Instituto Politécnico Nacional

Centro de Investigación en Computación
Laboratorio de Microtecnología y Sistemas Embebidos

Una Infraestructura de Comunicaciones Cliente-Servidor para Aplicaciones Móviles

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA INGENIERÍA DE CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES

P R E S E N T A

ING. ALEJANDRO SANTANA OROZCO

DIRECTOR (ES) DE TESIS

DR. Luis Alfonso Villa Vargas

DR. Víctor Hugo Ponce Ponce

MÉXICO, D.F. ENERO DE 2011

Índice General

INTRODUCCIÓN 1
1.1 Comunicaciones Móviles 2
1.1.1 Aplicaciones Móviles 2
1.1.2 Infraestructura de redes inalámbricas 3
1.1.3 Enlaces de comunicación en una Red Ethernet Inalámbrica 4
1.1.4 Enlaces de comunicación en una Red Celular 5
1.2 Planteamiento del problema 7
1.3 Justificación 8
1.4 Objetivos 9
1.4.1 Objetivo General 9
1.4.2 Objetivos Específicos: 9
1.5 Organización del trabajo 10
ESTADO DEL ARTE 11
2.1 Arquitecturas de dispositivos Móviles 12
2.1.1 Arquitectura digital móvil, Mobydick 15
2.1.2 Procesador Nomadik STN8800 de STMicroelectronics 16
2.1.3 Procesador PXA270 de Intel 16
2.1.4 Procesador Dragonball MC9328MX21 de Freescale 17
2.1.5 Otras Arquitecturas 21
2.2 Dispositivo Móvil 22
2.2.1 Tipos de Dispositivos Móviles 23
2.2.1.1 PDA 23
2.2.1.2 Teléfono Móvil 24
2.2.1.3 Smartphone 24
2.3 Sistemas operativos para dispositivos Móviles 24
2.3.1 PALM OS 25
2.3.2 Symbian 25
2.3.3 Windows Mobile 25
2.3.4 Iphone OS 26
2.3.5 Android 26
2.3.6 BlackBerry OS 27
2.4 Comparación de Sistemas Operativos para Móviles 27
2.5 Conclusiones 28
MARCO TEÓRICO 29
3.1 Redes Inalámbricas 30
3.1.1.1 Manera en que trabaja el estándar 802.11 31
3.1.1.2 Arquitectura del estándar IEEE 802.11 32
3.1.1.3 Handoff 32
3.1.1.4 Servicios del Estándar IEEE 802.11 33
3.1.1.4.1 Relacionados con Asociación 33
3.1.1.4.2 Relacionados con Distribución 33
3.1.1.4.3 Relacionados con Privacidad y Acceso 34
3.1.1.5 Comunicación con Punto de Acceso 34

3.1.1.5.1 Paquetes de Administración 35
3.1.1.5.2 Paquetes de Control 35
3.1.2 Red Celular WWAN 36
3.1.2.1 Evolución de las Redes celulares 36
3.1.2.1.1 Red Celular de 1G 36
3.1.2.1.2 Red Celular de 2G 36
3.1.2.1.3 Red Celular de 2.5 G 37
3.1.2.1.4 Red Celular de 3G 37
3.1.2.1.5 Red Celular de 4G 38
3.1.2.2 Forma en que trabaja un Teléfono Celular 38
3.1.2.2.1 Handover 39
3.1.2.3 Red GSM 39
3.1.2.3.1 Arquitectura de una red GSM 39
3.1.2.4 Red GPRS 42
3.1.2.4.1 Características de GPRS 43
3.1.2.5 Red EDGE 44
3.2 La plataforma Android 45
3.2.1 Arquitectura empleada en Android 46
3.2.2 Componentes de una Aplicación en Android 48
3.2.2.1 Activity 49
3.2.2.2 Broadcast Intent Receiver 49
3.2.2.3 Service 50
3.2.2.4 Content Provider 50
3.2.3 Ciclo de vida de una aplicación en Android 50
3.2.4 Seguridad en Android 52
3.2.5 El emulador 53
3.2.6 Instalación del Entorno para desarrollar Aplicaciones 54
3.2.6.1 SDK de Android 54
3.2.6.2 Eclipse 54
3.2.6.3 Plug-in de Android 54
3.2.6.4 Referenciar el SDK de Android 55
3.2.7 Contenido de un proyecto de Android 55
3.2.7.1 Carpeta \src 55
3.2.7.2 Carpeta \res 56
3.2.7.3 Carpeta \bin 57
3.2.7.4 Archivo AndroidManifest.xml 57
3.3 Conclusiones 58
DISEÑO 59
4.1 Diseño de la Arquitectura para dispositivo móvil con conectividad Wi-Fi y GPRS 60
4.1.1 Descripción de la arquitectura 60
4.2 Módulos para la Integración de Hardware 61
4.2.1 Módulo de evaluación OMAP35x EVM 61
4.2.2 Tarjeta WL1271 63
4.2.3 Modem Enfora GSM/GPRS EDG0308-11 64

4.3 Software para hacer más eficiente las comunicaciones.....65
4.3.1 Análisis y diseño de la aplicación para una Comunicación Eficiente 67
4.3.1.1 Introducción a “ComEficiente” 67
4.3.1.1.1 Casos de uso.....67
4.3.1.1.2 Modelo de clases.....69
4.3.1.1.3 Arquitectura 70
4.4 Conclusiones.....71
PRUEBAS Y RESULTADOS 72
5.1 Desarrollo de la arquitectura de cómputo móvil 73
5.1.1 Creación Del PCB.....73
5.1.1.1 Conectores para el procesador OMAP.....73
5.1.1.2 Conector SD/MMC 74
5.1.1.3 Interfaz LCD 74
5.1.1.4 Distribución de potencia 75
5.1.1.5 Conectores del TWL4030 (Modulo Administrador de Potencia) 75
5.1.1.6 Interfaz GSM/GPRS 75
5.2 Aplicación básica en Android 76
5.2.1 Creación de un nuevo proyecto en Android.....76
5.2.2 Aplicación básica usando el código XML.....80
5.3 Instalación del Hardware.....81
5.3.1 Instalación de un programa terminal.....81
5.3.1.1 Instalación de Minicom.....82
5.3.2 Modificaciones del OMAP.....84
5.3.3 Preparando la tarjeta OMAP para correr el sistema operativo.....85
5.3.4 Preparando la tarjeta SD para iniciar el sistema operativo 86
5.4 Descargar aplicaciones al OMAP.....87
5.5 Corriendo una aplicación en el OMAP 91
5.6 Compilando el Kernel.....94
5.7 Instalación del sistema operativo Android en una computadora portátil 97
5.8 Plataforma TAO 100
5.9 Desarrollo de la aplicación “ComEficiente” 101
5.9.1 Pruebas antes de la aplicación final 101
5.9.2 Captura de paquetes Beacon de los Puntos de Acceso.....102
5.9.3 Conexión a un Punto de Acceso.....104
5.10 Conclusiones.....105
CONCLUSIONES Y TRABAJOS FUTUROS 107
6.1 Conclusiones finales 108
6.1.1 Conclusión sobre los Objetivos Específicos: Software 108
6.1.2 Conclusión sobre los Objetivos Específicos: Hardware.....108
6.1.3 Conclusión sobre el Objetivo General:.....109
6.2 Trabajos Futuros.....109
Referencias Bibliográficas.....111

Referencias Electrónicas **112**
Abreviaciones **113**
Abreviaciones en ingles **115**
Términos **115**
ANEXO 1 ..... **117**
ANEXO 2 ..... **118**
ANEXO 3 ..... **119**
ANEXO 4 ..... **120**
ANEXO 5 ..... **121**
ANEXO 6 ..... **122**

Índice de figuras

Figura 2.1 Procesador PXA270 de Intel..... 17
Figura 2.2 Procesador Dragonball MC9328MX21 de Freescale..... 18
Figura 2.3 La estructura de hardware del teléfono inteligente..... 21
Figura 3.1 Simple configuración WLAN..... 30
Figura 3.2 Modo infraestructura del estándar 802.11..... 32
Figura 3.3 Evolución a 3G..... 38
Figura 3.4 Arquitectura de la Red GSM..... 41
Figura 3.5 Arquitectura de Red GPRS..... 44
Figura 3.6 Arquitectura empleada en Android..... 46
Figura 3.7 Ciclo de vida del componente Activity..... 51
Figura 3.8 Emulador de Android..... 53
Figura 3.9 Workspace de Eclipse..... 55
Figura 3.10 Carpeta “src”..... 56
Figura 3.11 Carpeta “res”..... 57
Figura 4.1 Diagrama de bloques de la arquitectura propuesta..... 61
Figura 4.2 Parte superior del OMAP 35x EVM..... 62
Figura 4.3 Parte inferior del OMAP 35x EVM..... 63
Figura 4.4 Parte superior de la tarjeta hija WL1271..... 63
Figura 4.5 Parte inferior de la tarjeta hija WL1271..... 64
Figura 4.6 Módem GSM/GPRS de Enfora..... 64
Figura 4.7 Cómo hacer más eficiente las comunicaciones..... 65
Figura 4.8 Diagrama de flujo que muestra el problema a tratar..... 66
Figura 4.9 Caso de uso asociado a el usuario..... 67
Figura 4.10 Caso de uso asociado a la aplicación..... 68
Figura 4.11 Modelo de clases..... 69
Figura 4.12 Arquitectura general del sistema completo..... 71
Figura 5.1 Conectores del módulo del procesador OMAP y su footprint..... 74
Figura 5.2 Conector SD/MMC y su footprint..... 74
Figura 5.3 Conector para el LCD y su footprint..... 74
Figura 5.4 Conectores del Módulo TWL4030 y su footprint..... 75
Figura 5.5 Conector del Módem GSM/GPRS y su footprint..... 75
Figura 5.6 Socket del SIM y su footprint..... 76
Figura 5.7 Ejecución del Ambiente de desarrollo Eclipse..... 76
Figura 5.8 Creación de un nuevo proyecto..... 77

Figura 5.9 Configuración del nuevo proyecto…….. 78
Figura 5.10 Aplicación “Hello, Android” corriendo en el simulador…….. 79
Figura 5.11 Consola corriendo el programa terminal Minicom……. 82
Figura 5.12 Selección del puerto serie 1 en la tarjeta OMAP…… 83
Figura 5.13 Mensajes de inicialización del OMAP en consola usando Minicom…... 83
Figura 5.14 Parte inferior de la tarjeta OMAP Rev-G…….. 84
Figura 5.15 Parte inferior de la tarjeta OMAP Rev-G…….. 84
Figura 5.16 Tarjeta SD e Interfaz de la tarjeta OMAP usada para cargar el sistema operativo…….. 85
Figura 5.17 DIP Switch 4 usado para controlar el modo de inicio del procesador… 85
Figura 5.18 Archivos en la memoria SD para correr un sistema operativo…… 86
Figura 5.19 Sistema operativo Android sobre la plataforma OMAP…… 87
Figura 5.20 Primer ajuste para correr las aplicaciones en el OMAP……... 88
Figura 5.21 Salida del comando “lsusb”…….. 88
Figura 5.22 Configuración del sistema operativo Android para correr aplicaciones en el OMAP…….. 89
Figura 5.23 Computadora de desarrollo y el OMAP conectados vía el puerto USB y USB-OTG…….. 89
Figura 5.24 Ejecución del comando “lsusb”…….. 89
Figura 5.25 Segundo ajuste para correr las aplicaciones en el OMAP……... 90
Figura 5.26 Muestra que el OMAP ha sido reconocido por el ambiente de desarrollo Eclipse…… 90
Figura 5.27 Elección manual del dispositivo donde se correrá la aplicación……. 91
Figura 5.28 Elección del OMAP como dispositivo para correr la aplicación…….. 92
Figura 5.29 Aplicación corriendo en el OMAP…… 92
Figura 5.30 Elección de un simulador del sistema operativo Android para correr la aplicación…… 93
Figura 5.31 Aplicación corriendo en el simulador de Android……... 93
Figura 5.32 Directorio de la herramienta de compilación cruzada…….. 94
Figura 5.33 Variable de entorno de la herramienta de compilación cruzada…… 94
Figura 5.34 Salida del comando “distclean” ……..…… 95
Figura 5.35 Ejecución en consola del comando para crear las fuentes del Kernel… 95
Figura 5.36 Salida del comando “menuconfig”……..…….. 96
Figura 5.37 Ejecución del comando “uImage”…….. 97
Figura 5.38 Directorio donde se creó la imagen del Kernel……... 97
Figura 5.39 Imagen ISO……... 98
Figura 5.40 Herramienta UNetbootin…….. 98
Figura 5.41 Ordenador iniciando Android…….. 99
Figura 5.42 Android corriendo en un ordenador con procesador ATOM…… 99
Figura 5.43 Usando el hardware Wi-Fi del ordenador…… 99
Figura 5.44 Procesador TAO……..... 100
Figura 5.45 Vista inferior de la tarjeta “Thunder”…… 100
Figura 5.46 Tarjeta “Thunder” con procesador TAO corriendo el Sistema Operativo Android…….. 101
Figura 5.47 Muestra que la interfaz Wi-Fi ha sido habilitada…….. 102
Figura 5.48 Paquetes “Beacon”…… 102
Figura 5.49 Tarjeta “Thunder” con procesador TAO corriendo la aplicación “ComEficiente”…… 105

Figura 5.50 Ordenador corriendo la aplicación “ComEficiente”……..... 105

Índice de tablas

Tabla 2.1 Especificaciones de teléfonos inteligentes, su conectividad a redes incluye: Wi-Fi, UMTS, WCDMA, HSDPA, GSM/GPRS/EDGE y Bluetooth…… 14
Tabla 2.2 Medidas comparativas entre procesadores multimedia…….. 19
Tabla 2.3 Comparativa entre sistemas operativos para móviles……. 27
Tabla 3.1 Versiones del estándar IEEE 802.11…….. 31
Tabla 4.5 Clases usadas en el diseño…….. 70
Tabla 5.1 Posición del DIP switch 4 para iniciar desde la interfaz MMC……... 86

Índice de códigos

Código 5.1 Clase principal “HelloAndroid”…… 78
Código 5.2 Archivo XML creado directamente con código fuente……..... 79
Código 5.3 Código XML alternativo de la aplicación anterior……... 80
Código 5.4 Contenido del archivo “strings.xml”…….. 81
Código 5.5 Modificación de la clase “HelloAndroid”……. 81
Código 5.6 Código que prende la interfaz Wi-Fi si está apagada y la apaga si está encendida…… 101
Código 5.7 Muestra el uso de la clase “WifiManager” de Android…….. 103
Código 5.8 Muestra el uso de la clase “WifiConfiguration” de Android……. 104

ORGANIZACIÓN

La organización del presente trabajo se divide en 6 capítulos, a continuación se comenta el contenido de los mismos:

Capitulo 1. Introducción. En este capítulo se da una introducción al tema de investigación, se define el planteamiento del problema, la justificación, el objetivo general, los objetivos específicos y la organización del trabajo para lograr el desarrollo de la tesis.

