones que se puedan ejecutar en dispositivos bajo la plataforma Android.

Este kit incluye ejemplos de proyectos con el código fuente, herramientas de desarrollo, un emulador y las librerías requeridas para la construcción de las aplicaciones.

De la misma manera que Eclipse, el SDK tiene distintas versiones para los distintos sistemas operativos, estos se pueden descargar del siguiente vínculo:

<http://developer.android.com/sdk/index.html>

Descargado el paquete para Ubuntu, el cual aparece en el sitio con el siguiente nombre Android-sdk_r15-linux.tgz, se debe continuar descomprimiendo dicho archivo en cualquier sitio, de preferencia en el mismo que se encuentre Eclipse.

2.2.1 Instalar plataformas y otros componentes

Para llevar a cabo la siguiente tarea es necesario ejecutar el SDK Manager, esta aplicación se utiliza para descargar las versiones de las plataformas de Android, documentación, herramientas, ejemplos, etc. El SDK descargado únicamente incluye las SDK Tools (depuración y pruebas de aplicaciones y otras utilidades) y para desarrollar aplicaciones es necesario disponer al menos de una API y de las SDK Platform-tools.

Existen diversas formas de poner en ejecución el SDK Manager, a continuación se detallan algunas:

- Desde Eclipse, seleccionar Window > Android SDK and AVD Manager.
- En el sistema operativo Windows, ejecutar el archivo SDK Manager.exe situado en el directorio raíz del SDK de Android.
- En Mac o Linux, desde el terminal y situado en el directorio tools del SDK de Android ejecutar `./android`

Al ejecutar cualquier de estos pasos aparece una ventana con una lista de componentes que se pueden instalar. Los obligatorios son: las plataformas de las distintas versiones de Android (API) y las SDK platform-tools, que contienen las herramientas que se requieren para el desarrollo y la depuración de aplicaciones. Mientras que los componentes recomendados son los ejemplos de códigos para las plataformas instaladas, los USB Drivers para Windows (la configuración de los drivers para Ubuntu se realiza de otra manera), documentación de las API y los Third party Add-ons, que proporcionan componentes que permiten crear un entorno de desarrollo utilizando una librería específica externa de Android (como la librería de Google Maps) [17].

Sin embargo es recomendable que se descarguen todos los componentes. Una vez instalados, estos aparecerán en el apartado Installed Packages.

Una vez descargado los paquetes, se puede revisar cuales se encuentran instalados, seleccionando Installed Packages , como se muestra en la Figura 7

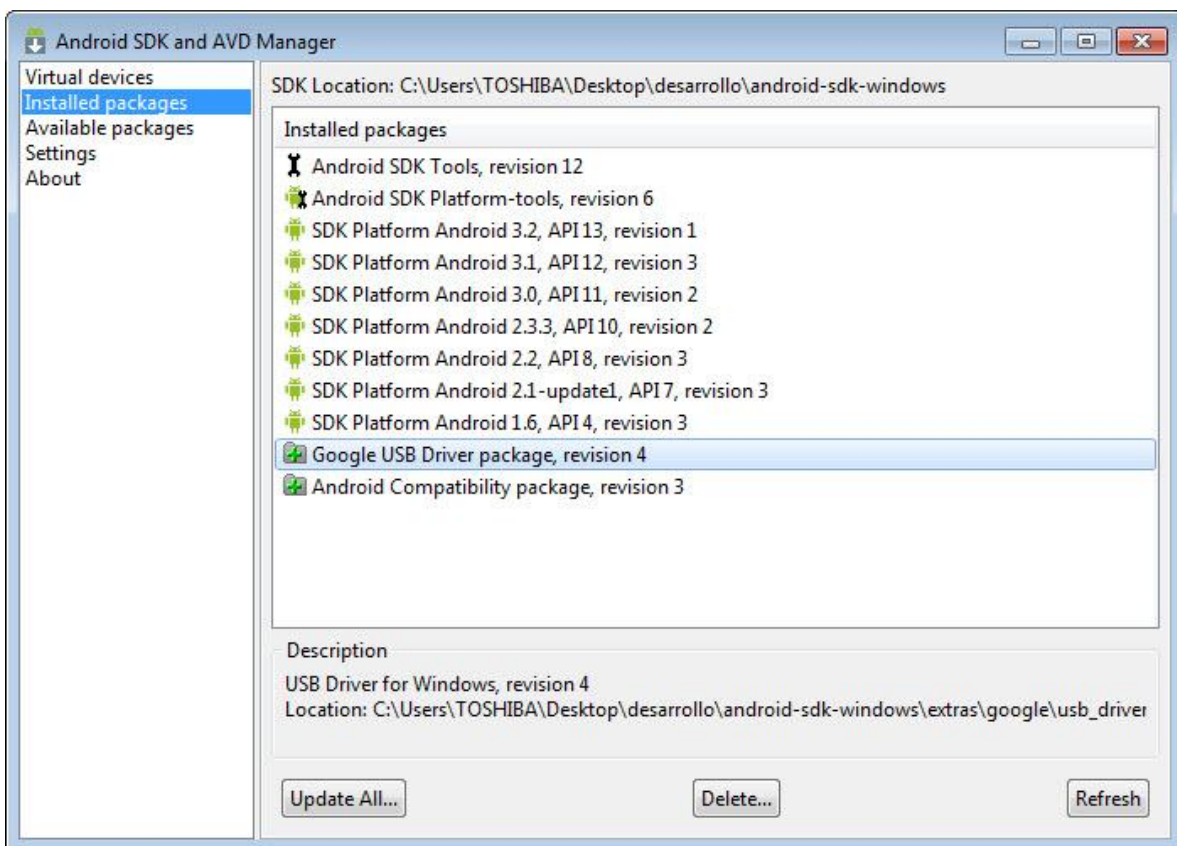


Figura 7. Android SDK Manager

2.3 Plugin ADT (Android Development Tool)

El ADT es un componente extra para el IDE de Eclipse. Está diseñado para brindar un mayor rendimiento en el entorno de las creaciones de las aplicaciones para Android.

Entre las ventajas están las que permiten ampliar las capacidades de Eclipse. Se puede configurar rápidamente nuevos proyectos en Android. Es posible crear aplicaciones con interfaz de usuario. Se pueden añadir componentes basados en el framework de la API. También se puede depurar las aplicaciones usando las herramientas del SDK y la exportación de las aplicaciones al finalizar la compilación ya sea al emulador o al propio terminal.

Es recomendable programar en Eclipse junto con el ADT ya que facilita el aprendizaje de crear aplicaciones.

2.3.1 Descarga e instalación del ADT

Después de haber instalado el SDK y el Eclipse, se procede a descargar el ADT, siguiendo los siguientes pasos:

- Iniciar Eclipse y luego seleccionar Help > Install New Software.
- Dar click en Add, en la esquina superior derecha.
- Al aparecer la ventana , ingresar en Name : ADT Plugin (puede agregarse cualquier otro nombre), y en Location , se añade la dirección del repositorio:
<https://dl-ssl.google.com/android/eclipse/>
- Click en OK.
- En la ventana Available Software, colocar el cheque junto a Developer Tools y luego click en Next.
- Se muestra una ventana en la que se podrá observar un listado de herramientas que pueden ser descargados. Click en Next.
- Luego aparecen los acuerdos de licencias, leer y aceptar.
- La instalación se completará y se deberá reiniciar Eclipse.
- Dar click en Apply y luego en OK.

2.3.2 Configuración del ADT

Ahora es el momento de enlazar el SDK con el IDE Eclipse. Por lo tanto hay que indicarle a Eclipse como llegar hasta el SDK, esto se explica a continuación.

- En Eclipse seleccionar Window > Preferences para abrir el panel de referencias.

- Seleccionar Android en el panel de la izquierda.
- Ahora en el lado derecho de SDK Location en el panel principal escoger Browse y localizar el directorio donde se encuentra el SDK.

Terminados estos pasos se tiene casi todo el ambiente para desarrollar cualquier aplicación, pero aún falta descargar otros archivos que se utilizan para la compilación de librerías nativas, muy indispensable para la compilación de la aplicación batphone.

2.4 Creación de un AVD (Dispositivo Virtual Android).

El Android SDK incluye un emulador de dispositivos móviles, que se ejecuta en la computadora. El emulador permite desarrollar y probar aplicaciones para Android sin necesidad de utilizar un dispositivo físico. Aunque se puede usar como una forma de verificar si la aplicación trabaja como se desea antes de instalarla al móvil. En la Figura 8 se muestra el dispositivo virtual emulado con la plataforma Android 2.1

El emulador imita todas las características de hardware y software de un dispositivo móvil real. Sin embargo no puede realizar llamadas de teléfono, pero las puede emular. También ofrece una variedad de teclas de navegación y control, que se puede "presionar" con el ratón o el teclado para generar eventos para cualquier aplicación. Emula también la pantalla del dispositivo con la que se podrá observar el entorno de cualquier plataforma Android con la que se levante cualquier dispositivo virtual.

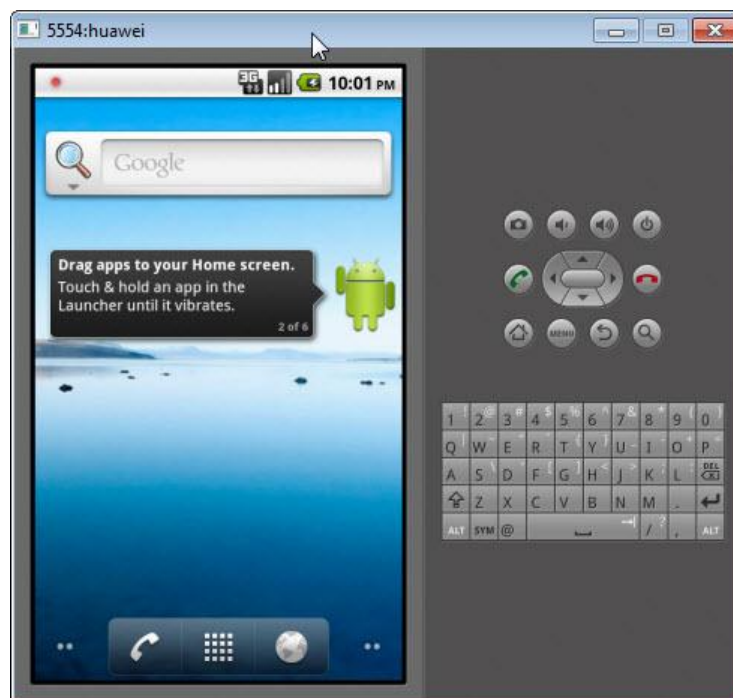


Figura 8. Emulador corriendo con plataforma Android 2.1

La imagen del sistema Android distribuido en el SDK contiene el código de un procesador ARM con el kernel Linux que es la base del emulador, las bibliotecas nativas, la máquina virtual Dalvik, y los diversos paquetes de Android.

El emulador permite definir ciertos aspectos del hardware del teléfono a emular y permite crear muchas configuraciones para poner a prueba muchas plataformas Android y sus características de hardware. También incluye una variedad de capacidades de depuración, tales como una consola desde la que se puede registrar la salida del núcleo, simular interrupciones (conocidos como Intents, por ejemplo: al recibir mensajes SMS o llamadas telefónicas), y simular los efectos de la latencia.

Entre el hardware que se puede emular se cuenta:

- Una CPU ARMv5.
- Una pantalla LCD de 16 bit.
- Uno o más teclados (un teclado Qwerty)
- Un chip de sonido con capacidad de entrada y salida.
- Particiones de memoria flash.
- Un módem GSM, incluyendo una simulación de la tarjeta SIM.

Por lo tanto para usar el emulador es necesario crear uno o más AVD (Dispositivo Virtual Android). En cada configuración, se especifica que plataforma Android se desea ejecutar en el emulador y el conjunto de opciones de hardware que se desee utilizar. Los pasos para crear un AVD se detallan a continuación:

- En Eclipse, seleccionar Window , luego Android SDK and AVD Manager (también se puede iniciar al correr el ejecutable SDK Manager que se encuentra en la carpeta raíz del SDK).
- Seleccionar Virtual Devices en el panel izquierdo.
- Click en New y aparece el cuadro de dialogo Create New AVD
- Digitar el nombre del AVD, por ejemplo “mi_AVD”.
- Escoger una tarjeta (La tarjeta es la versión de la plataforma de Android que se desea que ejecutar en el emulador).
- Click Create AVD.

Luego, para iniciar el emulador, se debe seleccionar la opción Start de la ventana Android SDK and AVD Manager o también desde la terminal ingresando los siguientes comandos:

```
Emulator -avd <nombre de avd> [-opciones]
```

Existen muchas opciones que se utilizan para arrancar el emulador con el fin de configurar su apariencia o comportamiento. Algunas de estas opciones se muestran en la Tabla 5.

CATEGORIA	OPCION	DESCRIPCION
Ayuda	-help	Muestra un listado de todas las opciones del emulador
	-help-<opción>	Muestra la ayuda para una opción específica
	-help-virtual-device	Muestra ayuda para el AVD a usar
	-help-disk-images	Muestra ayuda para el uso de imágenes de disco
Imagen de disco	-sdcard <ruta de archivo>	Usa la imagen de disco como la SD card del móvil
	-nocache	Inicia el emulador sin una partición cache
	-ramdisk <ruta de archivo>	Usa la imagen de disco como memoria ram del móvil
Depurador	-shell	Crea una consola "root" en el actual terminal
	-show-kernel <nombre>	Muestra los mensajes del kernel
Media	-radio <dispositivo>	Redirecciona la interfaz radio modem a un host de carácter dispositivo
	-noaudio	Deshabilita el soporte de audio en el actual emulador
	-useaudio	Habilita el soporte de audio en el actual emulador
Redes	-dns-server <servers>	Usa el servidor DNS especificado
	-http-proxy <proxy>	Realiza todas las conexiones TCP a través de un proxy HTTP/HTTPS especificado
Interfaz de usuario	-dpi-device <dpi>	Escala la resolución de el emulador al tamaño de la pantalla de un dispositivo real.
	-no-boot-anim	Deshabilita la animación de arranque durante el inicio del emulador

Tabla 5. Opciones con las que puede arrancar el emulador [18].

2.5 NDK (Native Development Tool).

El NDK de Android es una herramienta complementaria para el SDK de Android. Permite construir aplicaciones con porciones de código nativo. Es decir proporciona las cabeceras y las librerías que permiten crear actividades, gestionar las entradas del usuario, el uso de los sensores del hardware, acceso a recursos de aplicación, trabajar con los drivers del hardware del móvil, entre otras cosas. Trabajar con estos recursos se requiere el conocimiento de programación en C o C++. Si se escribe en código nativo, las aplicaciones siguen teniendo la extensión .apk y aún se ejecutan dentro de la máquina virtual del dispositivo. La estructura fundamental de la aplicación Android no cambia.

Cabe destacar que al utilizar el NDK hay un incremento en el rendimiento de la aplicación, pero también incrementa la complejidad de ésta.

Como la mayoría de software que se ha presentado anteriormente, el NDK también tiene las respectivas versiones para los diferentes sistemas operativos, estos se pueden descargar de la siguiente dirección:

<http://developer.android.com/sdk/ndk/index.html>

Luego de la descarga del paquete android-ndk-r6b-linux-x86.tar.bz2., se debe descomprimir y de preferencia, colocar el NDK en el mismo directorio en el que se encuentra el SDK. Ahora es necesario complementar los requerimientos para la correcta ejecución del NDK.

