



ACTIVACIÓN DE LA OPCIÓN DOBLE-SIM Y ACTUALIZACIÓN DE ANDROID PARA EL TELÉFONO INTELIGENTE HUAWEI P8 LITE, MODELO ALE-L02

DUAL-SIM OPTION ACTIVATION AND ANDROID UPDATE FOR HUAWEI P8 LITE, MODEL ALE-L02, SMARTPHONE

Yosvany Hervis Santana¹, Glauco A. Guillén Nieto²

1 LACETEL, Cuba, yosvany@lacetel.cu

2 LACETEL, Cuba, glauco@enet.cu

RESUMEN: Se detallan los pasos necesarios para la activación de la opción Doble-SIM y actualización del Sistema Operativo, mediante el reemplazo del Firmware original, para un teléfono inteligente Huawei P8lite, modelo ALE-L02. Se realiza una breve reseña del Sistema Operativo Android y de las herramientas disponibles para el fin propuesto, la estructura del firmware, la secuencia y proceso de actualización.

Palabras Clave: Doble-SIM, Sistema Operativo, Android, Firmware, Huawei, P8lite,

ABSTRACT: The necessary steps for activating the Dual-SIM option and updating the Android Operating System, by replacing the original Firmware, for a Huawei P8lite smartphone, model ALE-L02, are detailed. A brief review of the Android Operating System and the tools available for the proposed purpose, the firmware structure, the sequence and the update process are made.

KeyWords: Dual-SIM, Operating System, Android, Firmware, Huawei, P8lite

1. INTRODUCCIÓN

En el transcurso de la última década el mundo se ha visto envuelto en la era de los teléfonos móviles inteligentes. Cada día los terminales son más sofisticados y la introducción del sistema operativo Android ha contribuido a que una comunidad espontánea de desarrolladores participe aportando soluciones novedosas, en su mayoría no descritas por los fabricantes.

La creación de aplicaciones de los más diversos temas, la actualización y personalización de los sistemas operativos, el cambio de configuración del hardware, entre otros, constituyen contribuciones de esa comunidad.

El objetivo del presente trabajo es habilitar la opción de Doble-SIM en un Huawei P8 lite -cuyo hardware soporta esta opción- y actualizar el Sistema Operativo mediante el reemplazo del firmware. Para ello, se realiza una breve reseña del Sistema Operativo Android y de las herramientas necesarias.

2. CONTENIDO

2.1 Sistema operativo ANDROID

En octubre de 2003, en la localidad de Palo Alto, California, USA, Andy Rubin, Rich Miner, Chris White y Nick Sears fundan Android Inc., con el objetivo de desarrollar un sistema operativo para móviles basado en Linux. El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la crea-

ción de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto [1].

Los terminales con Android no estarían disponibles hasta el año 2008, pero las unidades vendidas de teléfonos inteligentes con Android se ubicaron rápidamente en el primer lugar en los Estados Unidos. A escala mundial alcanzó una cuota de mercado que llegó a ser el doble que la del segundo sistema operativo en el mercado (iOS de Apple, Inc.) [2].

Android tiene una gran comunidad de desarrolladores creando aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se ha superado el millón de aplicaciones disponibles en la tienda de aplicaciones oficial de Android (Google Play), sin tener en cuenta aplicaciones de otras tiendas para Android como la tienda de aplicaciones Samsung Apps, Slideme de Java y Amazon Appstore entre muchas otras.

La estructura del sistema operativo Android, hasta la versión 5.0, se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik. A partir de la siguiente versión cambió al entorno Android Runtime (ART) [3].

Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2.8 millones de líneas de lenguaje C, 2.1 millones de líneas de Java y 1.75 millones de líneas de C++ [4].

2.1.1 Historial de actualizaciones

Android ha visto numerosas actualizaciones desde su liberación inicial. Estas actualizaciones al sistema operativo base típicamente arreglan fallos y agregan nuevas funciones. Generalmente, cada actualización del Sistema operativo Android es denominada, en orden alfabético, bajo un nombre en código relacionado con dulces.

La reiterada aparición de nuevas versiones que, en muchos casos, no llegan a funcionar correctamente en el hardware diseñado para versiones previas, hacen que Android sea considerado uno de los elementos promotores de la obsolescencia programada. Lo mismo sucede con otros Sistemas Operativos.