Capítulo 2. Estado del Arte. En este capítulo se ofrece una descripción general de algunos de los aspectos técnicos de este proyecto como las Arquitecturas de dispositivos Móviles actuales, los sistemas operativos más utilizados para éstos y las características de los tipos de Dispositivos Móviles existentes hoy en día.

Capítulo 3. Marco Teórico. En este capítulo se explican las características de las redes inalámbricas Wi-Fi y Redes Celulares así como la forma en que un dispositivo móvil accede a éstas. Se explica al lector las características básicas del nuevo sistema operativo Android, su diseño, arquitectura y funcionamiento, así mismo se presenta una guía de instalación para el entorno de desarrollo con Eclipse.

Capítulo 4. Diseño del Sistema. Dentro de este capítulo se muestran los diagramas de bloques de los esquemas propuestos: software para hacer más eficientes las comunicaciones y una arquitectura para dispositivo móvil con conectividad Wi-Fi y GPRS y la descripción de cada uno de ellos.

Capítulo 5. Pruebas y Resultados. Describe de forma detallada el completo desarrollo de una aplicación para Android: desde su diseño y funcionalidad, hasta la implementación de cada una de sus clases y componentes básicos, El desarrollo de la arquitectura de dispositivo móvil su diagrama esquemático y su PCB, La instalación y compilación del sistema operativo Android para la plataforma OMAP, TAO y ordenadores x86.

Capítulo 6. Se exponen los resultados obtenidos tras la finalización del proyecto, comparándolos con los objetivos marcados inicialmente, y trazando posibles líneas futuras de desarrollo.

Por último, se incluyen una serie de anexos donde se puede encontrar el diseño esquemático de la arquitectura de un dispositivo móvil.

Resumen

Los dispositivos móviles son equipos de cómputo completos construidos en un solo circuito impreso, su diseño está centrado en sistemas de un procesador o multiprocesador con memoria RAM, puertos de E/S y todas las características necesarias para hacer una computadora funcional en un circuito impreso reducido. Estos sistemas constituyen una oportunidad de desarrollo tecnológico para ofrecer a los usuarios, en un mismo y reducido sistema, funciones de comunicación y procesamiento de datos que van más allá de las simples llamadas telefónicas o de la ejecución de aplicaciones básicas. Entre las funciones de comunicación encontramos dispositivos con conexión a Internet vía Wi-Fi, Red Celular y otros tantos con soporte de GPS. Las opciones de cómputo son variadas, desde procesadores básicos, Procesadores de Señales (DSP's), Procesadores vectoriales para manejo de gráficos y multinúcleos heterogéneos. Los sistemas de E/S generalmente definen la aplicación y van desde pantallas, teclados, pantallas táctiles, GPS, Sensores Industriales: presión, humedad, Ph, temperatura, etc. Sensores Clínicos: Presión, Temperatura, Flujo, etc. biosensores y biodetectores.

En esta tesis se analizan las características y arquitecturas de los dispositivos móviles actuales, se propone una arquitectura para computo móvil usando como componente principal el procesador OMAP de Texas Instruments, y se propone una infraestructura de comunicaciones Cliente-Servidor la cual consta de módulos de hardware como un módem GPRS y un módem Wi-Fi. Adicionalmente se presenta el desarrollo completo de una aplicación para el Sistema Operativo Android de Google para éste tipo de dispositivos. La aplicación optimiza el costo de las comunicaciones permitiendo al usuario hacer conexiones a Internet usando el hardware del dispositivo, la conexión a Internet será en primera instancia intentada por vía Wi-Fi y después vía GPRS.

Abstract

Mobile devices are full computers built in a single printed board; its design is based on uniprocessors or multiprocessors systems with RAM, I/O ports and all necessary features to make a functional computer in a single compacted printed board called *Single Board Computer*. These systems are becoming an opportunity of technology development to offer the user, on the same and small device, communications functions and data processing that are further beyond phone calls or running basic applications. In the range of communications functions we can find devices with an Internet connection via either Wi-Fi or Cellular Networks and others with support for GPS. The computing options are several, from basic processors, Digital signal processors, Vector Processors for graphics applications and Heterogeneous multicores. In general, I/O systems can define the applications; we can find devices as video screen, keyboard, touch-screen, GPS, Industrial sensors: Pressure, Humid, Ph, etc. and Medical Sensors: Temperature, Flow, bio-sensors and bio-detectors, etc.

This thesis studies the characteristics and architectures of new mobile devices, a SBC design for this kind of devices using as main component the OMAP processor of Texas Instruments is developed and a Client-Server communications infrastructure is proposed, which has both hardware communication modules, a GPRS and a Wi-Fi modem. Additionally, it covers the complete development of an application for the Android operating system of Google, for this kind of devices. The application optimizes the communications cost by allowing the SBC to get Internet connections using both hardware devices, connection is firstly intended via Wi-Fi and then via GPRS.

CAPÍTULO 1

INTRODUCCIÓN

En este capítulo se hace una breve Introducción a este proyecto, se describe de forma general las tecnologías utilizadas en las comunicaciones móviles. Se expone la motivación de este trabajo, los objetivos que persiguen y los resultados que se esperan de esta tesis. Al final se describe la forma en que se ha organizado la tesis.

1.1 Comunicaciones Móviles

La mejor forma en que diversos dispositivos pueden comunicarse es a través de una red de comunicaciones, estas redes hacen que muchos dispositivos puedan compartir cualquier tipo de información. Las comunicaciones para aplicaciones móviles se dan con tecnologías inalámbricas, también conocidas por su nombre en inglés *Wireless.* La tecnología inalámbrica es producto de la necesidad de aplicaciones en movimiento, por tanto el factor movilidad es uno de los principales aspectos a tratarse en las redes de comunicaciones. La transmisión y recepción de datos a través del aire tiene características como: la frecuencia de emisión (número de oscilaciones por segundo de la onda electromagnética) y la longitud de onda. Una antena convenientemente fijada a un módulo de comunicación puede emitir ondas electromagnéticas a frecuencias variadas y puede recibirlas a determinadas distancias.

Una red de computadoras se comunica con otra red de computadoras mediante enlaces punto a punto; los encargados de realizar este tipo de comunicación son los dispositivos de interconexión de redes, éstos se encargan de dirigir los paquetes desde un dispositivo de la red a uno o más dispositivos que están conectados mediante un enlace. De esta forma se pueden generar redes de computadoras de enorme escala y en función a la escala estas redes se distinguen en 3 grupos:

Redes de Área Local (LAN, Local Area Network): Agrupaciones de dispositivos que comparten información entre dispositivos que se encuentran localizados en no más de un metro de longitud uno del otro (una red de un campus universitario o de una pequeña empresa). Redes de Área Metropolitana (MAN, Metropolitan Area Network): Asociada con la dimensión de una ciudad, es decir, unos diez kilómetros. Redes de Área Extendida (WAN, Wide Area Network): Red que puede abarcar un país, incluso un continente. Todas ellas pueden o no estar conectadas a la red de redes, llamada Internet la cual abarca todo el planeta, y por tanto supone un paso más allá de las redes WAN. [1]

1.1.1 Aplicaciones Móviles

Las aplicaciones para sistemas de comunicaciones móviles cliente-servidor son muchas y variadas y dependen del ingenio de los desarrolladores. Sólo por citar algunas podemos encontrar: voz y video sobre IP**,** sistemas de monitoreo remoto, medidores de energía para agua o gas, sensores médicos personalizados, sistemas de seguridad personal, redes de comunicación privadas, puntos de venta, entre otros.

Con este tipo de comunicación un usuario puede enviar y recibir información cuando se encuentre en movimiento. El usuario debe ser capaz de enlazarse a Internet vía GPRS (*General Packet Radio Service*) mediante telefonía móvil o vía Ethernet mediante puntos de acceso inalámbricos de una LAN o una MAN.

1.1.2 Infraestructura de redes inalámbricas

Las Redes Inalámbricas facilitan la operación en lugares donde los dispositivos no puede permanecer en un solo lugar, según su cobertura, se pueden clasificar en: WPAN (Wireless Personal Área Network), WLAN (Wireless LAN), WMAN (Wireless MAN) y WWAN (Wireless WAN).

WLAN: Usa tecnologías inalámbricas basadas en Hiper LAN (High Performance Radio LAN), un estándar del grupo ETSI (Instituto Europeo de los Estándares de las Telecomunicaciones), o tecnologías basadas en Wi-Fi, que siguen el estándar IEEE 802.11 (Instituto de Ingenieros en Electricidad y Electrónica) con diferentes variantes. Un organismo que estandariza a las redes WLAN es el IEEE, esta norma está referida por la familia 802.11, donde los estándares más usados son: 802.11b, el 802.11a y el 802.11g.

El estándar 802.11b/g (más conocida como Wi-Fi), es un esquema de red que crea conexiones inalámbricas entre un dispositivo y una red o Internet, funciona como una red Ethernet común sin concentradores o cable a una velocidad máxima de 54 Mbps. El alcance en interiores varía de los 20 a 130 metros de distancia. El rango de frecuencias asignado para este estándar es de 2.4GHz (2.4-2.485 GHz), el cual está disponible para redes WLAN en todo el mundo, además las señales dentro de este rango pueden penetrar barreras físicas tales como muros y techos con mayor facilidad que las frecuencias más altas.

WMAN: Usan tecnologías basadas en WiMax (World Wide Interoperability for Microwave Access), es decir, Interoperabilidad Mundial para Acceso con Microondas, un estándar de comunicación inalámbrica basado en la norma IEEE 802.16. WiMax es un protocolo parecido a Wi-Fi, pero con más cobertura y ancho de banda.

WWAN: En estas redes encontramos tecnologías como UMTS (Sistema Universal de Telecomunicaciones Móviles), utilizada con los teléfonos móviles de tercera generación (3G) y sucesora de la tecnología GSM (Sistema Global para las Comunicaciones Móviles) para móviles 2G, o también la tecnología digital para móviles GPRS. Según el rango de frecuencias utilizado para transmitir, el medio de transmisión pueden ser las ondas de radio, las microondas terrestres o por satélite, y los infrarrojos.

Dependiendo del medio, la red inalámbrica variará en sus características, por ejemplo, con ondas de radio. En el caso de las ondas electromagnéticas como son omnidireccionales, no requieren de antenas parabólicas. La transmisión no es sensible a las atenuaciones producidas por la lluvia ya que se opera en frecuencias no demasiado elevadas, en este rango se encuentran las bandas desde la ELF (Frecuencia Extremadamente Baja) que va de 3 a 30 Hz, hasta la banda UHF (Frecuencia Ultra Alta) que va de los 300MHz a los 3GHz.

1.1.3 Enlaces de comunicación en una Red Ethernet Inalámbrica

Existen varias clases de dispositivos hardware que se pueden utilizar para implementar una red inalámbrica Wi-Fi: los Adaptadores Inalámbricos o Controladores de Interfaz de Red **NIC** *Network Interface Control* son tarjetas de red que cumplen con el estándar 802.11 que les permiten a un equipo conectarse a una red inalámbrica (tarjetas de Interconexión de Componentes Periféricos o PCI, adaptadores USB y tarjetas Compact Flash). Una estación es cualquier dispositivo que tenga este tipo de tarjeta. Los Puntos de Acceso **AP** *Access Point*, a veces denominados zonas locales de cobertura, pueden permitirles a las estaciones equipadas con Wi-Fi cercanas acceder a una red conectada a la que el AP se conecta directamente.

El estándar 802.11 define 2 modos operativos: el modo de infraestructura en el que los clientes de tecnología inalámbrica se conectan a un AP y el modo ad-hoc en el que se conectan entre sí, sin ningún AP como intermediario.