2.5.1 Descargar e instalar Cygwin 1.7 o mayor

Este paso es solamente necesario para Windows, ya que este programa es un emulador de la terminal de Linux, que permite ejecutar los comandos que se utilizan en Ubuntu evitándose la instalación del mismo en la computadora. El instalador se puede descargar de la siguiente dirección:

<http://cygwin.com/install.html>

Se escoge la opción setup.exe. Descargado el ejecutable, dar click sobre él y seguir los pasos de instalación.

2.5.2. Descarga e instalación de GNU Make 3.81

Este complemento es el encargado de realizar la compilación de los archivos que se encuentran indicados dentro del Makefile.

- **En Windows**

Ejecutar una vez más el instalador del cygwin y seguir los pasos de instalación hasta la página Select Packages. Ahora dentro de los paquetes buscar la opción Devel y desplegar la lista en el signo +.

Dentro de la lista hay que buscar los paquetes y marcarlos para indicar la descarga, los cuales son el gcc-core: el compilador C, gcc-g++: el compilador C++, gdb: el depurador GNU y make: la versión GNU de la utilidad "make". Luego que se han indicado, escoger siguiente y esperar la instalación.

Para confirmarla versión de cada uno de estos paquetes hay que ejecutar el cygwin y en la consola ingresar los siguientes comandos:

```
gcc --version
g++ --version
make --version
gdb --version
```

- **En Ubuntu**

La instalación del gnu make en Ubuntu viene con los compiladores gcc y g++, así que solamente se debe descargar el paquete build-essential, desde la terminal:

```
sudo apt-get install build-essential
```

Instalados los paquetes antes mencionados ya se tiene el entorno de desarrollo listo para comenzar a programar en Android; sin embargo es necesario conocer la estructura de un proyecto en Eclipse.

2.6 Composición de un proyecto Android en eclipse.

Cuando se crea un proyecto Android en el entorno de desarrollo Eclipse se genera automáticamente la estructura de carpetas necesarias para poder generar posteriormente la aplicación. En la Figura 9 se muestra la estructura de un proyecto. Esta estructura será común para cualquier aplicación, independientemente de su tamaño y complejidad.

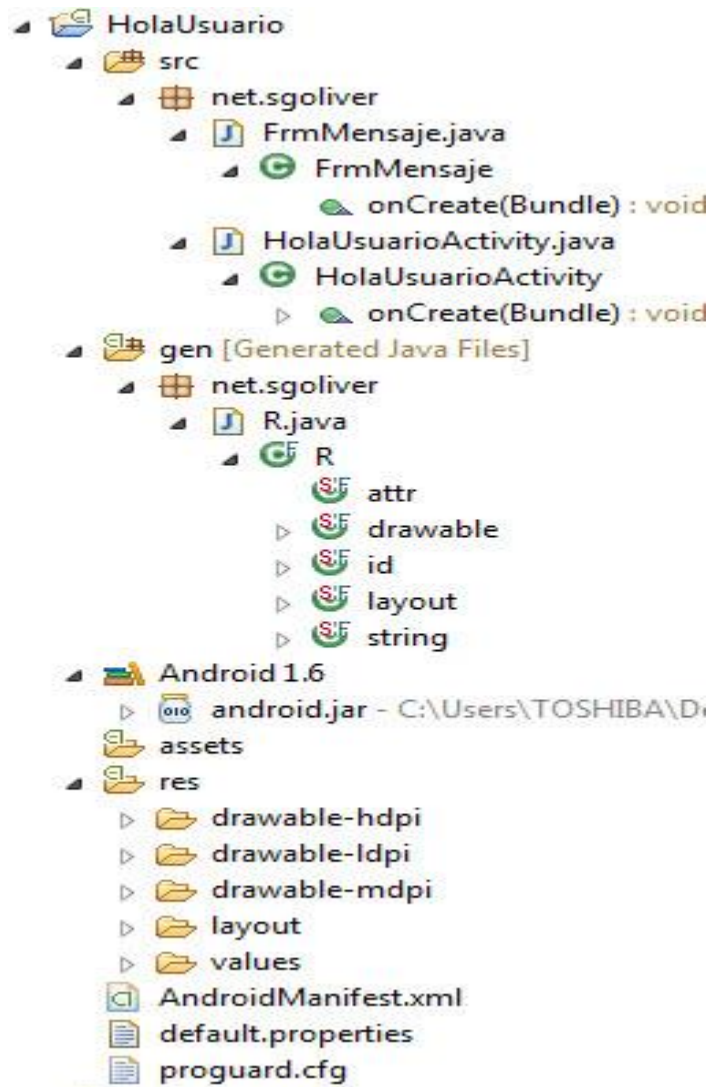


Figura 9. Estructura de un proyecto en Eclipse

La composición de un proyecto en Eclipse es:

- **Carpeta src**

Contiene todo el código fuente de la aplicación, código de la interfaz gráfica, clases auxiliares, etc. Cada clase en java pertenece a un paquete, además un paquete puede contener a su vez otros paquetes.

Al momento de crear la aplicación se debe especificar el nombre del paquete en el que se incluyen de forma predeterminada las clases que se irán utilizando.

En la figura 9, puede observarse que dentro de la carpeta src existen dos archivos java y cada uno de ellos contiene una clase. Para observar el código simplemente se da click sobre el archivo en interés.

- **Carpeta res**

Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc. Los diferentes tipos de recursos se deberán distribuir entre las siguientes carpetas:

- **++ /res/drawable/.**

En esta directorio se colocan las imágenes que se utilizarán, desde las que conformaran todo el entorno gráfico de la aplicación hasta la imagen que representara el ejecutable. Esta carpeta cuenta con tres subdirectorios: /drawable-ldpi, /drawable-mdpi y /drawable-hdpi en el que se guardarán las imágenes dependiendo de la densidad de puntos por pulgada que tenga el dispositivo en el que se ejecute la aplicación.

- **++ /res/layout/.**

Contienen los ficheros de definición de las diferentes pantallas de la interfaz gráfica. Desde el código de la aplicación se le indicará el archivo .xml que se quiere usar en cada momento, de esta forma se facilita modificar el código sin tener que tocar nada del entorno gráfico y viceversa.

- **++ /res/anim/.**

Contiene la definición de las animaciones utilizadas por la aplicación.

- **++ /res/menu/.**

Contiene la definición de los menús de la aplicación.

- **++ /res/values/.**

Contiene otros recursos de la aplicación como por ejemplo cadenas de texto (strings.xml), estilos (styles.xml), colores (colors.xml), etc. Se sigue la misma ideología para facilitar la modificación de código.

- **++ /res/xml/.**

Contiene los ficheros XML utilizados por la aplicación.

++/res/raw/.

Contiene recursos adicionales, normalmente en formato distinto a XML, que no se incluyan en el resto de carpetas de recursos.

Todos los archivos colocados en la carpeta res, son accesibles desde la clase R

- **Carpeta gen**

Contiene una serie de elementos de código generados por Eclipse automáticamente al compilar el proyecto. Cada vez que se genera el proyecto, la máquina virtual de Android genera una serie de ficheros fuente en java dirigido al control de los recursos de la aplicación.

El fichero más importante es el R.java, y la clase R. Esta clase R contendrá en todo momento una serie de constantes con los ID de todos los recursos de la aplicación incluidos en la carpeta /res/, de forma que se pueda acceder fácilmente a estos recursos desde el código a través de este dato. Así, por ejemplo, la constante R.drawable.icon contendrá el ID de la imagen "icon.png" contenida en la carpeta /res/drawable. Por lo tanto es recomendable que no se modifique este fichero.

Estos archivos se deben editar, ya que será Eclipse el que se encargará de modificarlos automáticamente según otros parámetros del proyecto.

- **Carpeta assets**

Contiene todos los demás ficheros auxiliares necesarios para la aplicación, como por ejemplo ficheros de configuración, de datos, etc.

La diferencia entre los recursos incluidos en la carpeta /res/raw/ y los incluidos en la carpeta /assets/ es que para los primeros se generará un ID en la clase R y se deberá acceder a ellos con los diferentes métodos de acceso a recursos. Para los segundos sin embargo no se generarán ID y se podrá acceder a ellos por su ruta como a cualquier otro fichero del sistema, para eso se debe usar la clase *AssetManager* para leer los datos como un stream3 de bytes.

Puede observarse que el método es mucho más sencillo utilizando la carpeta res; sin embargo escoger cual se usará dependerá según las necesidades de la aplicación.

- **Fichero AndroidManifest.xml**

Contiene la definición en XML de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, versión, icono, etc), sus componentes (pantallas, mensajes, etc), o los permisos necesarios para su ejecución. Por lo tanto se le puede definir como el archivo global de configuración, en éste archivo se puede definir qué es lo que puede hacer la aplicación, por ejemplo: informar al sistema que la aplicación puede

abrir archivos mp3, abrir enlaces http o que es capaz de manejar las llamadas telefónicas que se reciban. También se indica las actividades o servicios que ejecutará la aplicación y los permisos especiales que necesita utilizar en el caso de que se quiera acceder a recursos compartidos del sistema, como el acceso a la lista de contactos, uso del GPS, o la posibilidad de enviar mensajes SMS.

- **/Android x.x.x:**

Es la librería de desarrollo de Android (SDK). Dentro de esta carpeta se encuentran los paquetes y clases que incluye el SDK . Es decir todas las clases que se utilicen en el código java se deben encontrar en este directorio.

- **Archivo default.properties**

Es un fichero que genera automáticamente Eclipse, solamente contiene las propiedades del proyecto.

2.6.1 Creación de un proyecto en eclipse

Luego de comprender la estructura de un proyecto y de tener el conocimiento para crear un AVD, se explicarán los pasos a seguir para la creación de un proyecto Android desde Eclipse. Estos pasos son los siguientes:

- En Eclipse, seleccionar File > New > Project.
- Seleccionar Android Project y dar click en Next.
- Aparecerá una ventana como la figura 10, en el cual hay que rellenar los detalles del proyecto colocando los valores en las respectivas celdas.
- Click en Finish.

Ahora se puede visualizar la estructura del proyecto recién creado, a través del explorador de paquete. Al escoger esta opción aparecerá una ventana en el lado izquierdo del escritorio de Eclipse y mostrará todos los proyectos que se encuentran guardado en el espacio de trabajo.

Los proyectos aparecen como un árbol de carpetas (como la figura 9), de los cuales se puede ir modificando el código, agregando clases, entre otras cosas según la necesidad de la aplicación.

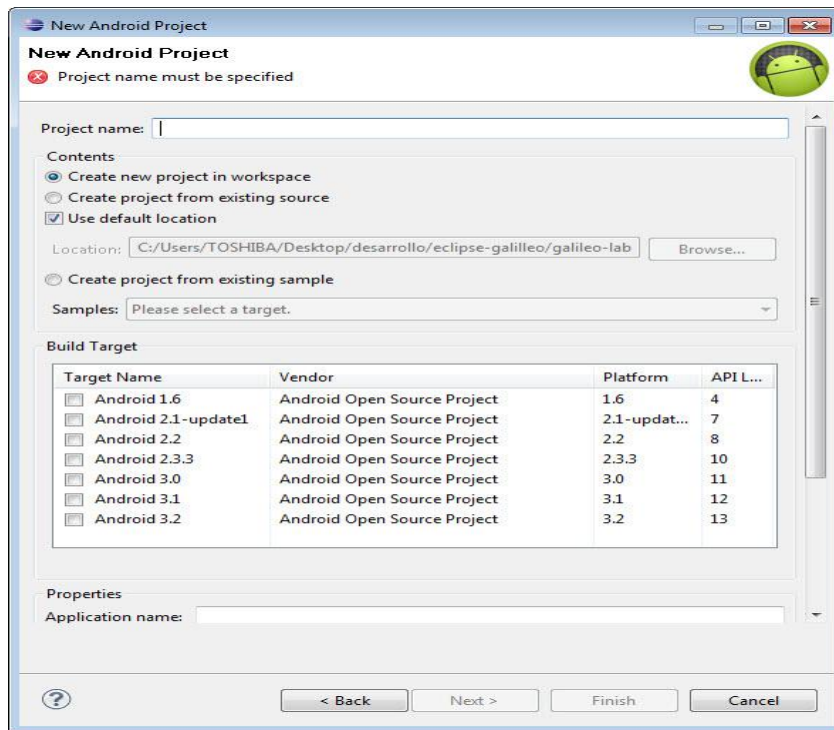


Figura 10. Creación de nuevo proyecto en Eclipse

2.6.1.1 Detalles para la creación de un proyecto.

Para tener una mejor idea sobre los datos que se ingresan al momento de crear un proyecto en Android se explican brevemente cada uno de los campos.

- **Project Name:** Este es el nombre del proyecto en Eclipse. Es el nombre de la carpeta que contiene los archivos del proyecto.
- **Build Target:** Es la versión del Android SDK que se usará para construir la aplicación. Por ejemplo, si se escoge el Android 2.1, la aplicación será compilada con la biblioteca de la plataforma Android 2.1. La tarjeta que se escoja no es necesario que coincida con la tarjeta del AVD o del dispositivo móvil, sin embargo la idea es que la tarjeta debe ser igual o menor que la tarjeta del AVD o de la versión del dispositivo móvil.

Otra forma de explicarlo es que si una aplicación que se compiló para una versión de Android 2.1 podrá ejecutarse en un dispositivo que tenga la versión de Android 2.2 o 2.3, pero no funcionará en un dispositivo con una plataforma Android 2.0.

- **Application Name:** Es el título de la aplicación que aparecerá en el escritorio del dispositivo.
- **Package Name:** Se ingresa el nombre del paquete que tendrá las clases. La asignación del nombre sigue las mismas reglas que se utilizan en la programación Java.

Este nombre debe ser único entre todos los paquetes instalados en el sistema Android, por esta razón es importante que el nombre lleve un dominio. Por ejemplo: com.ejemplo

- **Create Activity:** Esta es una subclase de la clase Activity perteneciente a Android. Una actividad es simplemente una clase que puede ejecutar y hacer cualquier trabajo.

Si se selecciona crear la actividad, el código que aparecerá por defecto en la clase Activity, será el que Eclipse asigna para crear su ejemplo de inicio Hello World. Sin embargo puede borrarse el código e iniciar desde cero para otra aplicación.

- **Min SDK Version:** Este valor especifica el nivel mínimo de API en el cual la aplicación se ejecutará. El Min SDK versión debe ser el mismo que la tarjeta en la cual se correrá la aplicación. Por ejemplo, si la tarjeta es Android 2.1, entonces el Min SDK versión debe ser 7 o menor.

2.7 Estructura de una aplicación en Android.

A continuación se explican los bloques principales que conforman una aplicación Android, Se advierte que una aplicación no necesariamente necesitará cada uno de ellos, pero si la combinación de algunos:

- **Activity (Actividad)**

Representa cada una de las principales tareas que el usuario puede llevar a cabo en la aplicación. Para hacer uso de ella, se necesita que la clase herede la superclase Activity.

La clase podrá mostrar una interfaz de usuario compuesta por Vistas (Views) y podrá reaccionar a los eventos del usuario (presionar un botón, escribir en un campo de texto etc). Así pues, cada pantalla en Android está implementada como Activity. La navegación entre pantallas se lleva a cabo mediante el arranque de una nueva Activity (generalmente mediante el método `startActivity(myintent)`), y existe un vínculo de comunicación entre pantallas de modo que la pantalla generada puede devolver valores a la pantalla que la creó para de esta forma mantener el buen funcionamiento de la aplicación.