Android ha sido criticado muchas veces porque sus

terminales no son actualizados constantemente por los distintos fabricantes. Se creyó que esta situación cambiaría tras un anuncio de Google en el que comunicó que los fabricantes se comprometerían a aplicar actualizaciones al menos 18 meses desde su salida al mercado, pero nunca se concretó y el proyecto se canceló. Google actualmente intenta enmendar el problema con su plataforma actualizable: Servicios de Google Play (que funciona en Android 2.2 y posteriores), separando todas las aplicaciones posibles del sistema (como Maps, el teclado, Youtube, Drive, e incluso la propia Play Store) para poder actualizarlas de manera independiente, e incluyendo la menor cantidad posible de novedades en las nuevas versiones de Android.

2.1.2 Arquitectura

Los componentes principales del sistema operativo de Android se muestran en la Figura.1 [3]:

- **Aplicaciones:** las aplicaciones base incluyen un cliente de correo electrónico, un programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Framework (Marco de trabajo) de aplicaciones:** los desarrolladores tienen acceso completo a las mismas API del entorno de trabajo usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del framework de aplicaciones de Android. Algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecutaba, hasta la versión 5.0, archivos en el formato de ejecutable Dalvik (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está ba-

sada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato .dex por la herramienta incluida dx. Desde la versión 5.0 utiliza el ART, que compila totalmente al momento de instalación de la aplicación.

- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema, como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto del software.

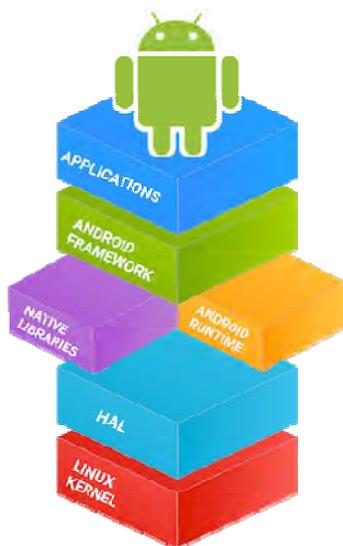


Figura. 1: Arquitectura del sistema Android

2.2 Android Debug Bridge (adb)

Android Debug Bridge (adb) es una herramienta versátil de línea de comandos que permite comunicarse con un dispositivo Android. El comando adb facilita una variedad de acciones con el dispositivo, como instalar y depurar aplicaciones y proporciona acceso a un shell Unix que puede usar para ejecutar comandos en un dispositivo. Es un programa cliente-servidor que incluye tres componentes:

- Un cliente, que envía comandos. El cliente se ejecuta en una máquina de desarrollo. Un *daemon* (adbd), que ejecuta comandos en un dispositivo. El daemon se ejecuta como un proceso en segundo plano en cada dispositivo.
- Un servidor, que gestiona la comunicación entre el cliente y el *daemon*. El servidor se ejecuta como un proceso en segundo plano en la máquina de desarrollo.

2.2.1 Funcionamiento de adb

Cuando inicia un cliente adb, el cliente primero veri-

fica si ya se está ejecutando un proceso de servidor adb. Si no, inicia el proceso del servidor. Cuando el servidor se inicia, se vincula al puerto TCP local 5037 y atiende los comandos enviados por los clientes de adb. Todos los clientes de adb usan el puerto 5037 para comunicarse con el servidor adb.

El servidor luego establece conexiones con todos los dispositivos presentes. Localiza emuladores escaneando puertos impares en el rango de 5555 a 5585, el rango utilizado por los primeros 16 emuladores. Donde el servidor encuentra un adb daemon (adbd), configura una conexión a ese puerto. Cada emulador utiliza un par de puertos secuenciales: un puerto de número par para conexiones de consola y un puerto de número impar para conexiones adb. Por ejemplo:

Emulador 1,	consola:	5554
Emulador 1,	adb:	5555
Emulador 2,	consola:	5556
Emulador 2,	db:	5557

y así...

El emulador conectado a adb en el puerto 5555 es el mismo que el emulador cuya consola escucha en el puerto 5554.