En el modo infraestructura, cada estación **EST** se conecta a un AP. La configuración formada por el AP y las estaciones ubicadas dentro del área de cobertura se llama conjunto de servicio básico **BSS** *Basic Set Service*. Estos forman una célula. Cada BSS se identifica a través de un identificador **BSSID** *Basic Set Service IDentifier* que es la dirección de Control de Acceso al Medio **MAC** *Media Access Control* del AP. Es posible vincular varios AP juntos y varios BSS con una conexión llamada Sistema de Distribución **SD** para formar un conjunto de servicio extendido **ESS** *Extended Set Service*. El SD también puede ser una red conectada, un cable entre dos AP o incluso una red inalámbrica. Cuando un usuario va de un BSS a otro mientras se mueve dentro del ESS, el adaptador de la red inalámbrica de su equipo puede cambiarse de AP, según la calidad de la señal que reciba desde los distintos AP, éstos se comunican entre sí a través de un SD con el fin de intercambiar información sobre las estaciones y si es necesario, para transmitir datos desde estaciones móviles. Esta característica que permite a las estaciones moverse "de forma transparente" de un AP al otro se denomina “itinerancia” (*Handoff*).

La comunicación con un AP se da cuando una estación se une a una célula, esta envía una solicitud de sondeo a cada canal la cual contiene el ESSID que la célula está configurada para usar y también el volumen de tráfico que su adaptador inalámbrico puede admitir. Si no se establece ningún ESSID, la estación escucha a la red para encontrar un Identificador del Conjunto de Servicio **SSID** *Set Service IDentifier*. Cada AP transmite una señal en intervalos regulares (10 veces por segundo aproximadamente) a esta acción se le denomina “señalización” y provee información de su BSSID, sus características y su ESSID, sí ésta corresponde.

Una estación dentro del rango de muchos AP (sólo los que tengan el mismo SSID) puede elegir el punto que ofrezca la mejor proporción entre capacidad de carga de tráfico.

La información se transmite por ondas de Radio Frecuencia **RF**, fragmentada en "paquetes" de distintos tamaños, los paquetes bajo el estándar 802.11 son paquetes de administración que establecen y mantienen la comunicación; Otros son los paquetes de control que ayudan a la entrega de datos y tiene funciones de coordinación. Y por último, los paquetes de datos que contienen información de usuario. [2]

1.1.4 Enlaces de comunicación en una Red Celular

GSM *Global System communication Mobil* / **GPRS** *General Packet Radio Service* es un sistema estándar definido para la comunicación mediante teléfonos móviles que incorporan tecnología digital, cualquier cliente de una red GSM puede navegar por Internet y utilizar distintas funciones digitales de transmisión de datos, esta transmisión de datos es de 9,6 kbit/s . Es claramente baja si la comparamos con los 56 Kbit/s que permite un módem convencional de acceso a Internet a través de la línea telefónica.

GPRS es una tecnología basada en la transmisión de paquetes y coexiste con GSM, compartiendo gran parte de la infraestructura desplegada en el mismo, pero ofreciendo al usuario un servicio portador más eficiente para las comunicaciones de datos, especialmente en el caso de los servicios de acceso a redes IP como Internet. La velocidad máxima que puede alcanzar GPRS es de unos 40 kbit/s en el enlace descendente y 9,6 kbit/s en el ascendente. Las principales ventajas que aporta GPRS respecto a GSM son una mayor velocidad de transmisión y una conexión permanente, convirtiéndolo así en el portador ideal para los servicios WAP y el acceso a Internet. Durante la conexión, a cada usuario se le asigna un par de canales de frecuencia, uno para subida y otro para bajada. Un usuario GPRS sólo usará la red cuando envíe o reciba un paquete de información. Todo el tiempo que esté inactivo podrá ser utilizado por otros usuarios para enviar y recibir información. Esto permite a los operadores dotar de más de un canal de comunicación sin miedo a saturar la red, de tal forma que mientras en

GSM sólo se ocupa un canal de recepción de datos del terminal a la red y otro canal de transmisión de datos desde la red al terminal, en GPRS es posible tener terminales que gestionen cuatro canales simultáneos de recepción y dos de transmisión.

GPRS soporta el protocolo IPv4 para asignar la dirección IP, se utiliza DHCP (Protocolo de configuración de Host Dinámico) por lo que las direcciones IP de los equipos móviles son casi siempre dinámicas.

El Subsistema de Estaciones Base **BSS**[1] *Base Station Subsystem*, "terminal de usuario – Estación Base **BS** *Base Station*– Controlador de Estación Base **BSC** *Base Station Controller*", es la capa inferior de la arquitectura y resuelve el problema del acceso del terminal al canal. La siguiente capa Subsistema de Conmutación y Red **NSS** se encargará, por un lado, del enrutamiento a través del Centro de conmutación Móvil **MSC** *Mobil Switching Center* y por otro de la identificación del abonado, tarificación y control de acceso.

Una BS puede alcanzar un radio de cobertura a su alrededor desde varios cientos de metros, en estaciones urbanas hasta un máximo de 35 Km. y en zonas rurales, según su potencia y la orografía del entorno. Sin embargo, el número de usuarios que puede atender cada BS está limitado por el ancho de banda (subdividido en canales) que el BSC asigna a cada estación, y aunque podría pensarse que las estaciones base deberían tener una gran potencia para cubrir mayor área, siempre emiten al menor nivel de potencia posible para evitar interferir con celdas lejanas que pudieran emplear el mismo rango de frecuencias, motivo por el cual es raro que se instalen modelos de más de 40 W. En zonas muy pobladas se instala un mayor número de BS de potencia muy limitada (menor que 2,5 W) para permitir la creación de las llamadas micro celdas, que permiten mejorar la reutilización de frecuencias (entre más estaciones, más reutilización de frecuencias y más usuarios admisibles al mismo tiempo). Por tanto, en zonas donde exista una gran concentración de usuarios, como ciudades, debe instalarse un gran número de BS de potencia muy limitada y en zonas de menor densidad de uso, como áreas rurales, puede reducirse el número de estaciones y ampliar su potencia.

Durante el transcurso de una llamada, el terminal no se encuentra emitiendo todo el tiempo. Para ahorrar batería y permitir un uso más eficiente del espectro se emplea el esquema de transmisión TDMA (Acceso Múltiple por división de Tiempo). El tiempo se divide en unidades básicas de 4,615 ms, y éstas a su vez en 8 time slots (ranuras de tiempo) de 577 µs.

Durante una llamada, se reserva el primer time slot para sincronización, enviada por la BS, unos slots más tarde, el terminal emplea un slot para enviar de terminal a BS y otro para recibir, y el resto quedan libres para el uso de otros usuarios en la misma BS y canal. Así se permite un buen aprovechamiento del espectro disponible y una duración de batería superior, al no usar el emisor del terminal constantemente, sino sólo una fracción del tiempo.

La comunicación no debe interrumpirse porque un usuario se desplace y salga de la zona de cobertura de una BS. Tanto el terminal del usuario como la BS calibran los niveles de potencia con que envían y reciben las señales e informan de ello al BSC. Además, normalmente varias BS al mismo tiempo pueden recibir la señal de un terminal y medir su potencia, de este modo el BSC puede detectar si el usuario va a salir de una celda para entrar en otra, avisa a ambas BS y al terminal para el proceso de salto de una BS a otra. Este proceso es mejor conocido como *handover* o traspaso entre celdas. [3]

1.2 Planteamiento del problema

En la actualidad, los dispositivos móviles aportan grandes ventajas convirtiéndose en un importante medio de comunicación, entre las que destacan su portabilidad y el soporte de acceso a Internet, entre otros. Hoy día, la mayoría de dispositivos móviles tienen capacidad para enlazarse a Internet, el medio más conocido para éstos, es por vía celular, mientras que otros pocos lo pueden hacer por medio de una red Wireless Ethernet y otros menos tienen la capacidad para hacerlo por ambos medios, en esta tesis está fundamentado el presente proyecto de investigación.

El desarrollo de la infraestructura para comunicación Cliente-Servidor para aplicaciones móviles requiere de un sistema de comunicación bidireccional utilizando sistemas de comunicación inalámbricas. El usuario debe ser capaz de enlazarse a Internet por dos medios conocidos vía redes GSM, a través de GPRS o por Wireless Ethernet (Wi-Fi) mediante AP inalámbricos de una LAN o MAN. Las técnicas de conexión inalámbrica a Internet serán llevadas a cabo mediante GPRS y Ethernet inalámbrico, actualmente usadas para aplicaciones móviles. Donde el dispositivo debe tener prioridad por un AP sobre una antena de comunicación celular, esto es, el dispositivo debe ser capaz de realizar una conexión a Internet, inicialmente por medio de Wi-Fi y en el caso de que no se encuentre dentro de la zona de cobertura en un AP, conectarse a Internet por medio de la Red Celular GPRS.

Esta infraestructura estará montada en una tarjeta de desarrollo, el OMAP 3530 de Texas Instruments, el cual implementa un hardware capaz de hacer una conexión a redes inalámbricas, entre las planteadas están Ethernet inalámbrico y Redes celulares. Asimismo, se le instalará el sistema operativo Android, diseñado para dispositivos móviles, con base en este sistema operativo se desarrollará el software de aplicación para la elección de la red inalámbrica con la cual el dispositivo móvil creará la conexión eligiendo como primera opción la de menor costo.

1.3 Justificación

Actualmente, vivimos en una sociedad donde el tiempo parece ser una de las máximas prioridades en nuestro acontecer diario, los conceptos de utilidad, eficacia y producción determinan nuestro rendimiento laboral e incluso personal por lo que muchas personas tienen la necesidad de realizar actividades donde el tiempo y la distancia son determinantes. En cualquier ámbito tener conexión a Internet, puede satisfacer muchas de las nuevas necesidades de esta sociedad contemporánea, como revisar el correo electrónico, ver las condiciones de clima, canales informativos, de entretenimiento y muchas cosas más. El acceso a Internet en los últimos años ha venido en aumento debido al desarrollo tecnológico.

En un inicio el tipo más común de acceso a Internet era con una conexión Ethernet desde una computadora personal a un módem que era proporcionado por un proveedor del servicio, ahora este tipo de conexión ha sido un poco remplazada por los AP que nos proporciona el proveedor de Internet, donde ahora con tan sólo tener en tu computadora personal una tarjeta de red inalámbrica del estándar 802.11, se puede hacer una conexión inalámbrica hacia el módem proporcionado, dicho módem colocado en nuestro hogar es capaz de dar una cobertura de aproximadamente 100 metros. Otra manera de tener acceso a Internet de forma inalámbrica que ha evolucionado en los últimos años es la proporcionada por compañías de telefonía celular, con sus tecnologías de 2 ½ y 3ra generación, dichas compañías ofrecen servicio de Internet, por medio de la infraestructura de Red Celular.

Analizando las dos formas de tener una conexión a Internet, ya sea por Red Celular o por AP de una red Ethernet inalámbrica y comparando en costos, se puede concluir que una conexión Ethernet inalámbrico, es mucho más barata que una conexión ofrecida por una Red Celular, tomando en cuenta que en un futuro no muy lejano se espera contar con una conexión a

una MAN por medio de AP colocados en gran parte de la ciudad y a un bajo costo.

Con esta investigación se pretende lograr que un dispositivo móvil siempre tenga acceso a Internet, ya sea por una conexión Ethernet Inalámbrica o por medio de una Red Celular. Contemplando el costo que genera una conexión a Internet y sabiendo que una conexión por medio de un AP es más barato que por una antena de Red Celular, se diseñó un módulo de conexión a internet para que el dispositivo móvil pueda tener prioridad por un AP de una red Ethernet inalámbrica, de esta manera, personas con este tipo de dispositivo móvil que tengan la necesidad de estar siempre conectadas a Internet, podrían reducir costos a través de una conexión a Internet mediante un AP, con la opción de poder hacerlo mediante una Red Celular, pero siempre se intentará realizar la conexión a Internet de la forma más económica, considerando que el dispositivo siempre tendrá disponible alguna conexión a Internet a través de estas dos opciones.

1.4 Objetivos

Los objetivos que se persiguen con el desarrollo de este proyecto se mencionan a continuación:

1.4.1 Objetivo General

Integrar y/o en su caso desarrollar hardware y software de infraestructura para comunicaciones Cliente-Servidor utilizando sistemas de comunicación inalámbrica GSM/GPRS/Ethernet para aplicaciones móviles embebidas.

1.4.2 Objetivos Específicos:

Software

Estudiar las redes WLAN y GSM, así como los protocolos para comunicación inalámbrica Ethernet y GPRS. Estudiar los diferentes tipos de acceso al canal su configuración para el sensado y conexión a puntos de acceso. Estudiar los sistemas de asignación de IP dinámicas DHCP y Dyn-DNA, así como estudiar la configuración para conexiones de redes Ad-hoc. Programación para comunicación cliente-servidor mediante programación con comandos AT.

Hardware

Estudiar y analizar la integración de los dispositivos módem GSM/GPRS, y de interfaz de red inalámbrica Wi-Fi 802.11g en sistemas embebidos para comunicaciones móviles.

1.5 Organización del trabajo

La organización del presente trabajo se divide en 6 capítulos, a continuación se comenta el contenido de los mismos:

Capitulo 1. Introducción. En este capítulo se da una introducción al tema de investigación, se define el planteamiento del problema, la justificación, el objetivo general, los objetivos específicos y la organización del trabajo para lograr el desarrollo de la tesis.

Capítulo 2. Estado del Arte. En este capítulo se ofrece una descripción general de algunos de los aspectos técnicos de este proyecto como las Arquitecturas de dispositivos Móviles actuales, los sistemas operativos más utilizados para éstos y las características de los tipos de Dispositivos Móviles existentes hoy en día.

Capítulo 3. Marco Teórico. En este capítulo se explican las características de las redes inalámbricas Wi-Fi y Redes Celulares así como la forma en que un dispositivo móvil accede a éstas. Se explica al lector las características básicas del nuevo sistema operativo Android, su diseño, arquitectura y funcionamiento, así mismo se presenta una guía de instalación para el entorno de desarrollo con Eclipse.