- **Intents (Receptores y gestores de eventos)**

Para terminar de comprender como se relacionan varias pantallas en una aplicación es necesario comprender que el Intent es una clase especial; que describe qué es lo que desea hacer la aplicación en ese momento.

La estructura típica de un Intent es: la acción que desea realizar (la intención) y los datos sobre los que actúa. Existe una clase relacionada llamada `IntentFilter`; la cual es una descripción de lo que demanda la aplicación mientras que el intent es la petición de realizar una acción.

IntentReceiver es usado para ejecutar código en reacción a un evento externo (puede observarse que se encuentra ligado a Intent), como por ejemplo, cuando el teléfono suena, etc. Este tipo de componentes tiene que estar registrado en el fichero AndroidManifest.xml o como alternativa mediante código haciendo uso del método Context.registerReceiver().

Por lo tanto un intent es el elemento básico de comunicación entre los distintos componentes Android que se han descrito anteriormente. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un intent se puede mostrar una actividad desde cualquier otra actividad, iniciar un servicio, enviar un mensaje broadcast, iniciar otra aplicación, etc.

- **Service**

Comúnmente, un servicio es código que se ejecuta en segundo plano y no necesita de una interfaz de usuario en su ciclo de vida. Otro componente de la aplicación podría iniciar un servicio y este continuará corriendo en un segundo plano, incluso si el usuario cambia a otra aplicación

- **ContentProvider**

Este componente es usado para compartir datos entre diferentes aplicaciones. Mediante estos componentes es posible compartir determinados datos de la aplicación sin mostrar detalles sobre su almacenamiento interno, su estructura, o su implementación. De la misma forma la aplicación creada podrá acceder a los datos de otra a través de los content provider que se hayan definido.

Android viene con una serie de content providers para tipos de datos comunes (audio, video, imágenes, información de contactos, entre otros).

- **View**

Los objetos view son los componentes básicos con los que se construye la interfaz gráfica de la aplicación. Son el equivalente a los controles de Java o .NET. Al momento de iniciar, Android pone a disposición una gran cantidad de controles básicos, como cuadros de texto, botones, listas desplegables o imágenes. También existe la posibilidad de extender la funcionalidad de estos controles básicos o crear propios controles personalizados.

- **Broadcast Receiver**

Un broadcast receiver es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema (por ejemplo: "Batería baja", "SMS recibido", "Tarjeta SD insertada", etc) o por otras aplicaciones (cualquier aplicación puede generar mensajes intents, en terminología Android) broadcast, es decir, no dirigidos a una aplicación concreta sino a cualquiera que quiera escucharlo

2.8 Compilación e instalación de una aplicación.

Después de finalizada una aplicación debe ser compilada. Luego, se debe comprobar su funcionamiento ya sea en el emulador o en el dispositivo móvil.

Si se compila la aplicación en Eclipse, este etiqueta el archivo con extensión .apk con una clave de depuración por defecto. De manera inmediata se llama al ADB que instala la aplicación en un emulador o en un dispositivo que esté conectado a la computadora. Las tareas de compilado e instalación no se realizan de manera manual. Aunque si se pueden realizar no se tratará dicho tema en el presente trabajo.

2.8.1 Corriendo aplicación en el emulador

Antes de que se corra la aplicación en el emulador, es necesario haber creado un AVD.

Al correr la aplicación desde Eclipse, escoger el proyecto en el explorador de paquetes y dar clic derecho y luego escoger la opción Run As > Android Application.

El complemento ADT creará automáticamente por defecto una configuración de ejecución para el proyecto. En ese momento Eclipse llevará acabo lo siguiente:

- Compila el proyecto (si ha habido cambios desde la última compilación)
- Crea una configuración de ejecución por defecto (sino existía ninguna para el proyecto)
- Instala y ejecuta la aplicación en un emulador o dispositivo.

Al mostrarse el AVD, normalmente este ya aparece con la aplicación que se instaló ejecutándose, de esta forma se podrá comprobar si la aplicación cumple con las expectativas.

2.8.2 Corriendo aplicación en dispositivo móvil

Antes de instalar la aplicación en un dispositivo, es necesario realizar alguna configuración básica del dispositivo:

- Habilitar USB Debugging en el dispositivo. En la mayoría de dispositivos Android esta opción puede encontrarse al ir a Settings > Applications > Development > USB Debugging.
- Asegurarse que la computadora puede detectar el dispositivo cuando esté conectado vía USB

Una vez el dispositivo está configurado de manera adecuada, conectado a la computadora vía USB, hay que navegar a la carpeta platform-tools de la carpeta SDK e instalar el .apk al dispositivo.

```
adb install ruta/a/la/aplicación.apk
```

Finalizada la transferencia de la aplicación, se debe buscar en el escritorio del dispositivo la aplicación instalada.

2.9 Instalación de los drivers USB.

La instalación de los drivers USB depende del sistema operativo. Éstos son muy indispensables para lograr instalar las aplicaciones al móvil, tomar el control del teléfono como superusuario, realizar depuraciones de la aplicación desde el móvil, etc.

El proceso de instalación en Windows viene incluido cuando se realiza la descarga de todos los paquetes desde el SDK Manager, tema que se abordó en el enunciado 2.2.1. Sin embargo este proceso es exclusivo para Windows; mientras que en Ubuntu es necesario realizar los siguientes pasos:

- Ingresar los siguientes comandos en la consola

```
cd /etc/udev/rules.d
sudo touch 70-android.rules
sudo gedit 70-android.rulescd /etc/udev/rules.d
```

- En este archivo se debe ingresar, lo siguiente
SUBSYSTEM=="usb", ATTR{idVendor}=="0bb4", MODE="0666"
- El idVendor depende del fabricante del dispositivo , por lo tanto para el fabricante HTC su idVendor es 0BB4 y el de Huawei es 12D1. En el Anexo B se muestran los idVendor para los diversos fabricantes.
- Luego ingresar los siguientes comandos

```
sudo chmod a+rx /etc/udev/rules.d/70-android.rules
sudo udevadm control -reload-rules
```

Luego de esto se deberá de comprobar si el teléfono es detectado con el comando adb devices.

2.10 ADB (Android Debug Bridge).

El ADB es una herramienta de línea de comandos indispensable del SDK. Permite una comunicación con el emulador o un dispositivo Android conectado a la computadora. Es un programa cliente servidor, que incluye tres componentes:

- Un cliente, que se ejecuta en la computadora. Se invoca al cliente desde una consola mediante el comando adb.
- Un servidor, que se ejecuta en un segundo plano en la computadora. El servidor gestiona la comunicación entre el cliente y el demonio del adb corriendo en un emulador o dispositivo.
- Un demonio, que se ejecuta en un segundo plano en cada emulador o dispositivo.

El ADB se encuentra en la carpeta platform-tools de la carpeta del SDK.

Al iniciar un cliente ADB, el cliente comprueba primero si hay un proceso de servidor adb que se esté ejecutando. Si no hay, se inicia el proceso del servidor. Al iniciar el servidor, éste se une al puerto local TCP 5037 y se detiene a escuchar, en espera de los comandos enviados desde los clientes (todos los clientes ADB usan el puerto 5037 para comunicarse con el servidor ADB).

2.10.1 Detectar dispositivo o emulador

Antes de emitir órdenes con el adb, es útil saber si hay un emulador o un dispositivo conectados al servidor adb. Para eso es necesario utilizar el siguiente comando:

```
adb devices
```

Como respuesta el ADB puede mostrar la siguiente información:

- **Serial:** Una serie de dígitos y caracteres creados por el ADB para identificar un emulador o un dispositivo.
- **Estado:** Muestra el estado de la conexión, si ha detectado un dispositivo o un emulador muestra “device” , si el adb no está funcionando o tiene algún problema mostrará “offline” y si no se estuviera ejecutando ningún emulador o no estuviera conectado un dispositivo en la consola se mostrará “no device” (también es posible que no muestre ningún comando).

Si hay varios emuladores ejecutándose y también algunos teléfonos móviles conectados, al utilizar el comando anterior, se mostrará en la consola la lista de dispositivos y emuladores.

Se puede especificar con que dispositivo se quiere realizar una acción, solamente con especificar el parámetro -s y el número de serie del dispositivo, seguido por el comando u operación a realizar:

```
adb -s [serial] [comando]
```

2.10.2 Instalando una aplicación

Se puede utilizar el adb para copiar una aplicación desde la computadora e instalarlo en un emulador o dispositivo. Para hacerlo, se debe utilizar el comando de instalación desde la línea de comandos en Windows o desde la consola en Ubuntu. Con el comando, se debe especificar la ruta al archivo .apk que se desea instalar:

```
adb install <ruta del .apk>
```

2.10.3 Otros comandos con ADB

La herramienta ADB que viene incluido en el paquete del SDK, tiene algunas funciones que se utilizan para la configuración del dispositivo móvil. Sin embargo profundizar cada una de ellas no es la finalidad de este reporte, por lo tanto se muestra la tabla 6, en la que se presenta un breve resumen de cada una de ellas:

Categoría	Comando	Descripción
Opciones	-d	Direcciona un comando del adb a un dispositivo en específico que se encuentre conectado con otros al puerto USB.
	-e	Direcciona un comando del adb a un emulador en específico que se esté ejecutando al mismo tiempo con otros emuladores
	-s <Serial>	Direcciona un comando del adb a un dispositivo/emulador en específico.
General	devices	Muestra una lista de todos los emuladores o dispositivos detectados
	help	Muestra una lista de todos los comandos del adb
	version	Muestra el número de versión del adb
Depuración	Logcat [opciones] [filtros]	Muestra los datos del archivo log en pantalla
	bugreport	Muestra toda la información del móvil como un reporte de depuración
	jdwp	Muestra una lista de procesos JDWP disponibles en un dispositivo
Datos	Install <ruta .apk>	Instala una aplicación .apk en un dispositivo o emulador
	pull <remoto> <local>	Copia un archivo específico desde un emulador o dispositivo a un lugar específico de la computadora
	push <local> <remoto>	Copia un archivo específico desde la computadora a una carpeta específica del emulador o dispositivo
Puertos y redes	forward <local> <remoto>	Permite realizar conexiones desde un puerto local específico a un puerto remoto del emulador o dispositivo
	ppp <tty> [parámetro]	Corre PPP sobre USB
informativos	get-serialno	Muestra el número de serie
	get-state	Muestra el estado de adb
	wait-for-device	Indicarle al adb que debe detenerse hasta que detecte un terminal o emulador
Servidor adb	start-server	Iniciar el servidor adb
	kill-server	Matar el servidor adb
Consola	shell	Accede a la consola del terminal
	shell [comandos]	Ejecuta comandos en la consola del terminal

Tablas 6. Opciones de ADB [19].

2.11 Herramienta DDMS.

Android viene con una herramienta de depuración denominada Dalvik Debug Monitor Server (DDMS) cuya pantalla principal se muestra en la Figura 11. El funcionamiento de esta herramienta es proporcionar servicios de redireccionamiento de puertos, captura de pantalla en el dispositivo, captura y listado de información en el dispositivo, procesos e información del estado del radio, simular llamadas entrantes o SMS, e incluso simular una posición de ubicación.

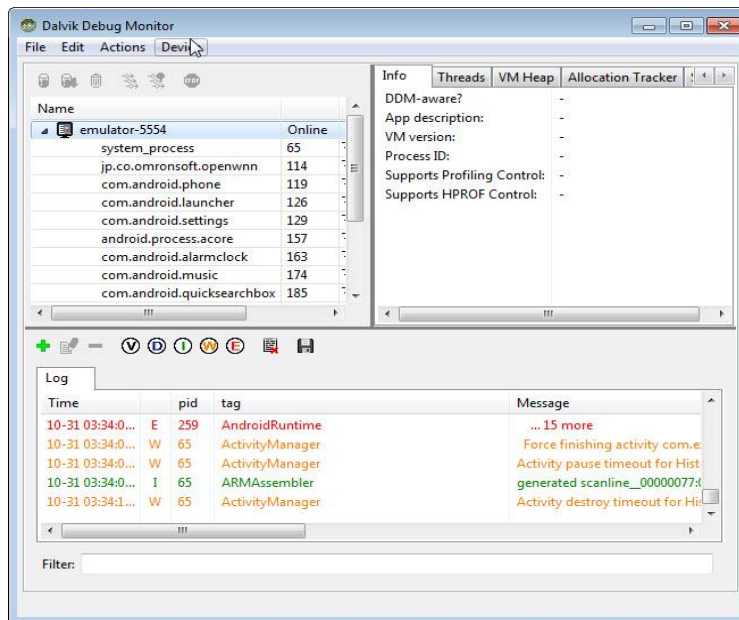


Figura 11. Entorno del DDMS

El DDMS se puede encontrar en el directorio tools/ del SDK. Este puede ejecutarse desde la línea de comandos. Trabaja con el emulador del SDK o un dispositivo conectado, si ambos están conectados y en funcionamiento al mismo tiempo, por defecto DDMS usará el emulador.

DDMS actúa como un intermediario para conectar el entorno de desarrollo Eclipse, con las aplicaciones que se ejecutan en el dispositivo. En Android, cada aplicación se ejecuta en su propio proceso, y cada uno de ellos cuenta con su propia máquina virtual (VM).

En el lado izquierdo del monitor de depuración se muestra cada emulador o dispositivo conectado, con una lista de todas las máquinas virtuales en ejecución dentro de cada uno. Las máquinas virtuales se identifican por el nombre del paquete de la aplicación que aloja, y se puede utilizar esta lista para encontrar y conectar con la máquina virtual de la actividad que se desea depurar. En la Figura 12 se puede observar las maquinas virtuales para distintas actividades.

Name	Online	huawei [2.2, debug]
emulator-5554	Online	huawei [2.2, debug]
system_process	67	8600
jp.co.omronsoft.openwnn	117	8601
com.android.phone	122	8602
com.android.launcher	126	8603
com.android.settings	132	8604
android.process.acore	159	8605
com.android.alarmclock	165	8606
com.android.music	177	8607
com.android.quicksearchbox	188	8608
com.android.protips	197	8609
android.process.media	204	8610
com.android.mms	214	8611
com.android.email	231	8612
org.sipdroid.sipua	247	8613

Figura 12. Maquinas virtuales para diversas aplicaciones

En la columna de la derecha se muestran los puertos a los que están conectadas cada máquina virtual y también el identificador de proceso de Linux al lado del nombre del paquete. Cada proceso Linux tiene su propia identificación o PID.

Dentro del entorno de DDMS se encuentra el panel de monitoreo de depuración, que en la figura 11 puede observar en el cuadro derecho superior. En la pestaña Control de Emulador se puede simular estados especiales o actividades de los dispositivos.

Las características de esta pestaña incluyen:

- **Telephony status:** Cambiar el estado de los planes de voz y datos del teléfono (home, roaming, searching, etc.), y simular diferentes tipos de velocidad y latencia de la red (GPRS, EDGE, UMTS, etc.)
- **Telephony actions:** Simular llamadas telefónicas o mensajes SMS en el emulador.
- **Location controls:** Enviar datos simulados de ubicación al emulador, por ejemplo para poder realizar operaciones de localización en cartografía GPS.

Por último en la sección inferior de la pantalla del DDMS, se encuentra los logs y outputs que son, los paneles más importantes. Aquí se puede realizar la depuración y seguimiento de tareas, en caso de que se tengan problemas con la aplicación que se está desarrollando.