Una vez que el servidor ha configurado las conexiones a todos los dispositivos, puede usar comandos adb para acceder a esos dispositivos. Debido a que el servidor gestiona las conexiones a los dispositivos y maneja los comandos de múltiples clientes adb, puede controlar cualquier dispositivo desde cualquier cliente (o desde un script)[5].

2.2.2 Comandos básicos de adb

adb devices

Este comando permite comprobar si se estableció la comunicación entre la PC y el móvil y como resultado se listan los dispositivos conectados (Figura.2)

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\Minimal ADB and Fastboot>adb devices
List of devices attached
GDU7N16C19000240    device

C:\Program Files (x86)\Minimal ADB and Fastboot>
```

Figura. 2: Dispositivos listados usando el comando adb devices.

adb push

Este comando permite copiar un archivo directa-

mente desde la consola al dispositivo (teléfono).

Ejemplo: adb push "mi video.mp4" /sdcard/videos/

adb reboot. Este comando resulta obvio, envía la orden de reinicio al terminal.

fastboot devices: Listar dispositivos conectados en modo HBOOT.

fastboot reboot : Reinicio normal.

fastboot reboot-bootloader. Reinicio en modo HBOOT.

fastboot flash "partición" "archivo.img" (*): Flashear una partición.

fastboot oem unlock contraseña: Desbloquear bootloader.

En este último comando la contraseña es requerida por algunos dispositivos, como es el caso de los terminales Huawei.

Estos comandos son los necesarios para cambiar el firmware del dispositivo.

El set completo de los comandos de adb y su descripción se puede obtener empleando:

Adb help: Lista los comandos para trabajar en modo adb.

Fastboot help: Lista los comandos para trabajar en modo fastboot.

2.3 Bootloader

En su nivel más básico, el almacenamiento en un teléfono inteligente Android es como en un disco duro, compuesto por varias particiones. Una de esas particiones contiene los archivos del propio sistema Android, otra contiene todos los datos de las aplicaciones de usuario (lo que permite actualizar el sistema Android sin perder la información de usuario) y otros para ejecutar diversas operaciones en segundo plano.

El bootloader actúa como un punto de control de seguridad y administrador para todas esas particiones. Cada computadora tiene uno, y es lo que le indica al hardware dónde buscar y cómo ponerse en marcha. En los teléfonos Android (y tabletas, cajas de TV e incluso hornos de microondas), el bootloader realiza determinadas comprobaciones de forma predeterminada, para asegurarse de que el software que intenta iniciar sea genuino. Así ocurre en la mayoría de los dispositivos móviles pues, si puede cambiarse el contenido original de esas particiones, existe el riesgo de que el dispositivo resulte averiado.

Un bootloader desbloqueado no verifica que el software sea genuino y permite la instalación y el uso de software personalizado en la ROM. Esta es la razón por la que puede existir interés en desbloquearlos.

Todos los teléfonos Android se comercializan con el bootloader bloqueado. En la mayoría de los casos el gestor de arranque está encriptado, como medida de seguridad. Algunos dispositivos se desbloquean fácilmente con un solo comando, pero los que tienen el bootloader encriptado, necesitan un "token" de seguridad específico (entiéndase como una contraseña) para desbloquearlos.

2.4 Condiciones necesarias para cambiar el contenido de la ROM

Para realizar cualquier cambio en el contenido de la ROM de un teléfono, por ejemplo Huawei, es requisito indispensable tener el bootloader desbloqueado. Para esta operación es necesario introducir varios datos en el sitio del fabricante <https://emui.huawei.com/en/> para obtener la contraseña correspondiente a un terminal específico. La Figura.3 muestra los datos necesarios que se demandan para obtener la contraseña.

Figura.3: Datos necesarios para obtener la contraseña del bootloader