Capítulo 4. Diseño del Sistema. Dentro de este capítulo se muestran los diagramas de bloques de los esquemas propuestos: software para hacer más eficientes las comunicaciones y una arquitectura para dispositivo móvil con conectividad Wi-Fi y GPRS y la descripción de cada uno de ellos.

Capítulo 5. Pruebas y Resultados. Describe de forma detallada el completo desarrollo de una aplicación para Android: desde su diseño y funcionalidad, hasta la implementación de cada una de sus clases y componentes básicos, El desarrollo de la arquitectura de dispositivo móvil su diagrama esquemático y su PCB, La instalación y compilación del sistema operativo Android para la plataforma OMAP, TAO y ordenadores x86.

Capítulo 6. Se exponen los resultados obtenidos tras la finalización del proyecto, comparándolos con los objetivos marcados inicialmente, y trazando posibles líneas futuras de desarrollo.

Por último, se incluyen una serie de anexos donde se puede encontrar el diseño esquemático de la arquitectura de un dispositivo móvil.

CAPÍTULO 2

ESTADO DEL ARTE

En este capítulo se presenta una descripción de las arquitecturas actuales de los dispositivos móviles, en donde se muestran los principales procesadores móviles usados, tales como el OMAP de Texas Instruments, SnapDragon de Qualcomm, Atom de Intel, Marvell con armada, entre otros. Se muestra una clasificación de los dispositivos móviles pasando por los PDA, teléfonos móviles y Smartphones. También se hará una descripción y comparación de las características de los actuales sistemas operativos usados en dispositivos móviles, entre los que sobresalen: PALM OS, Symbian, Windows Mobile, Iphone OS, Android, entre otros.

2.1 Arquitecturas de dispositivos Móviles

Los fabricantes de computadoras, teléfonos móviles, PDA y otros dispositivos, tales como consolas de juegos de video, están experimentando un incremento en demanda para conectividad de información en sus productos. Por ejemplo, casi el 90 % de computadoras laptop ahora están construidas con una interfaz WLAN (tal como 802.11 Wi-Fi).

Sin embargo, WLAN ofrece conectividad de información sólo cuando el usuario esta estacionariamente en una área de cobertura Wi-Fi. A su vez WWAN, es ofrecida sólo por servicios de información celular y éstos pueden a la precisión administrar la conectividad de usuarios en movimiento de manera local (quizás a la velocidad de un carro) y *roaming* a nivel nacional e internacional. [4]

Los dispositivos móviles están cada vez más equipados con sistemas operativos que aunque más ligeros se comparan en complejidad con aquellos de computadoras de escritorios. Los dispositivos móviles tienen regularmente instalados un número de aplicaciones, incluyendo clientes para servicios de localización y exploradores Web de propósito general. Estas aplicaciones usan características de hardware, tales como GPS (Sistema de Posicionamiento Global) y accesos a red vía 3G o Wi-Fi. Para soportar el aumento en complejidad de software y hardware sobre dispositivos móviles, los sistemas operativos para estos dispositivos han evolucionado. Los dispositivos móviles, comúnmente corren complejos sistemas operativos, tales como Linux, Windows Mobile, Windows Phone, Android y Symbian OS. [5]

Los procesadores móviles tienen el rendimiento apropiado para soportar sistemas operativos, decodificar audio y video, mostrar páginas Web y hacer cálculos de navegación; procesadores de tipo multimedia y de aplicación son diseñados para el equipamiento de dispositivos móviles, entre las compañías productoras están Texas Instruments (con OMAP), Marvell (con Armada y Tavor), Qualcomm (con MSM y SnapDragon), Broadcom, Freescale (con I.MX), Intel (con Atom), NetLogic (con Alchemy), Nvidia (con Tegra), Renesas (con SH- Mobile y SH-Navi), STMicroelectronics (con Cartesio), Anyka, Core Logic, MtekVision, y Telechips.

Los dispositivos móviles suministran el acceso a Internet a cualquier persona y a cualquier hora, por ejemplo, los Smartphones suministran voz y conectividad a Internet. *Personal Media Players* **PMP** nos permiten llevar en un bolsillo canciones y películas. *Electronic-book readers* (*e-readers*) proveen una gran pantalla para leer libros digitales. Los *Personal Navegation*

Devices **PND** nos muestran el camino más rápido a nuestro destino. Todos estos dispositivos requieren de un procesador avanzado que soporte una interfaz de usuario sofisticada y varias aplicaciones, ya que muchos de éstos deben decodificar audio y formatos de video, mostrando imágenes de alta resolución. [6]

Los procesadores embebidos han ido evolucionando rápidamente. Para hacer posible la rapidez del explorador Web, algunos de los más nuevos procesadores suministran uno o dos CPU operando casi a 1Ghz, muchos pueden mostrar y grabar videos de alta definición y para soportar interfaces de usuario sofisticadas, incluyen un acelerador de gráficos 3D.

El incremento de las redes inalámbricas ha creado al mismo tiempo un aumento en la demanda de dispositivos portátiles que integran diferentes servicios tales como procesamiento de texto, acceso a bases de datos remotas y procesamiento multimedia. Las exigencias de la sociedad de la información para acceder desde un ambiente móvil a servicios computacionales han motivado el desarrollo de sistemas convergentes que soporten de forma integrada diversas aplicaciones como multimedia y conectividad a redes inalámbricas. Para responder a estos requerimientos, distintas arquitecturas para sistemas embebidos han sido propuestas en los últimos años, todas ellas orientadas a ofrecer un alto desempeño en el procesamiento de datos multimedia en tiempo real, manejo eficiente de la energía, tamaño pequeño y baja complejidad en el diseño.

Actualmente, el diseño de arquitecturas orientadas a aplicaciones móviles se caracteriza por integrar en un sólo chip diferentes módulos de hardware (System-on-Chip, SoC), esto implica nuevos retos para los diseñadores, ya que es crítico no sólo el acoplamiento de módulos de procesamiento multimedia, sino también el diseño de plataformas de comunicación *on-chip* óptimas, que permitan el manejo eficiente de un gran flujo de datos. [7]

Los procesadores para dispositivos móviles se han clasificado en tres tipos: los procesadores multimedia que actúan como coprocesadores, llevando funciones de audio y video. Los procesadores de aplicación que sirven como un CPU principal, corriendo el Sistema Operativo, éstos regularmente también controlan tareas multimedia. Y por último, los procesadores de teléfonos inteligentes integrados que combinan el procesador de aplicación y un *"cellular baseband"* en un solo chip.

Texas Instruments ha dominado en procesadores de aplicación. Su procesador OMAP aparece en teléfonos como, el Palm Pre, Motorola Droid, y en muchos de los teléfonos inteligentes de Nokia. Marvell recientemente

actualizó su línea de procesadores de aplicación con su nueva familia denominada Armada. Esta familia incluye procesadores diseñados para teléfonos inteligentes y Netbooks. El Armada 166E es el primer procesador de la industria diseñado para soportar un controlador de pantalla de cristal liquido para e-readers. Marvell también ofrece un procesador de teléfono inteligente integrado conocido como Tavor, el cual aparece en muchas blackberry. La familia MSM 7200 de Qualcomm es la líder en procesadores de teléfonos inteligentes integrados, tomando ventaja en liderazgo en chips de Acceso Múltiple por División de Código **CDMA** y UMTS *baseband*. Sus procesadores generalmente ofrecen las soluciones más integradas, baja potencia y corto tiempo de diseño. Los procesadores SnapDragon son los productos basados en ARM *Advanced RISC Machines* de más alto rendimiento para Dispositivos de Internet Móviles **MID** y *Netbooks*.

La segunda generación ATOM de Intel es apta para netbooks, pero aún es demasiado voluminoso y gran consumidor de energía para teléfonos inteligentes. Por su parte, Freescale ha estado ganando terreno con sus procesadores I.MX, usado en los e-readers de Sony. Y el BCM4760 de Broadcom es el primer procesador de aplicación para integrar un completo subsistema de GPS, creando un PND-on-chip. [6]

El hardware en los dispositivos móviles es aún muy limitado en términos de poder de cálculo, memoria y reservas de energía. Las capacidades de hardware de estos dispositivos son similares a aquellas PC de escritorio de 1990, como se muestra en la tabla 2.1. [8]

Teléfono	**CPU (MHz)**	**RAM (MB)**	**Batería (tiempo de habla en horas)**
Iphone 3G	412	512	5
Android HTC G1	528	192	6
Blackberry Bold	624	128	4.5

Tabla 2.1 Especificaciones de teléfonos inteligentes, su conectividad a redes incluye: Wi-Fi, UMTS, WCDMA, HSDPA, GSM/GPRS/EDGE y Bluetooth.

El rango de arquitecturas propuestas para aplicaciones móviles va desde procesadores de aplicación específica, de propósito general **GPP** con extensiones multimedia, conocidos también como GPP de instrucciones con múltiples datos **SIMD**, procesadores multimedia de propósito especial, procesadores superescalares y de tipo **VLIW** *Very Long Instruction Word*. Así como plataformas completamente re-configurables.

Sin embargo, el rango de arquitecturas que han demostrado ser óptimas para responder acertadamente a las restricciones impuestas por las aplicaciones móviles es más reducido que la clasificación presentada anteriormente. En el caso de las arquitecturas superescalares y VLIW tradicionales, trabajan a altas frecuencias de reloj y usan complejas estructuras para la implementación de paralelismo, lo cual se traduce en un alto consumo de energía y en un incremento de los costos de la arquitectura. Para arquitecturas tradicionales SIMD, el alto costo que involucra el proceso de reorganización de datos necesario en algunos casos puede introducir grandes restricciones haciéndolos poco viables.

Los procesadores multimedia de propósito especial, tales como el STN8800 de STMicroelectronics, el S3C24A0 de Samsung, el Dragonball MC9328MX21 de Freescale y el PXA27x de Intel, tienen módulos de hardware en forma de periféricos para realizar funciones de procesamiento, lo que permite obtener un mejor desempeño. Existen por tanto, un número creciente de procesadores y arquitecturas multimedia de propósito especial diseñados con fines académicos y comerciales. En [7] se presenta un análisis de las características relevantes de algunas arquitecturas para cómputo móvil.

2.1.1 Arquitectura digital móvil, Mobydick

Arquitectura formada por un procesador de propósito general y módulos re-configurables autónomos (módulos de red, pantallas LCD, audio y video), todos ellos interconectados por un switch denominado *Octopus*. Los módulos son propuestos para integrar modos de conservación de energía. Esta arquitectura presenta un diseño centrado en la conexión por el switch *Octopus*, el cual reduce el tráfico de datos mediante la eliminación de copias innecesarias al remover la CPU del flujo de datos, mas no del flujo de control. Así, la CPU es reducida a operar como un controlador que inicializa el sistema, maneja la calidad de servicio QoS (*Quality of Service*) asociado con cada conexión y maneja la ejecución de algunos protocolos complejos.

El módulo de red incluye un módem WaveLAN, que proporciona la funcionalidad básica para enviar y recibir *frames* de datos en modo half-*duplex* 2 Mb/s; una memoria SRAM de 512 Kbytes que actúa como pila de conexión; un PGA Xilinx XC4010 que controla el flujo de datos entre el módem y el Host y proporciona funciones de corrección de errores; un microcontrolador PIC 16C66 el cual implementa el control de tráfico, el control del enlace de datos, y el protocolo MAC eficiente en energía E2MaC, que permite la conexión del sistema a una estación base a través de una red

inalámbrica ATM con un QoS apropiado. El módulo de *LCD* es de tamaño pequeño y baja resolución e incluye hardware con decodificación y post-procesamiento. [9]

2.1.2 Procesador Nomadik STN8800 de STMicroelectronics

Los procesadores de STMicroelectronics disponen de una arquitectura abierta de procesamiento distribuido, que usa aceleradores programables inteligentes para la realización en forma paralela de funciones intensivas en datos y un procesador de propósito general para el control. Como módulo CPU, el STN8800, utiliza el ARM926E-JS de 32 bits de la familia ARM9 que incluye MMU, caché de instrucciones de 32KB, caché de datos de 16KB, multiplicador de 16 x 32 bits que realiza operaciones MAC en un ciclo y un acelerador Java. Además la plataforma ARM integra una arquitectura de comunicación multicapas basada en un bus conocido como **AMBA** *Advanced Microcontroller Bus Architecture* con un bus tipo **AHB** *Advanced High-performance Bus*. Este subsistema consiste en un esquema de buses multicapas síncrono que permite realizar transferencias de datos entre módulos del sistema sin la intervención de la CPU en el flujo de datos. El procesador STN8800 posee dos módulos aceleradores, uno para audio y otro para vídeo. Éstos están basados en el core MMDSP, el cual es un Procesador de Señales Digitales **DSP** comercial de 75MHz tipo VLIW programable. El acelerador de audio soporta los estándares MPEG3, y MIDI. El acelerador de vídeo es un híbrido hardware/software que incluyen tanto al núcleo MMDSP como módulos de hardware dedicados que desempeñan funciones de pre-procesamiento, post-procesamiento, codificación y decodificación.