En esta pestaña, se tienen todos los mensajes del dispositivo, la dinámica para detectar el error es ver mensajes de excepciones y mensajes de indicación que se dejen en el código de la aplicación para ver hasta dónde ejecuta.

2.12 Conclusión.

Es muy importante conocer cuáles son los requerimientos del sistema operativo en el que se va a instalar el entorno de desarrollo. De esta manera se puede tener una idea ante cualquier problema que se presente al momento de programar e instalar complementos.

Para crear una aplicación en Eclipse, se debe de contar con un conocimiento básico del lenguaje de programación Java. Como pudo observarse existen muchos términos que solo una persona con conocimientos básicos puede comprender.

El conocer las funciones y características de cada uno de los software que complementan el entorno de desarrollo, así como las herramientas que presenta el SDK de Android, es una enorme ventaja, ya que el buen conocimiento y manejo de estos permite aprovechar al máximo la funcionalidad del entorno de desarrollo y aumentar las posibilidades de encontrar rápidas soluciones ante problemas planteados.

Capítulo 3: La aplicación Batphone.

En este capítulo se explicará el funcionamiento de la aplicación batphone. También se explicaran las aplicaciones que la conforman y la función que realizan cada una de ellas. De manera breve se muestra el contenido de cada una de los archivos .apk de cada aplicación.

Se brindan los pasos a seguir para que todo dispositivo móvil pueda soportar las aplicaciones necesarias para la configuración batphone.

Se muestran también los ajustes que se deben realizar en cada dispositivo cuando se tienen instaladas las aplicaciones Serval Batphone y Patched Siproduct, para que se pueda realizar la comunicación entre celulares.

3.1 Serval Project.

El proyecto Serval considera que la comunicación no debe ser utilizada para enriquecerse, más bien debe de ser gratuita. Consideran que la comunicación es un derecho de toda persona. Apoyándose en este pensamiento desarrollan la aplicación para dispositivos móviles batphone. Ésta tiene por objetivo lograr una comunicación de manera gratuita, sin la dependencia de utilizar infraestructuras, torres telefónicas, satélites y sin la presencia de operadores. Uno de los desarrolladores de esta aplicación fue Paul Gardner Stephen, que junto con un grupo de programadores, estudiantes y voluntarios continúan trabajos para mejorar la aplicación batphone [20].

El nombre Serval surge por dos motivos. Primero si se pronuncia despacio se puede escuchar uno de los objetivos que pretende alcanzar el proyecto Serve All (servicio para todos). El segundo es en honor a un gato salvaje de África con el nombre Serval, que tiene dentro de sus características orejas grandes con un oído muy agudo, permitiendo detectar las frecuencias ultrasónicas emitidas por roedores y pequeños animales.

La aplicación batphone se compone de aplicaciones ya existentes, que tenían otra finalidad. Las cuales al enlazar sus funciones y modificar un poco sus códigos presentan la herramienta indicada para lograr el objetivo de la comunidad Serval.

A pesar del logro obtenido la comunidad Serval tiene aun muchos errores por corregir. Sin embargo tiene bastantes expectativas y funciones para agregar al batphone, entre las cuales están:

- Funcionalidad de SMS.
- Mapeo y reporte de incidentes.
- Servicios de video.
- Adaptación a los sistemas Symbian e iOS.

3.2 Batphone.

El objetivo principal de la aplicación Batphone es llevar comunicación a los lugares donde no existe una infraestructura. También está orientado a ofrecer servicio telefónico en casos que suceda un desastre natural, asumiendo que toda estructura telefónica desaparezca.

Por lo tanto, se conocerá como Batphone a todo aquel teléfono móvil que pueda establecer una comunicación con cualquier otro dispositivo sin utilizar una infraestructura telefónica. Utilizando únicamente como infraestructura su propio hardware y como medio las bandas del espectro conocidas como Wi-fi y que soporte la incorporación del protocolo de enrutamiento BATMAN.

Todo teléfono configurado como batphone trabaja como una pequeña central telefónica. Éste se enlaza con otros batphones que se encuentran en sus cercanías (wifi tether), comunicándose con ellos y sirviendo como un nexo para comunicar otros dispositivos que se encuentran alrededor de él. También, un batphone necesita de un protocolo, (BATMAN) que permita una ordenada y adecuada comunicación, entre todos los teléfonos que se han configurado como batphone. Antes de comenzar la configuración es necesario comprender los requisitos que necesitan los dispositivos para poder lograr configurarse como batphone.

Es muy conocido que el hardware en el sistema operativo Linux se gestiona a través de archivos de texto, es decir, que cada dispositivo tiene su respectivo archivo de texto. Para modificar parámetros u obtener otras configuraciones deseadas basta solo con añadir los cambios en el archivo. En Linux este privilegio tiene un precio. No cualquier usuario puede realizar estas modificaciones, pues se necesita ingresar al sistema operativo como root ó superusuario. Ya que Android está basado en el núcleo de Linux trabaja de igual manera. Por lo tanto se debe de activar los permisos de superusuario en cada uno de los celulares sobre los que se va a trabajar.

Luego se instalan las aplicaciones: serval-batphone-helper y el serval-batphone-sipdroid-patched. Dichas aplicaciones son las encargadas de cumplir con las funciones que un batphone requiere.

3.3 Configuración preliminar en dispositivos móviles.

En los siguientes temas se mostrarán los pasos que se deben seguir en cada uno de los tres dispositivos, antes de poder instalar en ellos las aplicaciones modificadas por la comunidad Serval.

3.3.1 Configuración preliminar en el HTC Dream.

El HTC Dream fue el primer dispositivo con el que Serval inicio su proyecto. Su configuración es un poco más complicada que la de los otros móviles. En este teléfono se cambió la ROM para poder realizar tareas más complejas. La ROM original no disponía de muchos comandos y funciones que son indispensables para poder activar los permisos de super usuario y ejecutar las aplicaciones para el batphone. También, se modificó el espacio de la memoria flash, aumentando su capacidad para que soportara la nueva ROM.

Es importante comprender el concepto de ROM. En Android es un archivo que contiene todo el sistema operativo listo para instalar. Dicho de otra manera es el archivo a ser transferido a la memoria flash del teléfono.

Una ROM de Android es una copia de todos los ficheros que son necesarios para ejecutar el sistema operativo y las aplicaciones que vengan integradas en éste. Es donde reside tanto el núcleo de Linux, como todos los iconos e imágenes de las aplicaciones.

A continuación se detallan los pasos para realizar dicha configuración. Es importante mencionar que antes de realizar los siguientes pasos, se debe de realizar un respaldo de toda la información.

- Formatear la micro SD a FAT32 (en formato F16 probablemente no podría trabajar), luego copiar los siguientes archivos a la raíz de la micro SD, estos archivos deben tener el mismo nombre tal como se muestran en este documento.

++El firmware DREAMG.nhb: actualiza la memoria ROM, sobrescribiendo sobre la del fabricante HTC, se puede descargar del sitio:

http://wiki.cyanogenmod.com/index.php/Full_Update_Guide_-_G1/Dream_Firmware_to_CyanogenMod

++ G1 SPL.zip: este archivo es necesario para disponer de más memoria en la memoria flash, por la nueva ROM que se instalará. Su sitio descarga es:

<http://www.filecrop.com/latest-g1-spl.zip.html>

++Las aplicaciones de Google que el celular trae por defecto están en el archivo gapps-ds-ERE36B-signed.zip, las cuales se descargan del sitio:

<http://code.google.com/p/52and/downloads/detail?name=gapps-ds-ERE36B-signed.zip&can=2&q=>

++Luego se debe descargar la radio, que es la encargada de la funcionalidad de teléfono del dispositivo. Esto incluye GPRS, GSM, GPS y otras operaciones. También incluye la funcionalidad de la cámara. Se puede cambiar de ROM y no se pierden los datos. La radio es propia de cada dispositivo, lo que la hace un componente fundamental. Se debe tener cuidado en el momento de instalarla, ejecutar un mal procedimiento podría dejar el dispositivo “muerto”, sin ningún fundamento alguno. El link para la descargar el archivo ota-radio-2_22_19_26l.zip es:

http://code.google.com/p/sapphire-port-dream/downloads/detail?name=ota-radio-2_22_19_26l.zip&can=2&q=

++ Para entrar al modo recovery tal como se mencionó en el capítulo uno, se debe descargar la imagen recovery.img. Ésta es una aplicación que deja al teléfono en un estado anterior a la carga de la ROM, por lo que se puede cambiar, hacer un backup de ella o modificar las propiedades de la tarjeta SD. La imagen se obtiene de:

<http://www.filecrop.com/recovery.img.html>

++ La ROM a descargar es update-cm-5.0.7-DS-signed.zip, de CyanogenMod:

<http://code.google.com/p/cyanogenmod/downloads/detail?name=update-cm-5.0.7-DS-signed.zip&can=4&q=>

++Descargar la aplicación flashrec-1.1.3-20091107-2.apk que permite, montar una nueva imagen recovery en el dispositivo

<http://zenthought.org/content/project/flashrec>

- Todos los archivos deben colocarse en la tarjeta micro SD, excepto la aplicación flashrec que debe de colocarse en la carpeta platform-tools de la carpeta SDK. También se debe de retirar la tarjeta SIM del dispositivo. A continuación se dan los pasos para obtener la configuración root:

++ Apagar el teléfono e ingresar con el modo bootloader. Asegurarse que la batería del teléfono tiene suficiente carga.

++ Se deben seguir los pasos que aparezcan en pantalla para lograr la actualización de la ROM

++ Finalizado el paso anterior se debe reiniciar el teléfono presionando simultáneamente los botones de inicio de llamada y finalización de llamada.

++ Al reiniciar el teléfono, ya se estará corriendo el firmware que trae el archivo DREAIMG.nbh, siendo este el primer paso para obtener permisos de superusuario.

++ En este paso no se podrá tener acceso al celular , por lo tanto se deberá de digitar en el teclado qwerty del dispositivo el siguiente comando,

```
<Presionar enter> setprop persist.service.adb.enable 1 <Presionar enter> telnet <Presionar enter>
```

++ El cual ingresa a la shell en un segundo plano, habilita la depuración USB y lo prepara para correr en la shell en modo root. Probablemente parezca que no sucede nada, pero el buen funcionamiento depende de que se ingrese de manera correcta el comando.

++ Desde la consola del computador ingresar a la carpeta platform-tools del SDK e ingresar a la consola del dispositivo, con el comando adb shell

++ En la consola ingresar el comando ps, para verificar que existe una comunicación telnet con el dispositivo móvil.

++ Instalar la aplicación flashrec al teléfono desde líneas de comandos

++ En la computadora digitar lo siguiente para realizar una conexión telnet con el dispositivo indicando a través de que puertos se realizara la comunicación. Al ingresar estos comandos se está activando los permisos de superusuario al HTC DREAM.

```
adb forward tcp:9998 tcp:23; telnet localhost 9988
```

++ Ahora con los permisos de superusuario activados se debe ingresar:

```
setprop app-setupwizard.disable 1  
am start -a android.intent.action.MAIN
```

- Ejecutado el paso anterior se podrá ingresar al entorno Android, se debe buscar y ejecutar la aplicación Recovery Flasher.
- Se realiza el backup del flash recovery existente al escoger la opción Backup de la aplicación Flash Recovery.
- Luego, se carga una nueva imagen recovery, al ingresar la ruta /sdcard/recovery.img en la ventana y luego escoger "Flash Recovery Image". Terminado el proceso se debe de apagar el teléfono.
- Ahora se debe de ingresar al modo recovery del celular, cuando aparezca el menú se debe escoger wipe > wipe data/factory > reset. Presionar el botón HOME para confirmar.
- Finalizado presionar el botón Back
- Escoger la opción "flash ZIP from sdcard" e ingresar los archivos zips en el siguiente orden, este paso es muy importante ya que si no se sigue el orden existen riesgos de que el celular pierda su funcionalidad y se estropee.
 - ++ Seleccionar ota-radio-2_22_19_26l.zip, finalizado el proceso se debe reiniciar el dispositivo, el cual automáticamente mostrará el menú del recovery
 - ++ Luego ingresar el archivo Latest G1 SPL (no es necesario reiniciar después).
- Regresando al menú del recovery escoger "Apply sdcard" y seleccionar update-cm-5.0.7-DS-signed.zip para instalar la ROM y luego sin reiniciar escoger una vez mas "Apply sdcard" y gapps-ds-ERE36B-signed.zip.
- Continuando en el menú del recovery seleccionar "Factory reset" y luego reiniciar.
- El teléfono probablemente regrese al modo recovery y muestre "formateando caché", por lo tanto se debe escoger una vez mas "reboot".

Ejecutado el último paso el dispositivo iniciaría con un nuevo entorno. La ROM CyanogenMod 5.0.7 fue instalada en su memoria flash, la cual incluye la versión de Android 2.1, con los permisos de superusuario activados. Entonces solo se necesita descargar las aplicaciones, instalarlas y realizar la configuración de las mismas.

3.3.2 Configuración preliminar en el htc comet (Huawei Ideos U8150)

Para la configuración de este celular solamente basta con activar los permisos de superusuario. Las aplicaciones para que el dispositivo trabaje como batphone se publicaron dirigidas para este modelo, por lo tanto no hay necesidad de cambiar ROM. Estas aplicaciones se pueden ver como la segunda versión del batphone.

- Descargar e instalar la aplicación z4root.apk
- Ejecutar la aplicación y seleccionar Root permanente.

Con estos pasos se obtienen los permisos de superusuario, pudiendo instalarse las aplicaciones de configuración del batphone.

3.3.3 Configuración preliminar en el Huawei M835

La configuración de este celular presenta sus propios pasos para obtener la activación de los permisos de superusuario, a continuación se detalla cada uno de ellos:

- Descargar y descomprimir la carpeta SuperOneClickv1.9.5 del siguiente enlace:
<http://forum.xda-developers.com/attachment.php?attachmentid=624175&d=1307984755>
- Correr el ejecutable SuperOneClick que se encuentra en la carpeta que se descomprimió. Una vez presente la pantalla de usuario se debe seleccionar la opción Driver Check y al realizar esto deberá aparecer “Problemas no encontrado”
- Luego se debe seleccionar la opción Root y hay que esperar que el proceso se complete, siguiendo todos los pasos que se muestren.
- Finalmente reiniciar el teléfono. Si todo salió bien deberá aparecer una aplicación con el título Superuser, esto significa que se logró activar los permisos de superusuario.

Debe observarse que la configuración preliminar de cada teléfono es propia del modelo. Aunque en algunos el cambio de la ROM y la activación de los permisos de superusuario consiste en un proceso largo y complejo; en otros no es necesario el cambio de ROM y la instalación de una aplicación permite obtener los permisos root.

3.4 Serval-batphone-helper.

Esta aplicación es la encargada de arrancar el celular en modo batphone. En la interfaz de usuario cuenta con un icono de una red inalámbrica wifi. Al pulsarlo le ordena al teléfono que trabaje en modo ad-hoc e inicie el rastreo de otros batphones que se encuentren dentro del rango de cobertura del celular (aunque también detectará cualquier dispositivo que esté hablando en el protocolo batman). Para poder trabajar de manera adecuada debe tener instalado el protocolo batman especial para dispositivos móviles. Esta aplicación fue lanzada por primera vez para que trabajara con el móvil HTC DREAM G1, bajo la plataforma Android 2.1.