Para desbloquear el bootloader debe ponerse el teléfono en modo fastboot: apagado y presionando volumen (-), se conecta el teléfono a la PC. Mantener presionado volumen (-) hasta que aparezca una imagen como la mostrada en la Figura.4a.

Una vez en este modo introducimos el comando **fastboot oem unlock "contraseña"** (obtenida previamente en el sitio web). En la Figura.4b se muestra el teléfono después de aplicado el comando anterior.

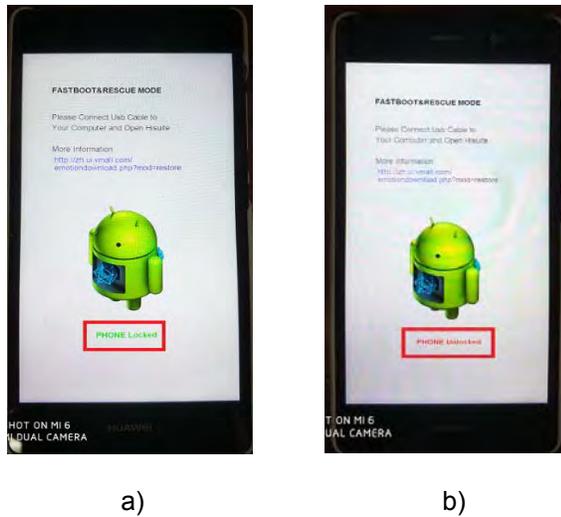


Figura. 4: Modo Fastboot. a) Bootloader bloqueado. b) Bootloader desbloqueado

2.5 Cambio del firmware para activación de la opción Doble-SIM y actualización del SO.

Es importante tener presente que solo es posible activar la opción Doble-SIM si el hardware del dispositivo lo soporta.

Lo anterior es condición necesaria pero no suficiente. Es necesario además que el firmware soporte esta opción.

Las premisas anteriores se cumplen en el Huawei P8 lite modelo ALE-L21: mismo hardware que el ALE-L02 y la opción Doble-SIM disponible.

2.5.1 Número de compilación

Accediendo a **Ajustes/Acerca del teléfono** se obtiene el número de compilación, en este caso particular es ALE-L02C403B150, donde:

ALE-L02 - se corresponde con el modelo del teléfono.

C403 – se corresponde a la zona geográfica para la cual fue fabricado el teléfono.

B150 – se corresponde con la versión de firmware que tiene instalado el teléfono.

Cada nueva versión de firmware puede incluir configuraciones del hardware que no necesariamente se repiten en las versiones siguientes. Esto deja dos alternativas: 1) actualizar secuencialmente cada una de las versiones de firmware liberadas o 2) asegurarse de que las elegidas contengan todas las configuraciones necesarias.

En el presente material se han identificado cuatro versiones de firmware que cumplen la última condición señalada.

2.5.2 Estructura del Firmware

Con el empleo de la herramienta Huawei Update Extractor v0.9.9.3 es posible conocer los componentes del firmware, como se muestra en la Figura.5.

Sequence	File	Size	Type
0000000C	BOOT.img	22.04 MB	BOOT
0000000E	CACHE.img	6.10 MB	CACHE
FF000000	CRC.img	174.68 KB	CRC
FFFFFFF0	CURVER.img	16 B	CURVER
0000000F	CUST.img	239.95 MB	CUST
00000008	DSP.img	2.14 MB	DSP
00000008	DTIMAGE.img	7.66 MB	DTIMAGE
00000000	EFI.img	17.00 KB	EFI
00000008	FASTBOOT.img	1.22 MB	FASTBOOT
00000012	FASTBOOT1.img	116.56 KB	FASTBOOT1
00000005	HIFI.img	2.47 MB	HIFI
00000004	MCUIIMAGE.img	144.94 KB	MCUIIMAGE
00000001	MODEMIMAGE.img	40.06 MB	MODEMIMAGE
00000002	MODEMNM1.img	10.21 MB	MODEMNM1
0000000A	RECOVERY.img	30.66 MB	RECOVERY
00000013	RECOVERY2.img	30.66 MB	RECOVERY2
FE000000	SHA256RSA.img	256 B	SHA256RSA
0000000D	SYSTEM.img	2.30 GB	SYSTEM
FFFFFFF5	TEEOS.img	676.25 KB	TEEOS
FFFFFFF6	TRUSTFIRMWARE...	46.56 KB	TRUSTFIRMW...
00000010	USERDATA.img	42.86 MB	USERDATA
FFFFFFF1	VERLIST.img	794 B	VERLIST

Figura. 5: Archivos que componen el sistema operativo

De ellos:

BOOT: Ejecuta el proceso de inicio del terminal.

RECOVERY: Garantiza las sucesivas actualizaciones del terminal.

SYSTEM: Es el sistema operativo en sí mismo.

CUST: Es la personalización del SYSTEM.

Estos cuatro componentes constituyen la base del Sistema Operativo.

Un firmware desarrollado para un modelo determi-

nado no es aceptado como actualización en otro modelo. Sin embargo, es posible extraer los componentes señalados de un firmware desarrollado para un modelo e insertarlos en el firmware de otro, siempre que se trate de un mismo hardware.

2.5.3 Instalación del Firmware

Se ha identificado que para el propósito propuesto deben instalarse 4 firmwares, el ALE-L21C432B046b, el B052, el B190 y el B550.

Del primero de ellos, el B046b, se extraen los componentes señalados en el epígrafe anterior y se depositan en la misma carpeta que contiene las herramientas adb.exe y fastboot.exe en la PC.

El terminal debe conectarse a la PC mientras se mantiene presionado "volumen menos" para entrar al modo Fastboot.

En estas condiciones es posible insertar los componentes mencionados en el firmware original del terminal ALE-L02, mediante la secuencia de comandos siguiente:

```
mfastboot flash boot boot.img
```

```
mfastboot flash recovery recovery.img
```

```
mfastboot flash cust cust.img
```

```
mfastboot flash system system.img
```

Una vez concluido este proceso el terminal se reiniciará, ahora convertido en un terminal ALE-L21 Doble-SIM, con sistema operativo Android 5.0.

Las restantes tres versiones de firmware: B052, el B190 y el B550, se instalan en ese orden, siguiendo el procedimiento descrito en [6]. Con ello, quedará actualizada la versión del Sistema Operativo a Android 6.0.

3. CONCLUSIONES

Se identificaron las características principales y componentes del sistema operativo Android y las herramientas necesarias para interactuar con los terminales.

Se confirmó la coincidencia de hardware entre los modelos ALE-L02 y ALE-L21 del Huawei P8 lite y la factibilidad de emplear el firmware del primero en el segundo.

Se señalan las versiones de firmware necesarias y se describe el proceso para convertir un Smartphone Huawei P8 lite modelo ALE-L02 en ALE-L21, con

la funcionalidad Doble-SIM y Sistema Operativo Android 6.0.

4. REFERENCIAS BIBLIOGRÁFICAS

1. Hernández, H. *La historia de Android: de Apple Pie 1.0 a Nougat 7.0* 2016; Available from: <http://www.malavida.com/es/analisis/la-historia-de-android>.
2. Whitney, L. *Android hits top spot in U.S. smartphone market*. 2010; Available from: <https://www.cnet.com/news/android-hits-top-spot-in-u-s-smartphone-market/>.
3. Steele, J. and N. To, *The Android Developer's Cookbook Building Applications with the Android SDK* 2011: Pearson Education, Inc.
4. Tracker, A.P. *The Android Source Code* 2017; Available from: <https://source.android.com/setup/>.
5. Báez, M., et al., *Introducción a Android* 2013. 121.
6. Huawei Technologies Co., L., *HUAWEI ALE-L21C432B550 Software Upgrade Guideline*, 2016.

5. SÍNTESIS CURRICULARES DE LOS AUTORES



MSc., Ing., Yosvany Hervis Santana: MATANZAS, 26 DE AGOSTO DE 1988. Ingeniero en Telecomunicaciones y Electrónica, CUJAE, La Habana, Cuba, 2012. Graduado de Master en Sistemas Digitales, CUJAE, La Habana, Cuba, 2017. Posición Actual: Investigador en LACETEL®, Instituto de Investigación y Desarrollo en Telecomunicaciones.



Dr., Ing., Glauco Antonio Guillén Nieto: LA HABANA, 17 DE OCTUBRE DE 1961. Ingeniero en Radiocomunicación y Radodifusión, Instituto Electrotécnico de Comunicaciones, Odessa, Ucrania, URSS, 1985. Doctor en Ciencias Técnicas: "Teoría de las Comunicaciones, Sistemas y Dispositivos de Transmisión de Información por Canales de Comunicación" y "Sistemas y Dispositivos Radiotécnicos y de Televisión"; Instituto Electrotécnico de Comunicaciones, Odessa, Ucrania, URSS, 1989. Posición Actual: Investigador Titular y Director General, en LACETEL®, Instituto de Investigación y Desarrollo en Telecomunicaciones; Académico Titular de la Academia de Ciencias de Cuba.