Estos procesadores incorporan técnicas de bajo consumo de energía, entre ellas se destacan: el uso de memoria on-chip como buffer de caché; el uso de modos de operación de bajo consumo de la CPU; la implementación de la técnica de clock-gating y la reducción del voltaje de alimentación. [10]

2.1.3 Procesador PXA270 de Intel

Los procesadores PXA, poseen características que les permiten soportar las exigencias del procesamiento multimedia en ambientes móviles. La familia PXA27x, integra un core XScale de Intel como módulo CPU, una memoria SRAM on-chip de 256KB la cual es usada como *frame buffer* para aplicaciones gráficas y de vídeo, diversos módulos multimedia (e.g. controlador de LCD e interfaz de cámara) y diferentes interfaces de periféricos (e.g. UART, USB). La arquitectura ha sido mejorada por un conjunto de nuevas tecnologías entre las cuales se incluyen: 1) la tecnología Wireless MMXTM, que proporciona funcionalidades para aplicaciones como

juegos 3D, mientras optimiza el consumo de energía; 2) la tecnología Wireless SpeedStep, que implementa técnicas de reducción de consumo de potencia; 3) la tecnología Wireless Trusted, la cual proporciona servicios de seguridad en el manejo de la información. Estos procesadores están en capacidad de operar en un rango de frecuencias desde 312MHz hasta 624MHz y con una capacidad de memoria hasta de 64 MB. El core XScale de Intel es una arquitectura tipo RISC de 32 bits basada en el core ARM V5TE con un subsistema de memoria tipo Harvard formado por cachés de instrucciones y de datos de 32KB con MMU independientes y una caché de datos de 2KB que permite almacenar datos streaming. Xscale posee una unidad de manejo de potencia, unidades JTAG, interfaz de alta velocidad para la tecnología Wireless MMXTM, y un coprocesador que permite procesar algoritmos de audio tipo multimedia **MAC** *Multimedia Algorihtms Coprocessor*. La micro arquitectura tiene la capacidad de procesar datos tipo entero de 32 bits en forma tradicional y de igual forma de 16 bits en modo SIMD. Las mejoras en las capacidades de procesamiento de la familia PXA27x se deben a la integración de la tecnología Wireless MMXTM. [7]

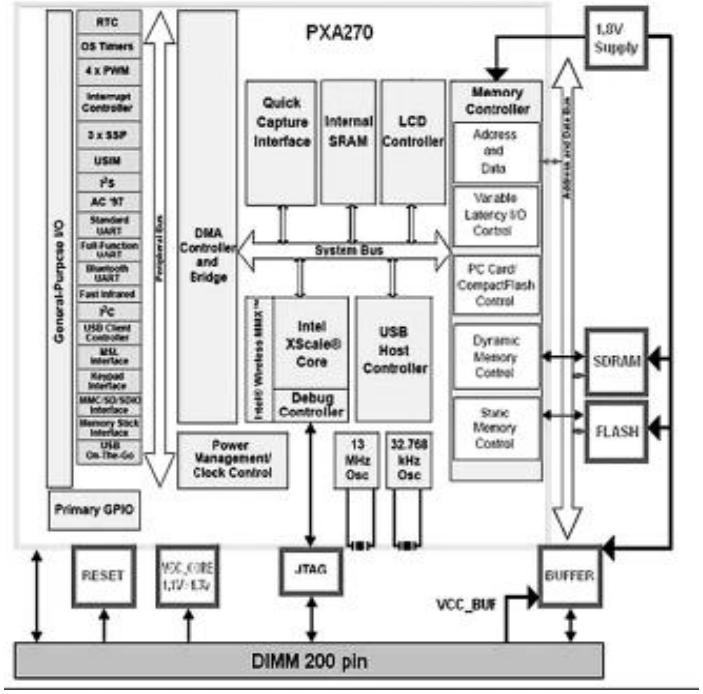


Figura 2.1 Procesador PXA270 de Intel.

2.1.4 Procesador Dragonball MC9328MX21 de Freescale

Este procesador posee como CPU el core ARM926EJ-S de 32bits de la familia ARM9, integrado en un SoC con módulos de hardware de

aplicación específica que operan en forma independiente. La CPU ARM926EJ-S es un subsistema de alto desempeño, cuya velocidad de procesamiento puede llegar hasta 266 MHz con una frecuencia máxima de reloj para el bus interno de 133 MHz, posee una arquitectura caché tipo Harvard y está formada por cuatro módulos principales: un núcleo entero ARM9EJ-S, cachés independientes de datos e instrucciones de 16KB y Unidad de Manejo de Memoria (MMU). La plataforma ARM integra el estándar AMBA-AHB, el cual permite realizar transferencias de datos entre módulos del sistema sin la intervención de la CPU en el flujo de datos. Entre los módulos específicos que posee este procesador están la Interfaz de Sensor CMOS (CSI, permite que el procesador I.MX21 se pueda conectar directamente a un sensor de imágenes CMOS externo), el Acelerador Multimedia Mejorado (eMMA) y el Controlador de Display de Cristal Líquido (LCDC, permite al procesador mostrar datos en paneles LCD).

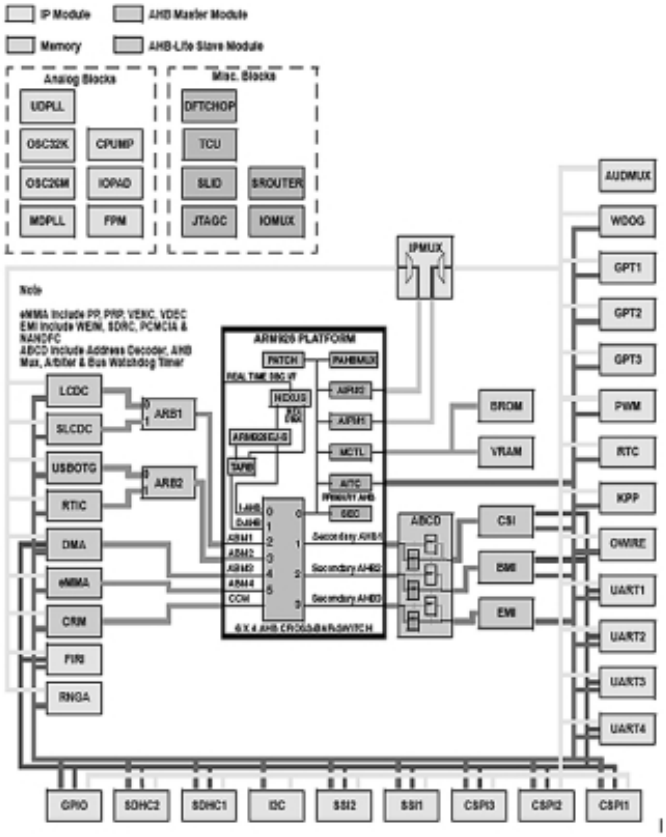


Figura 2.2 Procesador Dragonball MC9328MX21 de Freescale.

Algo importante es el uso de AMBA-AHB en el esquema de comunicación, el cual es conocido para esta tecnología como Multi-Layer AHB Crossbar Switch (MAX). Este esquema permite realizar transferencias de datos en forma simultánea y paralela entre diferentes módulos del

sistema. El estándar soporta múltiples maestros y ofrece un alto desempeño y gran ancho de banda. Los módulos manejados como maestros son: Icache y Dcache que proporcionan rutas para acceder la memoria externa, el módulo DMA y periféricos de aplicación como el CLCD, la interfaz USB y el módulo eMMA. Para reducir el consumo de potencia, el procesador MC9328MX21 soporta tres modos de operación (*run, standby, stop*). [7]

Una evaluación cuantitativa en desempeño y consumo de potencia de algunos de los procesadores orientados a aplicaciones móviles y multimedia es aportada por Synchromesh Computing, [11] el cual compara los procesadores i.MX21 de Freescale, PXA255 de Intel y el OMAP 1611 de TI. Para la realización de las pruebas se dispusieron ocho benchmarks (Hint, Stream, MPEG4 Encode, MPEG4 Decode, Sisoft Sandra, BMQ, Vobench, ByteMark) con el fin de medir diferentes parámetros en cada procesador:

	Nomadik STN8800 (sTMicroelectronics)	**PXA270 (Intel)**	**Dragonball MC9328MX21 (Freescale)**
Freq Procesador /Freq Bus	200 MHz	312MHz/104MHz 624MHz/208	MHz 266 MHz /133 MHz
Cache de Instrucciones	32KB	32KB	16K
Cache de Datos	16KB	32KB Y 2KB adicionales	16K
Memoria Interna RAM	Memoria SRAM on-chip de 48KB para almacenar los *frames* de video	SRAM de 256KB en bancos de 64KB	---
Potencia Modo Idle (mW)	100mW [22]	109 mW-260 mW (respectivamente a las frecuencias)	108mW
Potencia Modo Activo Solo manejo de UART (mW)	---	375 mW-925 mW (respectivamente a las frecuencias)	248mW
IC process	CMOS 130 nm	CMOS 130 nm	CMOS 130 nm

Tabla 2.2 Medidas comparativas entre procesadores multimedia.

Éstas fueron las conclusiones:

- OMAP1611 presenta el menor consumo de potencia (debido a la baja frecuencia del core 180MHz), pero presenta el peor desempeño

para las aplicaciones MPEG4 debido al tamaño pequeño de la caché de datos (8KB) y a la baja velocidad del bus de memoria.

- PXA255 presenta el más alto consumo de potencia con un desempeño relativo alto.
- i.MX21 a pesar de su baja velocidad de reloj respecto al PXA255 obtiene un buen desempeño con un nivel de consumo medio. De acuerdo con los autores, este procesador presenta la mejor relación desempeño/milivatios.

En la Tabla 2.2, se nota, en relación con el sistema de memoria, que todos los procesadores poseen una arquitectura tipo Harvard, con tamaños de caché entre 16KB y 32KB. Esta estructura de almacenamiento responde a las recomendaciones presentadas en [12] que establece para aplicaciones multimedia, cachés de datos con tamaños entre 16KB y 32KB debido a que ofrecen una baja tasa de error en los accesos a memoria comparada con otras aplicaciones de uso común. Igualmente para la caché de instrucciones se recomienda un tamaño de 32KB o mayor para reducir la tasa errores de acceso originadas por aplicaciones multimedia más complejas.

Por otra parte los sistemas de memoria de estos procesadores cuentan con técnicas orientadas a mejorar el desempeño y el ancho de banda tanto de las memorias caché como de la memoria principal, entre ellas se pueden citar el uso de técnicas de actualización como *write-buffer*, *write-through* y *write-back*, entre otras.

Respecto a la organización arquitectural de las plataformas, se observa que como resultado de las mejoras en los procesos tecnológicos, cada vez se integran mas componentes en un sólo chip y la comunicación entre éstos suele ser más crítica. Frente a esta situación la tendencia en el diseño de arquitecturas de comunicación está cambiando del tradicional sistema de bus compartido a estructuras de comunicación más complejas, basadas en múltiples buses que incrementan el ancho de banda [13]. Los procesadores STN8800 e IMX21 implementan el estándar AMBA, mientras que el procesador PXA270 posee una configuración tradicional en el sentido que tiene un bus central de comunicación que conecta módulos de hardware de aplicación específica a la CPU. Además, cada vez más se evidencia que la arquitectura de comunicación elegida no sólo afecta el desempeño, sino también el consumo de potencia.

Se puede concluir que en las arquitecturas embebidas orientadas a aplicaciones multimedia móviles presentadas anteriormente, el sistema de memoria y la estructura de comunicación tienen un impacto considerable en el desempeño, el costo y el consumo de potencia del sistema. La optimización de estos tres parámetros simultáneamente en el proceso de

diseño, es incompatible, ya que típicamente la mejora de uno de ellos implica el detrimento del otro. La clave para el diseño de estas arquitecturas es encontrar un punto óptimo para la relación costo/potencia considerando un desempeño suficiente para llevar a cabo la aplicación deseada de forma conveniente. [7]

2.1.5 Otras Arquitecturas

Otra arquitectura es la de un Smartphone basado en Linux, donde se diseña y se hace una implementación de hardware y software de un dispositivo móvil. El diseño de hardware contiene la mayor parte de las características para ser usados en servicios inalámbricos y la arquitectura de software que es basada en Linux. La figura 2.3 muestra la estructura de hardware del dispositivo móvil que consta de dos procesadores, un procesador de aplicación (PXA225 de Intel, alcanzando de 98 a 400 MHz) y un coprocesador (MQ1188 de nVidia) para aceleración de aplicaciones multimedia, las razones de su uso son para reducir el tráfico en el sistema de bus separando la información de tráfico entre el LCD y el *frame buffer* y para utilizar la característica de aceleración de hardware del coprocesador multimedia que ayuda a mejorar el rendimiento de aplicaciones multimedia. Esta forma de organización de hardware es constantemente utilizada en dispositivos móviles cuyo rendimiento y aplicaciones diversas son características claves. Asimismo, se puede observar que el módulo CDMA es conectado al procesador principal a través de una interfaz UART. El módulo CDMA ejecuta operaciones módem controlados por comandos AT del procesador principal. [14]

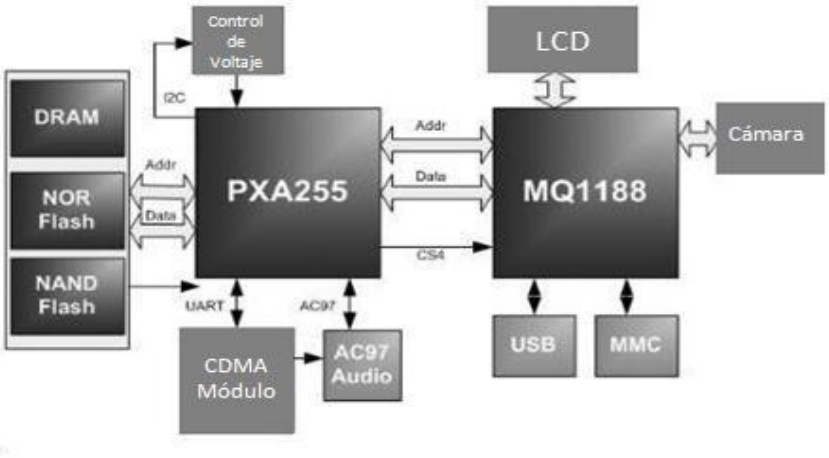


Figura 2.3 La estructura de hardware del teléfono inteligente.