3.4.1 Código base: Wifi Tether

El Android Wifi Tether es un programa que permite utilizar el terminal como un modem y así brindar el servicio de internet a otros móviles o computadoras, se puede navegar utilizando la red 3G o cualquier plan de datos que se tenga contratado con el dispositivo que realiza el papel de modem [21].

El motivo por el que esta aplicación llamó la atención de la gente del Serval Project es debido a que su principio básico es muy simple ya que la aplicación hace que el móvil actúe como punto de acceso Wifi (en modo ad-hoc) o incluso por Bluetooth. Al conectarse en modo ad hoc permite que los dispositivos inalámbricos (móviles en este caso) se conecten entre sí para formar una red punto a punto, es decir, una red en la que cada equipo actúa como cliente y

como punto de acceso simultáneamente. Una vez conectados entre sí, se podrá navegar libremente. Pero esta aplicación solicita permisos de super usuario al instalarse, ya que debe realizar una configuración en la interfaz inalámbrica, la cual consiste en activar el modo ad-hoc.

Un punto importante a mencionar es que además de este programa existen otras aplicaciones en Android orientadas a la misma finalidad. Sin embargo la gente de Serval Project se decidió por modificar el código de esta aplicación y adaptarlo a sus necesidades.

3.4.2 Estructura del .apk del serval-batphone-helper

Esta aplicación tiene una característica especial, ya que cuenta con librerías nativas, que son las encargadas de trabajar con los drivers del dispositivo móvil. Uno de los drivers con los que se trabaja es el de la tarjeta inalámbrica del dispositivo móvil. Una breve descripción de la estructura de la aplicación serval-batphone-helper se muestra en la Figura 13:

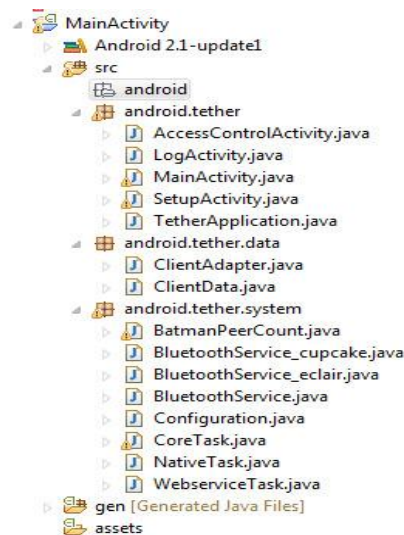


Figura 13. Contenido de la carpeta src de la aplicación Serval-Batphone-Helper

A continuación se describe la estructura de ficheros de la Figura 13:

- Dentro de la carpeta src se encuentran 3 paquetes, el android.tether, el android.tether.data y el android.tether.system , cada uno contiene sus respectivas clases, estas se pueden observar en la Figura 13, y a continuación se describen algunas de las clases:
 - ++ AccessControlActivity : Clase, relacionada con la opción Access Control del menú de la aplicación.
 - ++LogActivity : Clase, es la encargada de mostrar lo relacionado a la opción ShowLog.
 - ++MainActivity: Clase principal, es la primer interfaz que se observa al iniciar la aplicación.
 - ++SetupActivity: Clase, muestra la pantalla que aparece al escoger la opción setup.
 - ++TetherApplication: Clase, muestra la pantalla resultante y ejecuta los procedimientos

necesarios para levantar el batphone.

- Carpeta gen, donde se encuentra el archivo generado R.java generado después de la compilación de la aplicación.
- Cuenta también con una carpeta jni, donde se encuentra iwconfig.patch que es un código desarrollo en c que permite la comunicación desde la aplicación java a la interfaz inalámbrica.
- En la carpeta libs se contiene las librerías binarias de Linux, tal como se muestra en la Figura 14 :

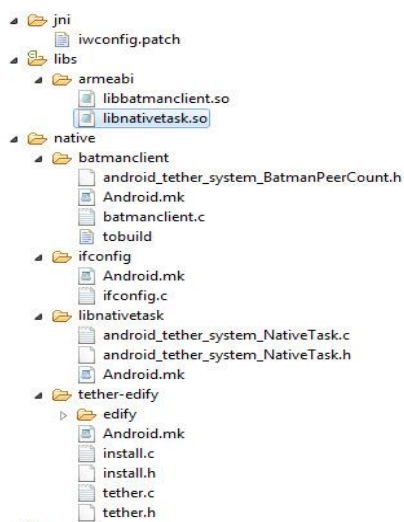


Figura 14 Contenido de las carpetas jni, libs y native de la aplicación Serval-Batphone

- La librería nativa, como su nombre lo indica contiene archivos que fueron escritos en otro lenguajes de programación, en la Figura 14, puede observarse que existe archivos que son propios de batman y que no se puede crear en Java.
- Las imágenes se encuentran en la carpeta res, asi como también archivos de configuración.
- Para finalizar también incluye un archivo AndroidManifest.xml.El cual se muestra a continuación.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="android.tether" android:versionCode="23" android:versionName="2.0.5pre2">
    <uses-sdk android:minSdkVersion="7" android:targetSdkVersion="7"/>
    <application android:icon="@drawable/icon" android:label="@string/app_name"
        android:name="TetherApplication">
        <activity android:name=".MainActivity"
            android:label="@string/app_name"
            android:configChanges="orientation|keyboardHidden"
            android:screenOrientation="portrait"
            android:launchMode="singleTask">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".LogActivity" android:screenOrientation="portrait"
            android:launchMode="singleTask"/>
        <activity android:name=".AccessControlActivity" android:screenOrientation="portrait"
            android:launchMode="singleTask"/>
        <activity android:name=".SetupActivity" android:launchMode="singleTask"/>
    </application>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
    <uses-permission android:name="android.permission.WAKE_LOCK"></uses-permission>
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <uses-permission android:name="android.permission.BLUETOOTH"></uses-permission>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"></uses-permission>
    <uses-permission android:name="android.permission.VIBRATE"></uses-permission>
</manifest>

```

Al observar la composición del archivo Manifest, se puede notar que al final del archivo se están solicitando los permisos para trabajar con el hardware del dispositivo móvil. También se puede observar la actividad principal que será la que se ejecute cuando se inicie la aplicación. Esta tiene el nombre MainActivity.

En la Tabla 7 se muestran las funciones a utilizar para que se obtengan algunos permisos de acceso en las aplicaciones modificadas del Wifi Tether y Sipedroid, que son el código base modificado por Serval. Estas funciones debe declararse en el Android Manifest.

PERMISOS	DESCRIPCION
RECEIVE_BOOT_COMPLETED	El software debe iniciarse automáticamente cuando el teléfono se encienda, si la aplicación estaba en funcionamiento cuando se apagó.
ACCESS_WIFI_STATE, CHANGE_WIFI_STATE	Se puede usar los controladores de wifi para establecer comunicación.
INTERNET	Este permiso es requerido para crear cualquier tráfico de red.
READ_PHONE_STATE	La primera vez que se configura el software, se trata de obtener el número real del dispositivo para que quede registrado en la red.
BLUETOOTH, BLUETOOTH_ADMIN	Permite habilitar la opción de usar audífonos durante una llamada.
PROCESS_OUTGOING_CALLS	Se interceptan las llamadas salientes, mientras el software se activa para dar la opción de llegar al contacto a través de la malla.
CALL_PHONE	Si se decide no utilizar el software, es necesario para realizar la

	llamada de nuevo.
RECORD_AUDIO	Utilizado para la transmisión de voz
DISABLE_KEYGUARD	Permite contestar llamadas entrantes sin tener que pasar por el desbloqueo de pantalla.
MODIFY_AUDIO_SETTINGS	Cambiar el volumen de la llamada.
WRITE_SETTINGS	Relacionado con la configuración de ringtones, volumen y vibración para la notificación de llamadas entrantes.
VIBRATE	Notificación de llamadas entrantes.
READ_CONTACTS	Mostrar el nombre de la persona que llama
WRITE_CONTACTS	Inserta información de llamadas entrantes y salientes
WAKE_LOCK	Asegura mantener el teléfono despierto para los paquetes de reinstalación.
CAMERA	Sipdroid presenta un soporte para video llamadas, sin embargo es probable que este aún no trabaje. Así que este permiso puede ser removido.
WRITE_EXTERNAL_STORAGE	Escritura de archivos de registro y almacenamiento para acelerar el proceso de desarrollo.
BROADCAST_STICKY	Se emite el estado de ad hoc del wifi a la interfaz de usuario y otras aplicaciones.
READ_SMS, WRITE_SMS	Si se instala el SMDDROID en el teléfono, el software se puede utilizar para enviar mensajes

Tabla 7. Permisos de acceso a hardware de dispositivos

3.5 Instalación de serval-batphone-helper en dispositivos.

En esta sección se detallan los pasos a seguir para la instalación de la aplicación serval batphone helper en cada uno de los tres dispositivos.

3.5.1 Instalación de serval-batphone-helper a htc dream g1

Asumiendo que se cuentan con permisos de superusuario, y que se han descargado la aplicación serval-batphone-helper, se comienza con la instalación. Los pasos de instalación comienzan por la instalación del protocolo batman que como se menciona anteriormente es necesario para llevar a cabo el enrutamiento de los dispositivos:

- Descargar el ejecutable desde el sitio:
http://afrimesh.googlecode.com/files/batmand-rv1543_armv6l
- Una vez descargado, el binario debe colocarse en una carpeta que se creará en el dispositivo con el nombre net.open-mesh.batman. Luego conectar el celular a la computadora e ingresar los comandos desde la consola, tal como se muestra a continuación:

```
adb shell mkdir /data/data/net.open-mesh.batman
adb push batmand-rv1543_armv6l /data/data/net.open-mesh.batman/batmand
```

- Se debe de levantar un socket para batman, que no es más que un puerto específico al que se le asignara la tarea de comunicar y recibir paquetes BATMAN. Los pasos a seguir para la instalación son:

```
adb shell
cd /data/data/net.open-mesh.batman
./batmand tiwlan0
```

Es muy probable que al momento de realizar el último comando muestre un aviso de error. Por lo tanto, es necesario realizar unos cambios en los archivos que trabajan con la interfaz inalámbrica del teléfono, luego se inicia la aplicación Serval Batphone. Los archivos y sus respectivos cambios que deben realizarse son:

- El archivo tiwlan.ini contiene la configuración de la red inalámbrica. En este documento lo importante es notar que se tenga la configuración en modo ad hoc. Ingresar los siguiente comandos y realizar los cambios:

```
adb pull /data/data/android.tether/conf/tiwlan0.ini
```

Al ejecutar el comando anterior, el archivo tiwlan0.ini se debe encontrar en la carpeta platform-tools del SDK. Al abrirse el archivo debe de cerciorarse que se tiene activada la opción ad-hoc. Para confirmarlo se debe de buscar la siguiente línea WifiAdHoc = 1. Si tiene el número 0 hay que cambiarlo a 1.

Luego guardar los cambios y ubicar una vez más el archivo en la ruta donde se encontraba:

```
adb push tiwlan0.ini /data/data/android.tether/conf/tiwlan0.ini
```

- El archivo wpa_supplicant.conf contiene un registro de todas las redes inalámbricas detectadas por el celular. También, contiene el nombre de la interfaz inalámbrica del dispositivo. En ese caso se debe extraer el archivo y configurarlo tal como se hizo con el archivo tiwlan0.ini. En este debe asegurarse que la interfaz tiene el nombre de tiwlan0.

```
adb pull /data/misc/wifi/wpa_supplicant.conf
```

Una vez se haya verificado o configurado el archivo, se debe guardar y colocar el archivo en la misma carpeta de donde se extrajo

```
adb push wpa_supplicant.conf /data/misc/wifi/wpa_supplicant.conf
```

Cabe mencionar que la configuración de los archivos pudo realizarse desde el terminal de la computadora o desde el dispositivo móvil utilizando el comando cat, que permite trabajar con archivos de texto.

El HTC DREAM G1, tiene un paso extra que se debe realizar, ya que se debe de incorporar en el directorio /data/data del teléfono la siguiente carpeta.