Entre otras arquitecturas sobresalientes en dispositivos móviles se encuentra la del Nokia N900, con conexión a Internet vía Wi-Fi y redes 3G a través de una tarjeta SIM. Gracias a la potencia del Sistema Operativo Maemo 5 de Nokia, el terminal Nokia N900 permite a los usuarios tener docenas de aplicaciones ejecutándose simultáneamente, mientras que al mismo tiempo se puede hacer uso de las funciones de un móvil, con la ventaja de la pantalla táctil de alta resolución WVGA (800x480 píxeles) y el teclado QWERTY completo. Viene equipado con un procesador ARM Cortex-A8 con capacidad de 1GB de memoria RAM. {1}

Otra Arquitectura en dispositivos móviles actuales es el Vodafone 360 H1 de Samsung. Éste viene con una pantalla táctil y de alta resolución OLED ("*Organic Light Emitting Diode*") de 3.5 pulgadas, memoria de 16 GB, Wi-Fi y una batería con tiempo de conversación de 400 minutos, todo basado en un sistema operativo LiMo (Linux Mobile). {2}

Existe también la arquitectura HTC HD2 viene equipado con la nueva versión del sistema operativo para móviles de Microsoft, Windows Mobile 6.5, Es el primer teléfono Windows con una pantalla TFT táctil capacitiva. Viene equipado por el procesador SnapDragon de Qualcomm. Posee conectividad de banda ancha móvil HSDPA 7,2 Mbps, HSUPA 2 Mbps; Wi-Fi 802,11 b/g y receptor GPS integrado con función A-GPS. {3}

El Sistema LG-GW620 cuenta con una pantalla táctil de 3 pulgadas y un teclado QWERTY deslizable, cuenta con el sistema operativo creado por Google, el Android OS v1.5 (Cupcake). [15]

El Sistema DEXT MB220, es el primer móvil de Motorola con el sistema operativo Android. El DEXT incorpora un teclado QWERTY completo, una pantalla TFT capacitiva, 1GB de memoria interna de almacenamiento, conectividad de banda ancha móvil HSDPA, Wi-Fi y Bluetoothy y receptor GPS integrado. {4}

El Sistema Xperia X10 de Sony Ericsson es también basado en Android, viene equipado con una pantalla táctil de cuatro pulgadas e incorpora el chip SnapDragon de Qualcomm, productos y servicios de comunicaciones inalámbricas digitales basados en la tecnología digital CDMA. Dicho chip ya es usado por el TG1 de Toshiba, o el HD2 de HTC Corporation, fabricante de dispositivos portátiles basados en Windows Mobile de Microsoft y Google Android. {5}

2.2 Dispositivo Móvil

Un dispositivo móvil se puede definir como un aparato de dimensiones pequeñas que cuenta con algunas capacidades de procesamiento, conexión permanente o intermitente a una red, memoria suficiente y que ha sido diseñado específicamente para una o pocas funciones, pero que al mismo tiempo puede llevar a cabo otras funciones más generales. De acuerdo con esta definición existe una gran variedad de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores GPS, pasando por los Tablet PC, teléfonos móviles y los PDA (Asistente Personal Digital).

2.2.1 Tipos de Dispositivos Móviles

Ya que el número de niveles de funcionalidad asociado con dispositivos móviles es variado, fue necesario clasificarlos, por esta razón en el año 2005, T38 y DuPont Global Mobility Innovation Team propusieron los siguientes estándares para la definición de dispositivos móviles.

- Dispositivo Móvil de Datos Limitados: Teléfonos móviles clásicos. Se caracterizan por tener una pantalla pequeña de tipo texto y ofrecen servicios de datos limitados a SMS y acceso WAP.
- Dispositivo Móvil de Datos Básicos: Se caracterizan por tener una pantalla de tamaño medio, menú o navegación basada en íconos, ofrece acceso a e-mails, lista de direcciones, SMS y en algunos casos a un navegador Web básico. Un ejemplo de este tipo de dispositivos son los Smartphones.
- Dispositivo Móvil de Datos Mejorados. Se caracterizan por tener pantallas de medianas a grandes (por encima de los 240 x 120 píxeles), navegación de tipo stylus y que ofrecen las mismas características que el "Dispositivo Móvil de Datos Básicos". Además de aplicaciones nativas de Microsoft Office Mobile como Word, Excel, PowerPoint y aplicaciones en versión móvil como SAP, portales intranet, entre otros. Este tipo de dispositivos incluyen Sistemas Operativos como Windows Mobile.

2.2.1.1 PDA

Un PDA es una computadora de mano originalmente diseñada como agenda electrónica con un sistema de reconocimiento de escritura. Actualmente, estos dispositivos pueden realizar muchas de las funciones de una computadora de escritorio, pero con la ventaja de ser portátil. Inicialmente los PDA incluían aplicaciones estrictamente relacionadas con su función, es decir, se reducían a agenda electrónica, calendario, lista de contactos, bloc de notas y recordatorios. Hoy día, ofertan aplicaciones que incluyen juegos, acceso al correo electrónico, reproducción de películas,

crear documentos y navegar por Internet. Cuentan con pantalla sensible al tacto, ranura para tarjeta de memoria, Bluetooth y Wi-Fi.

2.2.1.2 Teléfono Móvil

El teléfono móvil es un dispositivo inalámbrico electrónico basado en la tecnología de ondas de radio que tiene la misma funcionalidad que cualquier teléfono de línea fija. Aunque su principal función es la comunicación de voz, su rápido desarrollo ha incorporado funciones adicionales como mensajería instantánea, agenda, juegos, cámara fotográfica, agenda, acceso a Internet, reproducción de video, mp3 e incluso GPS.

2.2.1.3 Smartphone

Dispositivo que funciona como un teléfono móvil con características similares a las de un ordenador personal, oscila entre el tipo de un teléfono móvil clásico y una PDA ya que permite hacer llamadas y enviar mensajes de texto como un móvil convencional, pero además incluye características cercanas a las de un ordenador personal. Casi todos los teléfonos inteligentes permiten la instalación de programas para incrementar el procesamiento de datos y conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero. Los teléfonos inteligentes cuentan con pantallas táctiles, un sistema operativo, conectividad a Internet y el acceso al correo electrónico. Otras aplicaciones que tienen presentes son las cámaras integradas, la administración de contactos, el software multimedia para reproducción de música, visualización de fotos, video-clips, programas de navegación y en algunos casos la capacidad de manejar documentos en formatos como PDF y Microsoft Office. Asimismo, la mayoría de los teléfonos inteligentes también permiten al usuario instalar programas adicionales. [16]

2.3 Sistemas operativos para dispositivos Móviles

Partiendo de la definición de sistema operativo, capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos a nivel de máquina necesarios para llevarlas a cabo. Se deduce por tanto, que el uso de uno u otro determinarán las capacidades de los dispositivos y su forma de interactuar con el usuario. Existe una multitud de opciones, las más extendidas son Palm OS, Symbian, BlackBerry OS,

Windows Mobile, iPhone OS, Android de Google y los dispositivos con sistema operativo de di

2.3.1 PALM OS

Palm OS es un sistema operativo propietario destinado a dispositivos móviles, específicamente a PDA. En 1996 Palm OS comenzó su desarrollo, un año después Palm Inc. comenzó a licenciarlo con novedosos aparatos como la PalmPilot. Las primeras versiones de este sistema operativo estuvieron basadas en un S.O. multitareas creado por Motorola.

2.3.2 Symbian

Este es el sistema operativo para móviles más extendido entre “smartphones” y por tanto el que más aplicaciones desarrolladas tiene para su sistema. Symbian ocupa más del 65 % del mercado. Nació como resultado de la alianza entre varias empresas multinacionales y de gran renombre en el mercado, tales como Nokia, Sony Ericsson, Samsung, Motorola, entre otras. Sus orígenes provienen del EPOC32, otro sistema operativo para dispositivos móviles, originado a principios de los 90s. Después de algunos años pasaría a llamarse Symbian OS. Posee un núcleo de tiempo real, cuenta con un microkernel y capacidad multithreading, soporta las arquitecturas de los últimos CPU e incluso soporta hardware "single-chip", tiene un sistema de archivos de alta performance que soporta las últimas memorias NOR, NAND, SD (Secure Digital) y MMC (Multimedia Card). Symbian OS supone una plataforma abierta, ésta es la clave que une a las telecomunicaciones y a los estándares globales de Internet que de igual forma lo hace compatible con los estándares de conectividad y redes como Bluetooth, Wi-Fi, GSM/GPRS y WCDMA.

2.3.3 Windows Mobile

Windows Mobile es un sistema operativo de la familia Windows CE, desarrollado por Microsoft. A pesar de llevar el nombre Windows, no es un sistema derivado ni es una versión recortada del mismo, sino un nuevo sistema diseñado específicamente para dispositivos móviles. Este sistema operativo está estrechamente vinculado a otros productos de la misma marca (Office Mobile, Internet Explorer Mobile, etc.) y cuenta con una interfaz gráfica similar a la de los sistemas operativos Windows. Lo anterior, ayuda a disminuir la curva de aprendizaje de los usuarios pues proveen un entorno de trabajo muy similar al que se tiene en el hogar o en la oficina. Asimismo, cuenta con procesadores de arquitecturas x86, ARM, SH4 y MIPS. Una de las ventajas de Windows Mobile es que los programadores pueden

desarrollar aplicaciones para móviles utilizando los mismos lenguajes y entornos que emplean con Windows para PC.

2.3.4 Iphone OS

La historia del IPhone OS comienza con el nacimiento del IPhone en el año 2007. Sin embargo, se ha discutido que el sistema operativo que corre en el Iphone es en realidad una versión adaptada del OS X, por lo cual, en este sentido, este sistema ya tiene años en el mercado y ha sido puesto a prueba. El sistema ha sido adaptado, removiendo los componentes que no son críticos para un dispositivo móvil y se le adicionan funcionalidades que sí están relacionadas con el mundo de la telefonía móvil y tiene una gran inclinación en el desarrollo del sistema operativo a la interfaz del usuario. El iPhone es el sistema operativo para dispositivos móviles que brinda una mejor experiencia de usuario, con un modo de manejo revolucionario basado en su Pantalla táctil *Touch Screen*, implementado mediante el framework Cocoa Touch desarrollado por Apple. Éste ya superó a Windows Mobile en la cuota de mercado que abarca. IPhone OS es una versión reducida de Mac OS X optimizada para los procesadores ARM.

2.3.5 Android

Android OS es el más reciente de los sistemas operativos para móviles del mercado. Android está siendo desarrollado por The Open Handset Alliance, un grupo de más de 30 empresas de tecnología, entre ellas la principal participante es Google. Android es un sistema operativo abierto, multitarea que permite a los desarrolladores acceder a las funcionalidades principales del dispositivo mediante una API (Interfaz de Programación de Aplicaciones). Además, el sistema operativo no diferencia entre las aplicaciones básicas del teléfono y las aplicaciones de terceros, cualquier aplicación puede ser reemplazada libremente, incluso las que trae por defecto. Además, cuenta con soporte para gráfico 2D y 3D basado en la especificación Open GL 1.0 y con conectividad Bluetooth, EDGE (Enhanced Data Rates for GSM Evolution), 3G y Wi-Fi.

Una característica sobresaliente es que se basa en el kernel de Linux versión 2.6 para las principales funciones como manejo de memoria, procesos, networking y modelo de driver. Android dio a conocer un SDK (Software Development Kit) para que los desarrolladores que lo deseen puedan programar aplicaciones que corran en el sistema operativo con el lenguaje de programación Java. Las aplicaciones corren sobre una máquina virtual diseñada para ser usada de forma embebida denominada Dalvik, la cual se ejecuta sobre el kernel de Linux donde cada aplicación en Android corre en su proceso con su propia instancia de la máquina virtual Dalvik.

2.3.6 BlackBerry OS

BlackBerry es un sistema operativo multitarea que está arrasando en la escena empresarial, en especial por sus servicios para correo y teclado QWERTY. BlackBerry apareció en el mercado cuando comenzaba a demandarse un sistema operativo que permitiera utilizar de una forma fácil, cómoda y rápida los servicios de correo electrónico. BlackBerry es un dispositivo que permite redactar, enviar y recibir todo tipo de mensajes de correo electrónico, de la misma forma que en el programa que se utiliza en un ordenador, incluso, es proveedor de servicios de correo electrónico a dispositivos de otras marcas. Además, es posible realizar y contestar llamadas que se emitan a través de la red de telefonía móvil, lo cual permite sustituir el teléfono celular. Asimismo, los dispositivos de este fabricante permiten la navegación por Internet y tienen la capacidad de enviar o recibir mensajes SMS.

2.4 Comparación de Sistemas Operativos para Móviles

El Kernel de un sistema operativo es el núcleo del mismo, es el software responsable de facilitar a los distintos programas acceso seguro al ordenador o en una forma más simple, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Por ello, es importante conocer qué núcleo utiliza cada uno de los sistemas operativos de ésta comparativa.