```
adb push serval-data.data-20100724.tgz /sdcard
adb shell
cd /data/data
tar zxvf /sdcard/serval-data.data-20100724.tgz
exit
```

3.5.2 Instalación de serval-batphone-helper a htc comet y metropcs

Fechas después de haber publicado la aplicación serval-batphone para Android, lanzada especialmente para el HTC DREAM, aparece en el sitio de serval la aplicación orientada a otro modelo: el Huawei U8150. Este presentaba una mejora, la cual era la solución de no descargar el binario de batman e instalarlo al dispositivo, pues ya venía incluido en la aplicación para la plataforma Android 2.2. De la misma forma esta aplicación es compatible para el Huawei M835 que pertenece a la misma familia que el Ideos U8150.

Al descargarse el archivo del sitio de Serval Project, se debía de cambiar el nombre a serval-batphone-helper, luego se realizaba la instalación en el dispositivo, asumiendo también que ya se tiene la configuración como super usuario.

3.5.3 Configuración de la aplicación serval batphone.

Una vez instalada la aplicación es necesario realizar la configuración de manera individual a cada dispositivo que contenga la aplicación batphone. La configuración es la misma para todos los teléfonos así que se debe de seguir los siguientes pasos.

Iniciar la aplicación y presionar el botón menú. Luego, escoger Setup. Se debe escoger la opción "Cambiar SSID", que se refiere al nombre que se le asigna a toda la red mesh. Ésta debe tener el mismo nombre para todos los móviles. Aquí se decidió asignar el nombre Potato.

Luego, siempre en Setup, buscar la opción Cambiar canal (que se refiere a la frecuencia a la cual se trabajará, 2.4 - 2.5 GHz). En todos los teléfonos se debe escoger una frecuencia que sea compatible a wifi , se asignara el canal 11 que se encuentra a una frecuencia de 2462 MHz.

Siempre en el menú Setup escoger la opción Cambiar Lan, que es la dirección IP que se le asignara de manera manual al dispositivo móvil, se debe de aclarar que esta parte es importante porque el número con el que se le podrá llamar a cada dispositivo será el último octeto de la dirección IP.

3.5.4 Entorno de la aplicación batphone

A continuación se abordará todas las opciones que presenta la aplicación, desde la configuración principal a características sencillas que se encuentran en el serval batphone.

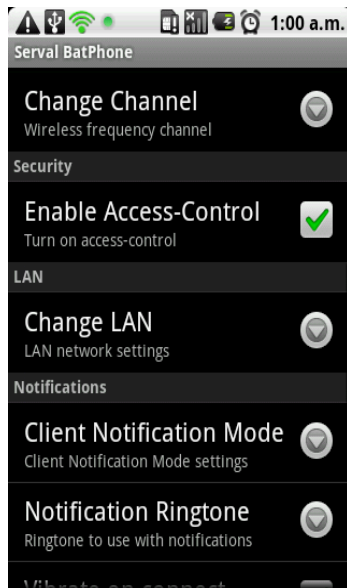
La pantalla principal está compuesta en la parte inferior de un indicador de subida de paquetes, un indicador de bajada de paquetes y un de detector de dispositivos comunicándose con el protocolo BATMAN. Este último es importante al momento de comprobar el funcionamiento del dispositivo como batphone.

También aparece el icono de conexión wifi en color verde esperando que se ejecute la aplicación batphone, cuando este se encuentra activo cambia el color a gris.

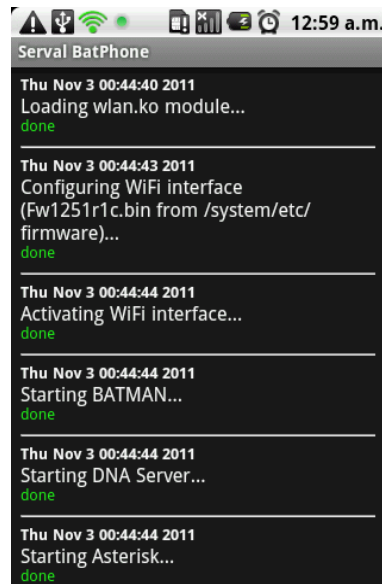
Al estar en la pantalla principal y presionar la tecla menú del dispositivo se presentan cuatro opciones las cuales son:

- **About:** Presenta información relacionada a la versión de la aplicación, el sitio oficial del Serval y del proyecto, sitio oficial de la aplicación wifi tether, información de licencias, nombre de los autores de la aplicación, nombre de contribuidores. También, se incorporan opciones que permite hacer donaciones a las organizaciones Serval y los creadores de Wifi Tether.
- **Show Log:** Esta opción es muy importante ya que permite verificar de manera inmediata si algún módulo o proceso falla en el momento de ejecutar el batphone. Es una manera de depuración de la aplicación desde el móvil. Al entrar a esta opción se muestra los pasos que se realizan en el dispositivo para lograr la configuración.
- **Setup:** Incluye varias configuraciones del dispositivo. Entre estas se encuentra la de cambiar de la interfaz inalámbrica wifi a bluetooth. Sin embargo esta opción es propietaria de la aplicación wifi tether y para la aplicación batphone no funciona, ya que batman se ejecuta solamente en wifi.

Dentro de la misma categoría se encuentra la configuración del wifi, que incluye habilitar la encriptación, sin embargo no es indispensable realizar este cambio en la aplicación. Las variables que interesan en esta categoría son el cambio de SSID que debe ser el mismo para todos los batphone, y el cambio de canal, que de igual manera deberá de ser igual para todos. Ver Figura 15(a).



(a)



(b)

Figura 15. Entorno de Serval Batphone (a) Setup, (b) Show Log.

- **Access Control:** En esta apartado de la aplicación se configura, si se autoriza la conexión de clientes. Ver Figura 15 (b).

La categoría que tiene un mayor interés es la de cambio de LAN. Ésta permite asignar el número que identificará a cada dispositivo móvil, el cual debe ser único por lo tanto ningún otro teléfono puede tener la misma IP.

De la dirección IP que se ingresa, debe tener mayor prioridad el último octeto, es decir el último término (después del tercer punto). Para realizar llamadas a otros batphones se debe marcar el último octeto perteneciente a la dirección IP de cada móvil. A continuación en la Tabla 8 se muestra un ejemplo de la configuración realizada a cada móvil

CONFIGURACION	HTC DREAM	HUAWEI U8150	HUAWEI M835
DIRECCION IP	10.130.1.180	10.130.1.181	10.130.1.182
SSID	potato	potato	potato
CANAL	1	1	1

Tabla 8. Parámetros de configuración en Bapthone Serval Helper

Por lo tanto, si se desea marcar al Huawei Ideos U8150, se debe ingresar el número 181 y luego iniciar la llamada.

3.6 Serval-Patched-Sipdroid.

El equipo de Serval Project modificó el código de la aplicación Sipdroid, para poder complementar el batphone. Como se observó en capítulos anteriores, al instalarle al celular la versión de Serval de wifi tether este queda dotado con la opción de trabajar en modo adhoc. Operando bajo este modo debe detectar los celulares y otros dispositivos que se encuentran a su alrededor utilizando el protocolo BATMAN. Ahora bien, para realizar una llamada se le puede instalar la aplicación Sipdroid, mas sin embargo por el hecho de tener que conectarse al PBX virtual este proyecto aun dependería de infraestructuras telefónicas, que van de la mano con proveedores del servicio de telecomunicaciones.

Lo ideal era evitar consultar una central externa para poder realizar las llamadas, eso condujo a la idea de levantar una centralita en el dispositivo móvil. Mediante la modificación del código y la incorporación del binario del protocolo de VoIP Asterisk.

Asterisk es un programa de software libre que proporciona funcionalidades de una central telefónica. Como cualquier PBX se puede conectar a un número determinado de teléfonos para hacer llamadas entre sí.

Entonces ahora la modificación consiste en lograr que la aplicación Sipdroid hiciera referencia a la centralita Asterisk ubicada en el móvil. La cual contendrá una tabla de registro de cada batphone que se encuentre en toda la red y este permitirá enrutar la llamada de acuerdo al número marcado.

3.6.1 Sipdroid

La aplicación Sipdroid es un cliente SIP/VoIP para Android que permite realizar llamadas desde un proveedor VoIP al que se deberá estar suscrito [22].

SIP significa Session Initiation Protocol (protocolo de inicio de sesión). Es un protocolo de señalización de telefonía IP utilizado para establecer, modificar y terminar llamadas VOIP. Este protocolo describe la comunicación necesaria para establecer una llamada telefónica. En un principio se debe de tener una cuenta en un servidor SIP sin importar cual sea, luego se debe de ingresar y crear una cuenta en el sitio:

https://www4.pbxes.com/index_e.php

El cual es una PBX virtual alojada en un servidor, que permite entre muchas cosas , acceder a funciones avanzadas desde un cliente SIP. La aplicación Sipdroid ya viene configurada para trabajar con esta centralita, por lo tanto es obligación crear el usuario en este sitio.

Luego que se han ingresado los datos, ya se puede configurar la aplicación Sipdroid para que trabaje realizando llamadas gratis a través de internet. Cabe mencionar que esta aplicación permite realizar su conexión con un plan de datos Edge, 3G o también utilizando la red inalámbricas Wifi.

3.6.2 Estructura del .apk del serval-patched-sipdroid

La aplicación serval-patched-sipdroid está compuesta de la siguiente forma:

- La carpeta src contiene 22 paquetes tal como se muestra en la Figura 16. La actividad principal se encuentra en el paquete org.sipdroid.sipua.ui, que muestra una ventana de configuración antes de iniciar la aplicación serval patched
- La carpeta gen donde se encuentra el archivo generado R.java generado después de la compilación de la aplicación.

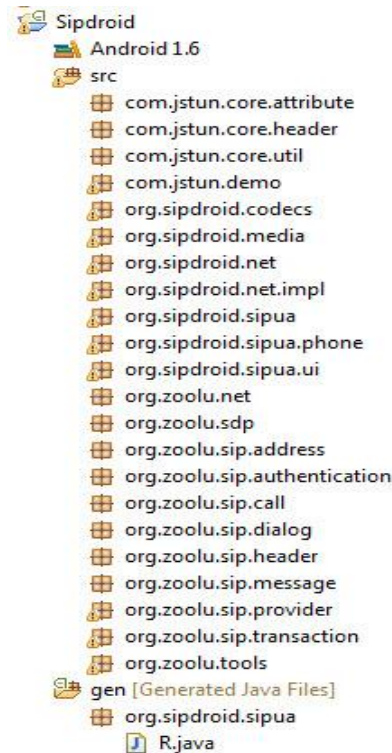


Figura 16. Contenido de carpeta src de aplicación serval patched sipdroid

- La carpeta jni en la cual se encuentra archivos en código nativos, todos escritos en código C y C++.
- En la carpeta libs se encuentran las librerías pertenecientes al kernel Linux.
- La carpeta res contiene las imágenes que se utilizan para la aplicación, separadas especialmente en dos carpetas. En la carpeta drawable se encuentran las imágenes utilizadas para toda la aplicación. En la carpeta drawable-finger están las imágenes utilizadas para la pantalla de marcado, en su mayoría son todas las imágenes utilizadas para formar el teclado al momento de ingresar el número para realizar la llamada. La carpeta res también contiene subcarpetas con definición de cadenas de caracteres, asignadas a cada interfaz de usuario que contiene la aplicación. También en su interior está

Ubicadas subcarpetas que contienen los archivos xml en los que se definen las distintas interfaces de usuario. En la carpeta layout están las interfaces que se presentan al recibir o realizar la llamada, así como también la pantalla principal de la aplicación. La carpeta layout-finger contiene la estructura de la interfaz que se muestra del teclado numérico al realizar la llamada.

- El archivo AndroidManifest.

3.7 Instalación de serval-patched-sipdroid.

La instalación para los tres modelos de celulares es la misma. Por lo tanto solo basta con descargar la aplicación. Luego, desde una consola dirigirse a la carpeta platform-tool del directorio SDK. Una vez estando ahí ejecutar el siguiente comando:

```
adb install serval-patched-sipdroid.apk
```

3.7.1 Configuración de la aplicación serval-patched-sipdroid.

Luego que se ha realizado la descarga y se ha instalado la aplicación se debe ejecutar. Una vez se encuentre en la pantalla principal presionar la tecla menú del dispositivo móvil y seleccionar la opción Settings.

En Settings se debe escoger "Ajuste a cuenta SIP". Luego, se debe escoger "Authorization Username". Mientras que en la aplicación Sipdroid se ingresa el nombre del usuario de la cuenta de un sitio SIP virtual, en Sipdroid Patched se ingresa el valor 4000. Ésta configuración ya viene incluida en la aplicación. Si se ingresa otro valor dicha aplicación no logrará alcanzar la comunicación.

También, debe modificarse el campo "Server", en ésta se coloca la dirección de la PBX virtual en la aplicación Sipdroid. Pero en la aplicación Sipdroid Patched se debe colocar la IP local del teléfono móvil la cual estará amarrada con el software asterisk que se encargará de la gestión de los números asignados a cada uno de los batphone. La dirección IP local es 127.0.0.1

Finalizado este paso, ya se tiene configurado el móvil como un batphone.

3.8 SERVAL-0-05.

El 15 de agosto del 2011, la comunidad serval publicó el prototipo de la nueva versión del batphone que incluía nuevos cambios. En esta oportunidad la aplicación consiste en un solo archivo .apk que puede ser descargado del sitio de Serval o del Market de Android.

Una diferencia muy importante de mencionar es que en esta versión ya viene incluida la Arquitectura de Distribución Numérica, tema que se explica en la sección 4.7. Por esta causa no se puede comunicar la versión Serval-0-05 con las otras versiones del batphone, ya que en estas últimas no se ejecuta el DNA.

Un cambio particular son los que se realizaron en la interfaz de usuarios e imágenes de la aplicación. Este prototipo contiene las siguientes funcionalidades:

- La habilidad de participar en una red mesh en modo ad hoc (siempre y cuando el teléfono tenga permisos de super usuario)
- La habilidad de conectarse a un punto de acceso
- La habilidad de ajustar de manera manual el modo de conexión del Wi-Fi.
- Iniciar y recibir llamadas en una red mesh (batphone)
- Iniciar y recibir llamadas sobre un punto de acceso.

3.8.1 Configuración del Serval Batphone 0-05

Al inicio la aplicación solicita que se ingrese el número telefónico con el que se identificará quiere se le identifique dentro de la red, permitiendo un mínimo de 5 dígitos y un máximo de 32.

Al encontrarse en la pantalla principal, se presentará un menú con tres opciones:

- Opción de inicio: Gestiona la activación (start) o desactivación (suspend) de la aplicación Serval.
- Opción de asignación de número (Reset): Con esta opción se ingresa el número telefónico que identificará al teléfono dentro de la red Mesh. Los símbolos permitidos son los símbolos +, * , # y los dígitos entre 0 y 9.
El número que se asigne no puede comenzar con los dígitos "11", ya que es el rango reservado para el número de emergencia
- Realizar llamadas (Make a Call): Esta opción permite seleccionar si la llamada será para la red Mesh o una llamada utilizando la tarjeta SIM del proveedor de servicio telefónico

De la misma manera como las versiones anteriores, el batphone contiene opciones para realizar la configuración del teléfono que se desee. A continuación se verán de manera detallada cada uno de ella.

3.8.2 Menú Ajuste

Dentro de este menú se encuentran las siguientes opciones:

- Modo Wifi
Este presenta una lista desplegable con cuatro opciones:
 - ++ Adhoc: que permitirá que el dispositivo participe en una red Mesh. Esta opción estará disponible si se han activado los permisos de superusuario en el teléfono.
 - ++ Cliente: Permite la conexión a un punto de acceso.
 - ++ Punto de Acceso: Permite a otros teléfonos en modo cliente conectarse al dispositivo móvil.
 - ++ Apagado: Deshabilita el wifi en el teléfono.
- Modo Cíclico Automático
Cuando esta opción se encuentra activada, se estarán realizando cambios entre los modos Wifi cada 30 segundos, hasta que otro batphone sea detectado. Cuando la conexión se pierde, el ciclo se reanuda.
- Ajustes Wifi
Modifica las configuraciones de Wifi para solucionar problemas con el wifi o la conexión a una red mesh.

Como puede observarse la única configuración que se debe de realizar es la asignación de un número telefónico. Todos los pasos que se realizan con los otros modelos de batphone se reducen y ocasionalmente se tendrá que configurar el canal de frecuencia, dependiendo de la frecuencia a la que trabaje la red.

3.9 Conclusión

La necesidad de lograr el enlace entre los celulares sin contar con ninguna infraestructura, requiere el trabajo de lograr la configuración adecuada en los dispositivos. Esta configuración dependerá del dispositivo a utilizar. Como se observó en este capítulo, el proceso de configuración, activación de permisos de superusuario, cambio de ROM, es único para cada modelo de celular. Comprender cada uno de los requerimientos que necesita un celular para ser configurado como batphone, permitirá determinar de manera más sencilla si un dispositivo móvil puede aplicar o no a dicha aplicación.

Es importante también mencionar que la estructura de las aplicaciones bases permite comprender el funcionamiento del batphone y la finalidad que busca la comunidad Serval. La mayor información según se explicó se puede encontrar en los archivos Android Manifest y luego en todas las clases que son declaradas como actividades.

Capítulo 4: Batphone en red mesh.

En este capítulo se describe en qué consiste una red de telefonía inalámbrica Mesh Potato, como está constituida y cuál es su finalidad. Se presenta el aporte que brinda la aplicación Batphone dentro de ese contexto.

Finalizando con las configuraciones que deben de realizarse en cada uno de los dispositivos de la red para que se pueda establecer la comunicación telefónica.

4.1 Red mesh.

El batphone es un derivado del proyecto Village Telco, que es una comunidad que tiene como finalidad construir redes telefónicas con hardware y software de bajo costo. Su idea consiste en independizar la telefonía de los grandes operadores.

El elemento principal de la red es el router MP, que tiene un puerto FXS diseñado para conectar un teléfono analógico. Dentro de la red mesh existen diversos nodos comunicándose entre sí (Ver Figura 17). Cada nodo no es más que un router inalámbrico, que cuenta con un puerto ATA (adaptador de teléfono analógico) al que se le conecta a través un cable con conductores de cobre (RJ11) un teléfono analógico para poder establecer una llamada con cualquier otro teléfono que se encuentra conectado a otro router inalámbrico dentro de la red. Estos nodos deben de poder trabajar en modo ad hoc y tener la capacidad de soportar el enrutamiento BATMAN.

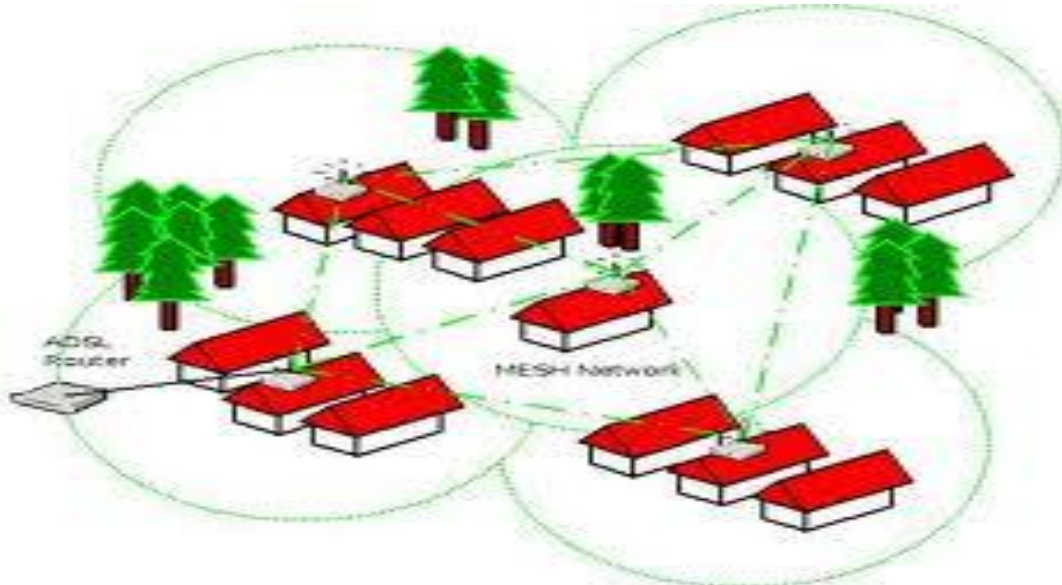


Figura 17. Red Mesh

Las características de un nodo son prácticamente idénticas con las que cumplen los dispositivos móviles configurados como batphone. Debido a esto se puede considerar a cada celular con la aplicación batphone instalada como un nodo móvil. De la misma manera la característica más

importante entre ambos nodos es que los dos tienen un lenguaje que les permite comunicarse entre sí. Dicho lenguaje es el protocolo BATMAN.

4.1.1 Configuración de la Red

Existen parámetros muy importantes que deben ser idénticos en cada uno de los dispositivos que conforman la red Mesh Potato. Estos parámetros deben ser similares para todo dispositivo que forma parte de la red. A continuación se detallan dichos parámetros.

- Asignación de números.

Tal como funcionan todas las redes telefónicas, cada dispositivo que forma parte de la red tiene nombre único, que le permita ser identificado de los demás dispositivos. Este identificador es el número telefónico, que dentro de la red Mesh Potato corresponde a una dirección IP. Por lo tanto se crean dos grupos de direcciones IP. El primer grupo corresponde a las direcciones de los dispositivos móviles. El segundo grupo a las direcciones de los routers MP01.

Para nuestras pruebas se decidió asignar a la red Mesh Potato el grupo de direcciones IP destinada a dispositivos fijos van desde la 10.130.1.2 hasta la 10.130.1.179. Luego los dispositivos móviles cuentan con el rango 10.130.1.180 hasta la 10.130.1.254

Como se mencionó en el capítulo 3, la dirección que se le asigna a un dispositivo móvil debe ingresarse en el campo cambiar LAN del menú Setup de la aplicación Serval Batphone.

- Selección de Frecuencia

Se debe de establecer una frecuencia que será con la que trabajara toda la red, esta debe de encontrarse en el rango de las frecuencias libre. La red a la cual se conectarán los teléfonos móviles operan en la frecuencia 2412 MHz.

La selección de la frecuencia 2412 MHz en dispositivo móvil se realiza en la opción Cambiar Canal, al buscar la frecuencia definida esta pertenece al canal 1.

- Asignación de la SSID

El SSID (Service Set Identifier) es un nombre incluido en todos los paquetes de una red inalámbrica (Wi-Fi) para identificarlos como parte de esa red. El código consiste en un máximo de 32 caracteres que la mayoría de las veces son alfanuméricos (aunque el estándar no lo especifica, así que puede consistir en cualquier carácter). Todos los dispositivos inalámbricos que intentan comunicarse entre sí deben compartir el mismo SSID.

El SSID que se define para la Red Mesh es "potato", el cual se ingresa al celular a través de la opción cambiar SSID, del menú ajuste.

Una vez más se recuerda que los parámetros mencionados anteriormente deben ser los mismos para todos los dispositivos que se encuentran en la red mesh, en caso contrario el dispositivo no podrá establecer ningún tipo de comunicación.

4.1.2 Configuración de un router desde un dispositivo móvil

Para realizar la configuración de parámetros en un móvil basta con acceder a la aplicación instalada en el celular e ingresar los valores en los campos adecuados. Por otra parte cuando se desea modificar algún parámetro en algún router, se puede realizar de diferentes maneras. Si es a través de una computadora de escritorio, ésta deberá de ingresar al router a través de un puerto Ethernet. Una vez establecida la comunicación se realiza un enlace vía telnet hacia el punto al que se desea conectar.

Otra forma de acceder a un router es a través de una computadora portátil. Para ello se debe estar muy cerca del nodo y la interfaz inalámbrica debe trabajar en modo adhoc y, además, soporte el protocolo BATMAN.

Otra alternativa es establecer comunicación mediante un enlace telnet desde el teléfono hacia el router. Una de las formas para ingresar desde un dispositivo móvil al router es a través del comando telnet (para esto es necesario descargar desde el Market e instalar la aplicación emulador de terminal). Ingresando el comando telnet y colocando la dirección IP del router al que se quiere ingresar.

```
$ telnet 10.130.1.140
```

Al ingresar al router, se necesitara conocer los comandos para realizar la modificación via consola. También se puede ingresar al entorno gráfico del router, esto se realiza abriendo un explorador en el dispositivo. Se debe ingresar la dirección IP del router. Esto cargará la interfaz de usuario del router para realizar los cambios adecuados.

4.2 Realizar llamadas.

El MP01 corre una distribución de Linux, sobre la que se montó el servidor VoIP Asterisk que es un software de código abierto que realiza el trabajo de una central telefónica (PBX) que administra la línea telefónica y las llamadas entre los Mesh Potato.

Una innovación fundamental desarrollada por el equipo de Village Telco fue configurar Asterisk para que pudiera operar haciendo un plan de marcación basado en direcciones IP. Por lo tanto para realizar una llamada de un teléfono a otro se marca la dirección IP. También, es posible establecer la comunicación marcando el último octeto de la dirección IP

Esta arquitectura permite que los router Mesh Potato puedan comunicarse sin ninguna autoridad central. A pesar de que la colisión de direcciones IP es un riesgo es una desventaja que se puede controlar. Este problema se controla con el serval DNA que se implementa en la última versión

pública del Serval Batphone. La cual consiste en asignar un número telefónico a cada dispositivo móvil, esto se realiza asignándole un SID (ID por abonado en español). Las llamadas se realizan marcando el último octeto de la dirección IP del batphone con el que se desea establecer comunicación. Mientras que todas las llamadas presentarán el número 4000 que es la extensión de la IP local asignada al Asterisk, tal como se muestra en la Figura 18.



Figura 18. (a) Llamada realizada al Huawei U8150 según tabla 8, (b) Llamada entrante El Serval Project se basa en la idea de Village Telco para agregar un número telefónico a la capa de asignación de dirección SIP, que convierte los números marcados en direcciones SIP. Esto establece el contacto con el batphone que tenga la IP marcada, usando el mismo método que se ha utilizado en la red Mesh Potato.

De esta manera es posible realizar una llamada para un dispositivo que se encuentre dentro de la red Mesh Potato a un teléfono configurado como batphone y viceversa.

4.3 Monitoreo de batphones.

Village Telco utiliza BATMAN como protocolo de enrutamiento para la red Mesh Potato, Serval imitó esa idea, de ahí el nombre de Bathphone. BATMAN ha demostrado tener un buen rendimiento dentro de la red Mesh Potato así como también en la comunicación entre los móviles Batphone.

Cualquiera que sea el protocolo de enrutamiento que se desea usar para una red malla con dispositivo móviles, debe tener una escalabilidad aceptable. Los protocolos de enrutamiento deben de ser capaz de lograr comunicación en una red grande si consumir demasiado ancho de banda. Para BATMAN esta modificación puede resultar relativamente fácil de lograr. Cada nodo anuncia el estado de sus vecinos y se puede modificar por un lado el TTL (Time To Life), de los

paquetes que se utilizan para detectar nodos vecinos. De manera que los anuncios cercanos tienen mayor prioridad al reducir el tiempo de vida, se estarían descartando los nodos lejanos y por lo tanto se tiene un mayor ancho de banda.

Otra ventaja que presenta el protocolo BATMAN es que dentro de sus herramientas viene incluido el monitoreo de todos los dispositivos que forman parte de la red. Que permite identificar la dirección IP de todos los usuarios dentro de la red. Este comando se puede realizar en cualquier elemento que soporte el protocolo BATMAN, al ingresar los comandos que se presentan a continuación