La principal diferencia entre un kernel de libre distribución y uno propietario es en que los de libre distribución como Linux cuentan con una amplia y experimentada comunidad de desarrolladores, gracias a los cuales se detectan rápidamente agujeros de seguridad, fallos, etc., y se realizan mejoras para lograr solucionar estos problemas, así como para adaptarse a los nuevos tiempos. En los sistemas cerrados o propietarios, el costo para encontrar errores y mejorarlos es más elevado, ya que deben ser los propios desarrolladores del sistema los que detecten y realicen las mejoras, por lo que deben dedicarse más recursos a la investigación.

	Android	BlackBerry OS 4.7	iPhone OS 3.0	S60 5th Edition	Palm WebOS	Windows Mobile 6.5
Kernel	Linux con máquina virtual Dalvik	Propietario	OS X	Symbian	Linux	Windows CE
Conectividad	3G, WiFi, GSM, GPRS	3G, GSM, CDMA, WiFi	3G, GSM, WiFi	3G. GSM, WiFi	3G, GSM, CDMA, WiFi	3G, GSM, CDMA, WiFi

Tabla 2.3 Comparativa entre sistemas operativos para móviles.

La “Adaptabilidad” es la capacidad o facilidad para poder adaptarlo a diferentes terminales o en diferentes máquinas. En este sentido, Android es el que mayor adaptabilidad presenta, ya que cada vez se está empleando en más dispositivos, no sólo en teléfonos móviles, sino también en netbooks y como sistema empotrado. En cambio en el resto de los sistemas operativos el nivel de adaptabilidad es menor y más complicado.

La “Conectividad” es primordial para poder sacar el máximo provecho a las funciones que ofrece cada sistema operativo por lo que es indispensable contar con acceso a Internet. En este sentido, se valora de gran manera el hecho de contar con acceso Wi-Fi a Internet, así como conectividad 3G que permitan el acceso a Internet desde cualquier lugar. [16]

2.5 Conclusiones

En este capítulo se puede concluir que las arquitecturas actuales de dispositivos móviles cuentan con una estructura de hardware que consta de dos procesadores, un procesador de aplicación y un coprocesador para la aceleración de aplicaciones multimedia. Texas Instruments ha dominado en procesadores de aplicación; su procesador OMAP aparece en teléfonos como, el Palm Pre, Motorola Droid, y en muchos de los teléfonos inteligentes de Nokia. El procesador OMAP tiene una arquitectura Dual-Core con un procesador ARM como procesador de aplicación y un DSP como coprocesador, por lo que será el procesador OMAP el usado como componente principal en la arquitectura del dispositivo móvil a desarrollar.

Con respecto a los sistemas operativos para dispositivos móviles se concluyó que Android es el sistema operativo más popular debido a que es el que mayor adaptabilidad presenta, ya que cada vez se está empleando en más dispositivos, no sólo en teléfonos móviles, sino también en netbooks. En cambio en el resto de los sistemas operativos el nivel de adaptabilidad es menor y más complicado, además de que Android cuenta con kernel de Linux de libre distribución. La conectividad en Android es primordial para contar con acceso a Internet, en este sentido, cuenta con soporte para Wi-Fi y conectividad 3G que permiten el acceso a Internet desde cualquier lugar. Por lo tanto, el sistema operativo Android será usado para correrlo en un procesador OMAP y para correr aplicaciones, en este caso una aplicación para hacer más eficiente las comunicaciones.

CAPÍTULO 2

ESTADO DEL ARTE

En este capítulo se presenta una descripción de las arquitecturas actuales de los dispositivos móviles, en donde se muestran los principales procesadores móviles usados, tales como el OMAP de Texas Instruments, SnapDragon de Qualcomm, Atom de Intel, Marvell con armada, entre otros. Se muestra una clasificación de los dispositivos móviles pasando por los PDA, teléfonos móviles y Smartphones. También se hará una descripción y comparación de las características de los actuales sistemas operativos usados en dispositivos móviles, entre los que sobresalen: PALM OS, Symbian, Windows Mobile, Iphone OS, Android, entre otros.

2.1 Arquitecturas de dispositivos Móviles

Los fabricantes de computadoras, teléfonos móviles, PDA y otros dispositivos, tales como consolas de juegos de video, están experimentando un incremento en demanda para conectividad de información en sus productos. Por ejemplo, casi el 90 % de computadoras laptop ahora están construidas con una interfaz WLAN (tal como 802.11 Wi-Fi).

Sin embargo, WLAN ofrece conectividad de información sólo cuando el usuario esta estacionariamente en una área de cobertura Wi-Fi. A su vez WWAN, es ofrecida sólo por servicios de información celular y éstos pueden a la precisión administrar la conectividad de usuarios en movimiento de manera local (quizás a la velocidad de un carro) y *roaming* a nivel nacional e internacional. [4]

Los dispositivos móviles están cada vez más equipados con sistemas operativos que aunque más ligeros se comparan en complejidad con aquellos de computadoras de escritorios. Los dispositivos móviles tienen regularmente instalados un número de aplicaciones, incluyendo clientes para servicios de localización y exploradores Web de propósito general. Estas aplicaciones usan características de hardware, tales como GPS (Sistema de Posicionamiento Global) y accesos a red vía 3G o Wi-Fi. Para soportar el aumento en complejidad de software y hardware sobre dispositivos móviles, los sistemas operativos para estos dispositivos han evolucionado. Los dispositivos móviles, comúnmente corren complejos sistemas operativos, tales como Linux, Windows Mobile, Windows Phone, Android y Symbian OS. [5]

Los procesadores móviles tienen el rendimiento apropiado para soportar sistemas operativos, decodificar audio y video, mostrar páginas Web y hacer cálculos de navegación; procesadores de tipo multimedia y de aplicación son diseñados para el equipamiento de dispositivos móviles, entre las compañías productoras están Texas Instruments (con OMAP), Marvell (con Armada y Tavor), Qualcomm (con MSM y SnapDragon), Broadcom, Freescale (con I.MX), Intel (con Atom), NetLogic (con Alchemy), Nvidia (con Tegra), Renesas (con SH- Mobile y SH-Navi), STMicroelectronics (con Cartesio), Anyka, Core Logic, MtekVision, y Telechips.

Los dispositivos móviles suministran el acceso a Internet a cualquier persona y a cualquier hora, por ejemplo, los Smartphones suministran voz y conectividad a Internet. *Personal Media Players* **PMP** nos permiten llevar en un bolsillo canciones y películas. *Electronic-book readers* (*e-readers*) proveen una gran pantalla para leer libros digitales. Los *Personal Navegation*

Devices **PND** nos muestran el camino más rápido a nuestro destino. Todos estos dispositivos requieren de un procesador avanzado que soporte una interfaz de usuario sofisticada y varias aplicaciones, ya que muchos de éstos deben decodificar audio y formatos de video, mostrando imágenes de alta resolución. [6]

Los procesadores embebidos han ido evolucionando rápidamente. Para hacer posible la rapidez del explorador Web, algunos de los más nuevos procesadores suministran uno o dos CPU operando casi a 1Ghz, muchos pueden mostrar y grabar videos de alta definición y para soportar interfaces de usuario sofisticadas, incluyen un acelerador de gráficos 3D.

El incremento de las redes inalámbricas ha creado al mismo tiempo un aumento en la demanda de dispositivos portátiles que integran diferentes servicios tales como procesamiento de texto, acceso a bases de datos remotas y procesamiento multimedia. Las exigencias de la sociedad de la información para acceder desde un ambiente móvil a servicios computacionales han motivado el desarrollo de sistemas convergentes que soporten de forma integrada diversas aplicaciones como multimedia y conectividad a redes inalámbricas. Para responder a estos requerimientos, distintas arquitecturas para sistemas embebidos han sido propuestas en los últimos años, todas ellas orientadas a ofrecer un alto desempeño en el procesamiento de datos multimedia en tiempo real, manejo eficiente de la energía, tamaño pequeño y baja complejidad en el diseño.

Actualmente, el diseño de arquitecturas orientadas a aplicaciones móviles se caracteriza por integrar en un sólo chip diferentes módulos de hardware (System-on-Chip, SoC), esto implica nuevos retos para los diseñadores, ya que es crítico no sólo el acoplamiento de módulos de procesamiento multimedia, sino también el diseño de plataformas de comunicación *on-chip* óptimas, que permitan el manejo eficiente de un gran flujo de datos. [7]

Los procesadores para dispositivos móviles se han clasificado en tres tipos: los procesadores multimedia que actúan como coprocesadores, llevando funciones de audio y video. Los procesadores de aplicación que sirven como un CPU principal, corriendo el Sistema Operativo, éstos regularmente también controlan tareas multimedia. Y por último, los procesadores de teléfonos inteligentes integrados que combinan el procesador de aplicación y un *"cellular baseband"* en un solo chip.

Texas Instruments ha dominado en procesadores de aplicación. Su procesador OMAP aparece en teléfonos como, el Palm Pre, Motorola Droid, y en muchos de los teléfonos inteligentes de Nokia. Marvell recientemente

actualizó su línea de procesadores de aplicación con su nueva familia denominada Armada. Esta familia incluye procesadores diseñados para teléfonos inteligentes y Netbooks. El Armada 166E es el primer procesador de la industria diseñado para soportar un controlador de pantalla de cristal liquido para e-readers. Marvell también ofrece un procesador de teléfono inteligente integrado conocido como Tavor, el cual aparece en muchas blackberry. La familia MSM 7200 de Qualcomm es la líder en procesadores de teléfonos inteligentes integrados, tomando ventaja en liderazgo en chips de Acceso Múltiple por División de Código **CDMA** y UMTS *baseband*. Sus procesadores generalmente ofrecen las soluciones más integradas, baja potencia y corto tiempo de diseño. Los procesadores SnapDragon son los productos basados en ARM *Advanced RISC Machines* de más alto rendimiento para Dispositivos de Internet Móviles **MID** y *Netbooks*.

La segunda generación ATOM de Intel es apta para netbooks, pero aún es demasiado voluminoso y gran consumidor de energía para teléfonos inteligentes. Por su parte, Freescale ha estado ganando terreno con sus procesadores I.MX, usado en los e-readers de Sony. Y el BCM4760 de Broadcom es el primer procesador de aplicación para integrar un completo subsistema de GPS, creando un PND-on-chip. [6]

El hardware en los dispositivos móviles es aún muy limitado en términos de poder de cálculo, memoria y reservas de energía. Las capacidades de hardware de estos dispositivos son similares a aquellas PC de escritorio de 1990, como se muestra en la tabla 2.1. [8]

Teléfono	**CPU (MHz)**	**RAM (MB)**	**Batería (tiempo de habla en horas)**
Iphone 3G	412	512	5
Android HTC G1	528	192	6
Blackberry Bold	624	128	4.5

Tabla 2.1 Especificaciones de teléfonos inteligentes, su conectividad a redes incluye: Wi-Fi, UMTS, WCDMA, HSDPA, GSM/GPRS/EDGE y Bluetooth.

El rango de arquitecturas propuestas para aplicaciones móviles va desde procesadores de aplicación específica, de propósito general **GPP** con extensiones multimedia, conocidos también como GPP de instrucciones con múltiples datos **SIMD**, procesadores multimedia de propósito especial, procesadores superescalares y de tipo **VLIW** *Very Long Instruction Word*. Así como plataformas completamente re-configurables.

Sin embargo, el rango de arquitecturas que han demostrado ser óptimas para responder acertadamente a las restricciones impuestas por las aplicaciones móviles es más reducido que la clasificación presentada anteriormente. En el caso de las arquitecturas superescalares y VLIW tradicionales, trabajan a altas frecuencias de reloj y usan complejas estructuras para la implementación de paralelismo, lo cual se traduce en un alto consumo de energía y en un incremento de los costos de la arquitectura. Para arquitecturas tradicionales SIMD, el alto costo que involucra el proceso de reorganización de datos necesario en algunos casos puede introducir grandes restricciones haciéndolos poco viables.

Los procesadores multimedia de propósito especial, tales como el STN8800 de STMicroelectronics, el S3C24A0 de Samsung, el Dragonball MC9328MX21 de Freescale y el PXA27x de Intel, tienen módulos de hardware en forma de periféricos para realizar funciones de procesamiento, lo que permite obtener un mejor desempeño. Existen por tanto, un número creciente de procesadores y arquitecturas multimedia de propósito especial diseñados con fines académicos y comerciales. En [7] se presenta un análisis de las características relevantes de algunas arquitecturas para cómputo móvil.

2.1.1 Arquitectura digital móvil, Mobydick

Arquitectura formada por un procesador de propósito general y módulos re-configurables autónomos (módulos de red, pantallas LCD, audio y video), todos ellos interconectados por un switch denominado *Octopus*. Los módulos son propuestos para integrar modos de conservación de energía. Esta arquitectura presenta un diseño centrado en la conexión por el switch *Octopus*, el cual reduce el tráfico de datos mediante la eliminación de copias innecesarias al remover la CPU del flujo de datos, mas no del flujo de control. Así, la CPU es reducida a operar como un controlador que inicializa el sistema, maneja la calidad de servicio QoS (*Quality of Service*) asociado con cada conexión y maneja la ejecución de algunos protocolos complejos.