```
# ./batmand -cd1
./batmand -cd1
  Originator (#/255)      Nexthop [outgoingIF]:  Potential nexthops ... [B.A.T.M.A.N.0.3.2,
MainIF/IP: eth0/10.130.1.182, UT: 0d 0h 0m]
10.130.1.181 ( 39)  10.130.1.181 [ eth0]:  10.130.1.181 ( 39)  10.130.1.180 ( 12)
10.130.1.180 ( 39)  10.130.1.180 [ eth0]:  10.130.1.180 ( 39)  10.130.1.181 ( 9)
  Originator (#/255)      Nexthop [outgoingIF]:  Potential nexthops ... [B.A.T.M.A.N. 0.3.2,
MainIF/IP: eth0/10.130.1.182, UT: 0d 0h 0m]
10.130.1.181 ( 49)  10.130.1.181 [ eth0]:  10.130.1.181 ( 49)  10.130.1.180 ( 16)
10.130.1.180 ( 44)  10.130.1.180 [ eth0]:  10.130.1.180 ( 44)  10.130.1.181 ( 13)
  Originator (#/255)      Nexthop [outgoingIF]:  Potential nexthops ... [B.A.T.M.A.N. 0.3.2,
MainIF/IP: eth0/10.130.1.182, UT: 0d 0h 0m]
10.130.1.181 ( 54)  10.130.1.181 [ eth0]:  10.130.1.181 ( 54)  10.130.1.180 ( 20)
10.130.1.180 ( 49)  10.130.1.180 [ eth0]:  10.130.1.180 ( 49)  10.130.1.181 ( 18)
  Originator (#/255)      Nexthop [outgoingIF]:  Potential nexthops ... [B.A.T.M.A.N. 0.3.2,
MainIF/IP: eth0/10.130.1.182, UT: 0d 0h 0m]
10.130.1.181 ( 59)  10.130.1.181 [ eth0]:  10.130.1.181 ( 59)  10.130.1.180 ( 26)
10.130.1.180 ( 54)  10.130.1.180 [ eth0]:  10.130.1.180 ( 54)  10.130.1.181 ( 18)
```

Al ingresar a la consola de un router Mesh Potato, se podrá ejecutar el monitoreo con solo ingresar los comandos mencionados; sin embargo en el dispositivo móvil es necesario ubicarse a través de la aplicación emulador de terminal en el directorio /data/data/org.servalproject/bin, ya que en este directorio se encuentra el binario batman para el celular que viene incluido en la aplicación serval-batphone-helper.

El batphone también presenta el monitoreo en su interfaz de usuario, sin necesidad de ejecutar el binario desde la consola, lo cual es más amigable que realizarlo desde consola, en la Figura 19 se muestra la aplicación batphone.

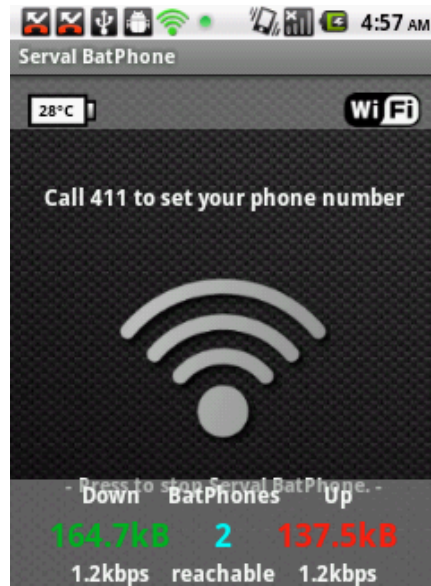


Figura 19. Aplicación Batphone ejecutándose

Puede observarse en la Figura 19 que, al ejecutar las aplicaciones *serval batphone helper* y *serval patched sipdroid* se detectan 2 batphones o dos dispositivos que están ejecutando el protocolo BATMAN.

4.4 DNA Arquitectura de Distribución Numérica.

El DNA cumple la misma función que el protocolo ARP (Protocolo de resolución de direcciones) en los sistemas informáticos, el cual es el responsable de encontrar la dirección de hardware (MAC) que corresponde a una determinada ip.

Una de las desventajas que puede presentar la red inalámbrica es que la cantidad de direcciones ipv4 que se pueden asignar es de 254 números, esto puede ser factible para una pequeña población pero a medida que la red vaya creciendo, se irán incorporando nuevos usuarios y los números se irán agotando. Una solución podría ser implementar ipv6, sin embargo no todos los dispositivos móviles lo soportan.

Por lo tanto se debe de continuar la opción de seguir trabajando con ipv4 y tomar las medidas adecuadas para mitigar el riesgo de colisión de direcciones. Un avance sería en lograr que toda ip representara un número telefónico y de esta manera pasar de 254 direcciones hasta 4,294,977,296 direcciones ip. Esto se puede implementar relacionando a la dirección ip un identificador de usuario y el número telefónico para cada miembro de la red, también conocido como DID

En la última versión de *serval batphone Serval* se implementa este método DNA que no se incluye en las primeras versiones. En el cual a cada miembro se le asigna un ID por usuario (SID), el cual toma la forma de una llave pública. Cada SID y la información asociada es guardada en una pequeña base de datos mantenida por el software en el teléfono móvil. Esta base de datos recibe el nombre de Home Location Register (HLR).

Cada SID tiene una cantidad de números telefónicos (DID) relacionados con sus respectivas localizaciones (usualmente direcciones SIP) por las cuales se puede llegar al batphone con el que se quiere comunicar. Se debe aclarar que la dirección SIP está conformada por la dirección de la central telefónica que en este caso es 4000 seguido de la dirección ip a la que se desea llamar, 4000@ip.

Entonces el proceso de establecer una llamada en una red mesh inicia con un broadcast de DID, es decir, se envía el número telefónico que se ha marcado esperando los correspondientes SID. Al no se recibe ningún SID no se podrá realizar la llamada. De lo contrario se recibirán uno o más SID.

Si un SID es recibido, la llamada continuara por preguntar la ubicación del SID, generalmente una dirección SIP, y luego le pide a Asterisk una conexión a esa ubicación. Al resumir, el protocolo de petición permite DID > SID > resolución de la ubicación de un DID, entonces la respuesta que se reciba contendrá un SID y la información de ubicación.

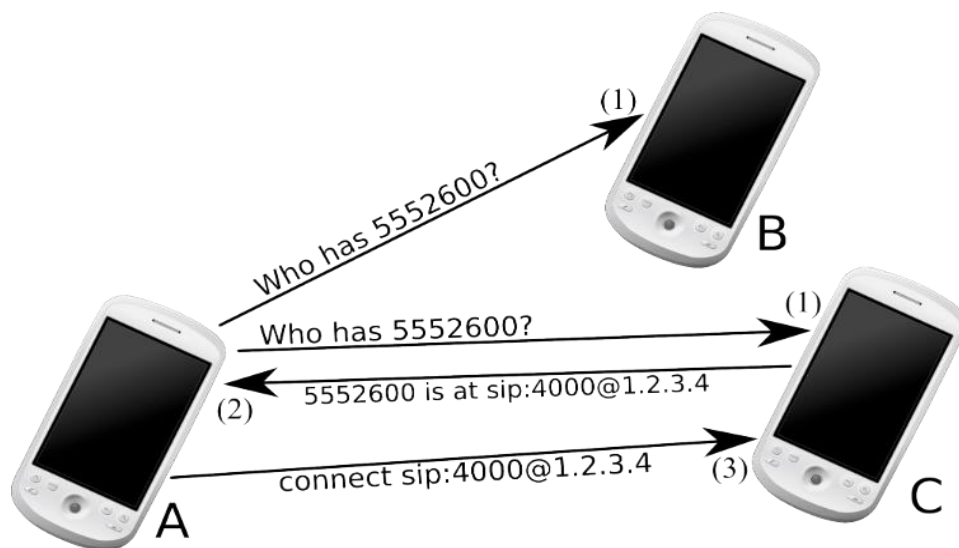


Figura 20. Resolución de una simple llamada usando Serval DNA [23].

En un caso donde se recibe una sola respuesta se ilustra en la figura 20, en la cual un teléfono A desea llamar al dispositivo con número telefónico 5552600. (1)Se inicia enviando un broadcast solicitando la ubicación (dirección SIP) del usuario que está relacionado con ese número. (2)El teléfono C responde con una dirección SIP que puede ser usada para llamar al número. (3)Finalmente, el teléfono A establece la llamada usando la dirección SIP suministrada.

Por lo tanto si se quiere establecer una comunicación entre un dispositivo con la primer versión de batphone y otro con la última versión de batphone, la comunicación no se logrará establecer ya que ambos dispositivos deben ejecutar el DNA.

Sin embargo al querer realizar este mismo procedimiento en una red mesh, todos los routers Mesh Potato deben también ejecutar el Serval DNA.

4.5 Conclusión

Se puede observar el impacto que tiene el batphone dentro de la red Mesh, no solo por comunicarse con los dispositivos MP, sino que también se puede ejecutar en ellos la configuración de cualquier parámetro vía telnet mediante consola o interfaz gráfica para la configuración del router Mesh Potato.

Por otra parte es muy importante considerar la idea de que este proyecto puede implementarse para cualquier tipo de dispositivo móvil sin crear una preferencia o modelo, para que en el futuro si se logra implementar la red mesh en lugares pobres se pueda complementar con la adaptación de la mayor cantidad de batphones que puedan conseguir a bajo costos, porque como se estudio durante este reporte a mayor cantidad de teléfonos mayor será la cobertura de toda la red, y esto es muy importante para no depender de ningún tipo de infraestructura en la comunicación telefónica.

ANEXOS

ANEXO A

A. Versiones de Android.

Desde hace cuatro años la plataforma Android ha venido evolucionado, en cada versión presenta mejoras en sus características. Cada versión en Android ha sido presentada por un postre y estos muestran en orden alfabético, cada una posee sus características, tal como se muestra en la Tabla 9.

VERSION DEL ANDROID	CARACTERISTICAS
Versión 1.0	Lanzada el 23 de septiembre de 2008 con el primer dispositivo móvil el HTC G1, también conocido como HTC DREAM.
Versión 1.1	Lanzada el 9 de febrero de 2009, solo para el dispositivo HTC DREAM. En esta actualización se resolvieron algunos problemas de la versión anterior, y se aplicaron cambios en la API.
Version 1.5 (Cupcake)	Basada en el kernel de Linux, 2.6.27, se incluyeron actualizaciones y nuevas características a la interfaz de usuario. Estas son: <ul style="list-style-type: none">• Posibilidad de grabar y reproducir videos a través del modo camcorder (cámaras digitales grabadoras)• Capacidad de subir videos a YouTube e imágenes a Picasa directamente desde el teléfono• Un nuevo teclado con predicción de texto• Soporte para Bluetooth A2DP (Advanced Audio Distribution Profile, es el perfil estándar para transferir sonido estéreo de alta calidad de una fuente de sonido a un dispositivo Bluetooth, es decir desde el teléfono a audífonos estéreo bluetooth) y AVRCP (Audio/Video Remote Control Profile, una interfaz estándar para el control de televisores y aparatos de música entre otros, de forma que un mando único pueda agrupar todo el control).• Nuevos widgets y carpetas que se pueden colocar en las pantallas de inicio• Transiciones de pantalla animadas
Version 1.6 (Donut)	Basada en el kernel de Linux 2.6.29, en esta se incluye: <ul style="list-style-type: none">• Una experiencia mejorada en el Android Market• Una interfaz integrada de cámara y galería• La galería ahora permite a los usuarios seleccionar varias fotos para eliminarlas• Búsqueda por voz actualizada, con respuesta más rápida y mayor integración con aplicaciones nativas, incluyendo la posibilidad de marcar a contactos• Experiencia de búsqueda mejorada que permite buscar marcadores, historiales, contactos y páginas web desde la pantalla de inicio.• Actualización de soporte para CDMA/EVDO, 802.1x, VPN y

	<p>text-to-speech</p> <ul style="list-style-type: none"> • Soporte para resoluciones de pantalla WVGA • Mejoras de velocidad en las aplicaciones de búsqueda y cámara • Framework de gestos y herramienta de desarrollo GestureBuilder • Navegación gratuita turn-by-turn de Google
<p>Versión 2.0/2.1 (Eclair)</p>	<p>Basado en el kernel de Linux 2.6.29. La versión 2.01 fue lanzada el 3 de diciembre del 2009 y la versión 2.1 el 12 de enero del 2010. Esta actualización contiene:</p> <ul style="list-style-type: none"> • Velocidad de hardware optimizada • Soporte para más tamaños de pantalla y resoluciones • Interfaz de usuario renovada • Nuevo interfaz de usuario en el navegador y soporte para HTML5 • Nuevas listas de contactos • Una mejor relación de contraste para los fondos • Mejoras en Google Maps 3.1.2 • Soporte para Microsoft Exchange • Soporte integrado de flash para la cámara • Zoom digital • MotionEvent mejorado para captura de eventos multi-touch46 • Teclado virtual mejorado • Bluetooth 2.1 • Fondos de pantalla animados
<p>Versión 2.2 (Froyo)</p>	<p>Basado en el kernel de Linux 2.6.32 y lanzada el 20 de mayo del 2010 . Los cambios que incluía son:</p> <ul style="list-style-type: none"> • Optimización general del sistema Android, la memoria y el rendimiento • Mejoras en la velocidad de las aplicaciones, gracias a la implementación de JIT (Just In Time, mejora el rendimiento de las aplicaciones, este tema se observa en las siguiente páginas). • Integración del motor JavaScript V8 del Google Chrome en la aplicación Browser • Soporte mejorado de Microsoft Exchange (reglas de seguridad, reconocimiento automático, GAL look-up, sincronización de calendario, limpieza remota) • Lanzador de aplicaciones mejorado con accesos directos a las aplicaciones de teléfono y Browser • Funcionalidad de Wi-Fi hotspot (funcionar como punto de acceso a internet para otros equipos wifi) y tethering por USB (el dispositivo móvil conectado a través de una interfaz usb a un equipo con acceso a internet ,funciona como modem para otros equipos) • Permite desactivar el tráfico de datos a través de la red del

	<p>operador</p> <ul style="list-style-type: none"> • Actualización del Market con actualizaciones automáticas • Cambio rápido entre múltiples idiomas de teclado y sus diccionarios • Marcación por voz y compartir contactos por Bluetooth • Soporte para contraseñas numéricas y alfanuméricas • Soporte para campos de carga de archivos en la aplicación Browser • Soporte para la instalación de aplicación en la memoria expandible • Soporte para Adobe Flash 10.154 • Soporte para pantallas de alto número de Puntos por pulgada, tales como 4" 720p
<p>Versión 2.3 (Gingerbread)</p>	<p>Basado en el kernel de Linux 2.6.35, lanzado el 6 de diciembre del 2010, presenta las siguientes características:</p> <ul style="list-style-type: none"> • Actualización del diseño de la interfaz de usuario • Soporte para pantallas extra grandes y resoluciones WXGA y mayores • Soporte nativo para telefonía VoIP SIP • Soporte para reproducción de videos WebM/VP8 y decodificación de audio AAC • Nuevos efectos de audio como reverberación, ecualización, virtualización de los auriculares y refuerzo de graves • Soporte para Near Field Communication • Funcionalidades de cortar, copiar y pegar disponibles a lo largo del sistema • Teclado multi-táctil rediseñado • Soporte mejorado para desarrollo de código nativo • Mejoras en la entrada de datos, audio y gráficos para desarrolladores de juegos • Recolección de elementos concurrentes para un mayor rendimiento • Soporte nativo para más sensores (como giroscopios y barómetros) • Un administrador de descargas para descargar archivos grandes • Administración de la energía mejorada y control de aplicaciones mediante la administrador de tareas • Soporte nativo para múltiples cámaras
<p>Versión 3.0/3.1 (Honeycomb)</p>	<p>Basado en kernel Linux 2.6.36, lanzado solo para tabletas en la fecha 22 de febrero del 2011. Esta nueva versión cuenta con las siguientes características:</p> <ul style="list-style-type: none"> • Escritorio 3D con widgets rediseñados • Sistema multitarea mejorado • Mejoras en el navegador web predeterminado, entre lo que destaca la navegación por pestañas, autorelleno de formularios, sincronización de favoritos con Google Chrome y

	<p>navegación privada</p> <ul style="list-style-type: none"> • Soporte para videochat mediante Google Talk • Añade soporte para una gran variedad de periféricos y accesorios con conexión USB: teclados, ratones, dispositivos de juego y cámaras digitales. Cuando un accesorio está conectado, sistema busca la aplicación necesaria y ofrece su ejecución. • Los widgets pueden redimensionarse de forma manual sin la limitación del número de cuadros que tenga cada escritorio.
--	---

Tabla 9. Versiones de la plataforma Android

ANEXO B

B. Identificadores USB de los fabricantes.

Para que la computadora reconozca a los dispositivos móviles conectados a través del puerto USB es necesario instalar los respectivos drivers. En el sistema operativo Windows esta configuración se realiza con solo descargar el paquete Google USB Driver package, opción que se encuentra en el SDK Manager. Sin embargo en el sistema operativo Ubuntu, se debe de configurar un archivo “rules” tal como se explica en la sección 2.9

Los identificadores USB de los fabricantes es el valor que deberá de copiarse en el campo ATTR{idVendor}. Como su nombre lo dice cada fabricante posee un único valor como lo muestra la Tabla 10

Fabricante	Identificador USB
Acer	0502
ASUS	0b05
Dell	413C
Foxconn	0489
Garmin-Asus	091E
Google	18D1
HTC	0BB4
Huawei	12D1
K-Touch	24E3
Kyocera	0482
Lenevo	17EF
LG	1004
Motorola	22B8
NEC	0409
Nook	2080
Nvidia	0955
OTGV	2257
Pantech	10A9
Pegatron	1D4D
Philips	0471
PMC-Sierra	04DA
Qualcomm	05C6
SK Telesys	1F53
Samsung	04E8
Sharp	04DD
Sony Ericsson	0FCE
Toshiba	0930
ZTE	19D2

Tabla 10. Identificadores de USB

BIBLIOGRAFIA.

- [1] *"Sitio oficial de la Open Handset Alliance"* (última visita: 5 de julio de 2011).
http://www.openhandsetalliance.com/oha_members.html
- [2] *"Información sobre el Sistema Operativo Android"* (última visita: 5 de julio de 2011).
<http://es.wikipedia.org/wiki/Android>
- [3] *"Información sobre los nuevos miembros de la Open Handset Alliance"* (última visita: 10 de agosto de 2011).
http://es.wikipedia.org/wiki/Open_Handset_Alliance
- [4] *"Blog de Android"* (última visita: 10 de julio de 2011).
<http://androidandme.com/2011/05/news/comscore-android-still-number-one-rim-will-soon-be-number-three/>
- [5] *"Documentación sobre el Sistema Operativo iOS"* (última visita: 30 de septiembre de 2011).
<http://es.wikipedia.org/wiki/IOS>
- [6] *"Información de los diversos OS para móviles"* (última visita: 01 de noviembre de 2011).
<http://www.tss.com.pe/moviles.php>
- [7] *"Información sobre el Sistema Operativo Windows Mobile"* (última visita: 10 de agosto de 2011).
http://es.wikipedia.org/wiki/Windows_Phone
- [8] *"Información sobre el Sistema Operativo webOS"* (última visita: 10 de agosto de 2011).
http://es.wikipedia.org/wiki/HP_webOS
- [9] *"Información sobre las versiones de Android"* (última visita: 13 de julio de 2011).
<http://getmovil.com/android>
- [10] *"Información sobre la Arquitectura del Sistema Operativo Android"* (última visita: 22 de octubre del 2011).
<http://androideity.com/2011/07/04/arquitectura-de-android/>
- [11] *"Información sobre la Arquitectura del Sistema Operativo Android"* (última visita: 02 de noviembre del 2011).
<http://otrotutorialdeandroid.wikispaces.com/3.1+Bloque+1>
- [12] *"Información sobre el Android Market"* (última visita: 20 de noviembre del 2011).
http://es.wikipedia.org/wiki/Android_Market
- [13] *"Información sobre el HTC DREAM"* (última visita: 12 de julio del 2011).
<http://www.xataka.com/moviles/htc-dream-en-espana-con-movistar>

[14] *"Información de las características técnicas del Huawei IDEOS U8150"* (última visita: 01 de noviembre del 2011).

<http://www.smart-gsm.com/moviles/huawei-u8150-ideos>

[15] *"Introducción a Eclipse"* (última visita: 18 de octubre del 2011)

http://www.atenas.cult.cu/ri/informatica/manuales/sl/introduccion_al_SL/eclipse.html

[16] *"Sitio oficial de Eclipse"* (última visita: 25 de octubre del 2011)

<http://www.eclipse.org/downloads/>

[17] *"Información sobre la instalación del SDK"* (última visita 21 de octubre del 2011)

<http://davidmaestre.com/2011/03/instalando-el-sdk-de-android.html>

[18] *"Información sobre el Emulador de dispositivos Android"* (última visita: 15 de octubre del 2011).

<http://developer.android.com/guide/developing/tools/emulator.html>

[19] *"Información sobre la herramienta adb"* (última visita: 20 de noviembre del 2011)

<http://developer.android.com/guide/developing/tools/adb.html>

[20] *"Sitio oficial de la comunidad Serval"* (última visita: 20 de noviembre del 2011)

<http://www.servalproject.org/about>

[21] *"Código de la aplicaciones Wifi Tether"* (última visita: 20 de noviembre del 2011).

<http://code.google.com/p/android-wifi-tether/>

[22] *"Código de la aplicación Sipdroid"* (última visita: 20 de noviembre del 2011)

<http://sipdroid.org/>

[23] *"Información de la aplicación Batphone"* (última visita: 18 de noviembre del 2011).

http://developer.servalproject.org/site/docs/2011/Serval_Introduction.html#fn17x0