El módulo de red incluye un módem WaveLAN, que proporciona la funcionalidad básica para enviar y recibir *frames* de datos en modo half-*duplex* 2 Mb/s; una memoria SRAM de 512 Kbytes que actúa como pila de conexión; un PGA Xilinx XC4010 que controla el flujo de datos entre el módem y el Host y proporciona funciones de corrección de errores; un microcontrolador PIC 16C66 el cual implementa el control de tráfico, el control del enlace de datos, y el protocolo MAC eficiente en energía E2MaC, que permite la conexión del sistema a una estación base a través de una red

inalámbrica ATM con un QoS apropiado. El módulo de *LCD* es de tamaño pequeño y baja resolución e incluye hardware con decodificación y post-procesamiento. [9]

2.1.2 Procesador Nomadik STN8800 de STMicroelectronics

Los procesadores de STMicroelectronics disponen de una arquitectura abierta de procesamiento distribuido, que usa aceleradores programables inteligentes para la realización en forma paralela de funciones intensivas en datos y un procesador de propósito general para el control. Como módulo CPU, el STN8800, utiliza el ARM926E-JS de 32 bits de la familia ARM9 que incluye MMU, caché de instrucciones de 32KB, caché de datos de 16KB, multiplicador de 16 x 32 bits que realiza operaciones MAC en un ciclo y un acelerador Java. Además la plataforma ARM integra una arquitectura de comunicación multicapas basada en un bus conocido como **AMBA** *Advanced Microcontroller Bus Architecture* con un bus tipo **AHB** *Advanced High-performance Bus*. Este subsistema consiste en un esquema de buses multicapas síncrono que permite realizar transferencias de datos entre módulos del sistema sin la intervención de la CPU en el flujo de datos. El procesador STN8800 posee dos módulos aceleradores, uno para audio y otro para vídeo. Éstos están basados en el core MMDSP, el cual es un Procesador de Señales Digitales **DSP** comercial de 75MHz tipo VLIW programable. El acelerador de audio soporta los estándares MPEG3, y MIDI. El acelerador de vídeo es un híbrido hardware/software que incluyen tanto al núcleo MMDSP como módulos de hardware dedicados que desempeñan funciones de pre-procesamiento, post-procesamiento, codificación y decodificación.

Estos procesadores incorporan técnicas de bajo consumo de energía, entre ellas se destacan: el uso de memoria on-chip como buffer de caché; el uso de modos de operación de bajo consumo de la CPU; la implementación de la técnica de clock-gating y la reducción del voltaje de alimentación. [10]

2.1.3 Procesador PXA270 de Intel

Los procesadores PXA, poseen características que les permiten soportar las exigencias del procesamiento multimedia en ambientes móviles. La familia PXA27x, integra un core XScale de Intel como módulo CPU, una memoria SRAM on-chip de 256KB la cual es usada como *frame buffer* para aplicaciones gráficas y de vídeo, diversos módulos multimedia (e.g. controlador de LCD e interfaz de cámara) y diferentes interfaces de periféricos (e.g. UART, USB). La arquitectura ha sido mejorada por un conjunto de nuevas tecnologías entre las cuales se incluyen: 1) la tecnología Wireless MMXTM, que proporciona funcionalidades para aplicaciones como

juegos 3D, mientras optimiza el consumo de energía; 2) la tecnología Wireless SpeedStep, que implementa técnicas de reducción de consumo de potencia; 3) la tecnología Wireless Trusted, la cual proporciona servicios de seguridad en el manejo de la información. Estos procesadores están en capacidad de operar en un rango de frecuencias desde 312MHz hasta 624MHz y con una capacidad de memoria hasta de 64 MB. El core XScale de Intel es una arquitectura tipo RISC de 32 bits basada en el core ARM V5TE con un subsistema de memoria tipo Harvard formado por cachés de instrucciones y de datos de 32KB con MMU independientes y una caché de datos de 2KB que permite almacenar datos streaming. Xscale posee una unidad de manejo de potencia, unidades JTAG, interfaz de alta velocidad para la tecnología Wireless MMXTM, y un coprocesador que permite procesar algoritmos de audio tipo multimedia **MAC** *Multimedia Algorihtms Coprocessor*. La micro arquitectura tiene la capacidad de procesar datos tipo entero de 32 bits en forma tradicional y de igual forma de 16 bits en modo SIMD. Las mejoras en las capacidades de procesamiento de la familia PXA27x se deben a la integración de la tecnología Wireless MMXTM. [7]

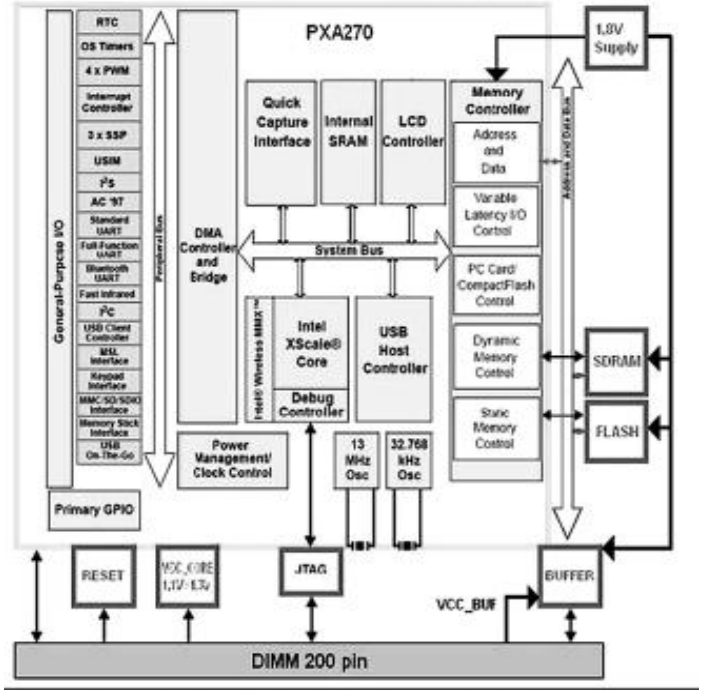


Figura 2.1 Procesador PXA270 de Intel.

2.1.4 Procesador Dragonball MC9328MX21 de Freescale

Este procesador posee como CPU el core ARM926EJ-S de 32bits de la familia ARM9, integrado en un SoC con módulos de hardware de

aplicación específica que operan en forma independiente. La CPU ARM926EJ-S es un subsistema de alto desempeño, cuya velocidad de procesamiento puede llegar hasta 266 MHz con una frecuencia máxima de reloj para el bus interno de 133 MHz, posee una arquitectura caché tipo Harvard y está formada por cuatro módulos principales: un núcleo entero ARM9EJ-S, cachés independientes de datos e instrucciones de 16KB y Unidad de Manejo de Memoria (MMU). La plataforma ARM integra el estándar AMBA-AHB, el cual permite realizar transferencias de datos entre módulos del sistema sin la intervención de la CPU en el flujo de datos. Entre los módulos específicos que posee este procesador están la Interfaz de Sensor CMOS (CSI, permite que el procesador I.MX21 se pueda conectar directamente a un sensor de imágenes CMOS externo), el Acelerador Multimedia Mejorado (eMMA) y el Controlador de Display de Cristal Líquido (LCDC, permite al procesador mostrar datos en paneles LCD).

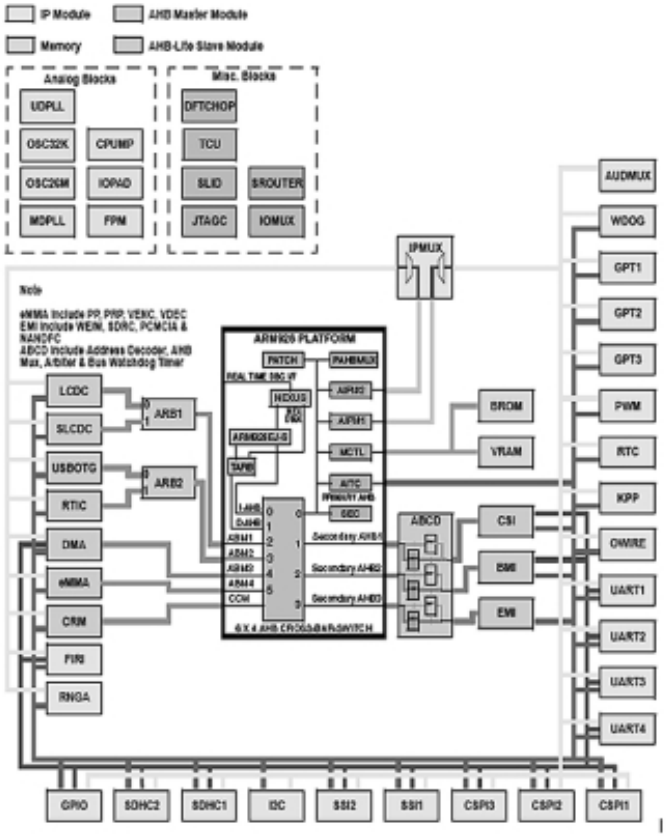


Figura 2.2 Procesador Dragonball MC9328MX21 de Freescale.

Algo importante es el uso de AMBA-AHB en el esquema de comunicación, el cual es conocido para esta tecnología como Multi-Layer AHB Crossbar Switch (MAX). Este esquema permite realizar transferencias de datos en forma simultánea y paralela entre diferentes módulos del

sistema. El estándar soporta múltiples maestros y ofrece un alto desempeño y gran ancho de banda. Los módulos manejados como maestros son: Icache y Dcache que proporcionan rutas para acceder la memoria externa, el módulo DMA y periféricos de aplicación como el CLCD, la interfaz USB y el módulo eMMA. Para reducir el consumo de potencia, el procesador MC9328MX21 soporta tres modos de operación (*run, standby, stop*). [7]

Una evaluación cuantitativa en desempeño y consumo de potencia de algunos de los procesadores orientados a aplicaciones móviles y multimedia es aportada por Synchromesh Computing, [11] el cual compara los procesadores i.MX21 de Freescale, PXA255 de Intel y el OMAP 1611 de TI. Para la realización de las pruebas se dispusieron ocho benchmarks (Hint, Stream, MPEG4 Encode, MPEG4 Decode, Sisoft Sandra, BMQ, Vobench, ByteMark) con el fin de medir diferentes parámetros en cada procesador:

	Nomadik STN8800 (sTMicroelectronics)	**PXA270 (Intel)**	**Dragonball MC9328MX21 (Freescale)**
Freq Procesador /Freq Bus	200 MHz	312MHz/104MHz 624MHz/208	MHz 266 MHz /133 MHz
Cache de Instrucciones	32KB	32KB	16K
Cache de Datos	16KB	32KB Y 2KB adicionales	16K
Memoria Interna RAM	Memoria SRAM on-chip de 48KB para almacenar los *frames* de video	SRAM de 256KB en bancos de 64KB	---
Potencia Modo Idle (mW)	100mW [22]	109 mW-260 mW (respectivamente a las frecuencias)	108mW
Potencia Modo Activo Solo manejo de UART (mW)	---	375 mW-925 mW (respectivamente a las frecuencias)	248mW
IC process	CMOS 130 nm	CMOS 130 nm	CMOS 130 nm

Tabla 2.2 Medidas comparativas entre procesadores multimedia.

Éstas fueron las conclusiones:

- OMAP1611 presenta el menor consumo de potencia (debido a la baja frecuencia del core 180MHz), pero presenta el peor desempeño

para las aplicaciones MPEG4 debido al tamaño pequeño de la caché de datos (8KB) y a la baja velocidad del bus de memoria.

- PXA255 presenta el más alto consumo de potencia con un desempeño relativo alto.
- i.MX21 a pesar de su baja velocidad de reloj respecto al PXA255 obtiene un buen desempeño con un nivel de consumo medio. De acuerdo con los autores, este procesador presenta la mejor relación desempeño/milivatios.

En la Tabla 2.2, se nota, en relación con el sistema de memoria, que todos los procesadores poseen una arquitectura tipo Harvard, con tamaños de caché entre 16KB y 32KB. Esta estructura de almacenamiento responde a las recomendaciones presentadas en [12] que establece para aplicaciones multimedia, cachés de datos con tamaños entre 16KB y 32KB debido a que ofrecen una baja tasa de error en los accesos a memoria comparada con otras aplicaciones de uso común. Igualmente para la caché de instrucciones se recomienda un tamaño de 32KB o mayor para reducir la tasa errores de acceso originadas por aplicaciones multimedia más complejas.

Por otra parte los sistemas de memoria de estos procesadores cuentan con técnicas orientadas a mejorar el desempeño y el ancho de banda tanto de las memorias caché como de la memoria principal, entre ellas se pueden citar el uso de técnicas de actualización como *write-buffer*, *write-through* y *write-back*, entre otras.

Respecto a la organización arquitectural de las plataformas, se observa que como resultado de las mejoras en los procesos tecnológicos, cada vez se integran mas componentes en un sólo chip y la comunicación entre éstos suele ser más crítica. Frente a esta situación la tendencia en el diseño de arquitecturas de comunicación está cambiando del tradicional sistema de bus compartido a estructuras de comunicación más complejas, basadas en múltiples buses que incrementan el ancho de banda [13]. Los procesadores STN8800 e IMX21 implementan el estándar AMBA, mientras que el procesador PXA270 posee una configuración tradicional en el sentido que tiene un bus central de comunicación que conecta módulos de hardware de aplicación específica a la CPU. Además, cada vez más se evidencia que la arquitectura de comunicación elegida no sólo afecta el desempeño, sino también el consumo de potencia.

Se puede concluir que en las arquitecturas embebidas orientadas a aplicaciones multimedia móviles presentadas anteriormente, el sistema de memoria y la estructura de comunicación tienen un impacto considerable en el desempeño, el costo y el consumo de potencia del sistema. La optimización de estos tres parámetros simultáneamente en el proceso de

diseño, es incompatible, ya que típicamente la mejora de uno de ellos implica el detrimento del otro. La clave para el diseño de estas arquitecturas es encontrar un punto óptimo para la relación costo/potencia considerando un desempeño suficiente para llevar a cabo la aplicación deseada de forma conveniente. [7]

2.1.5 Otras Arquitecturas

Otra arquitectura es la de un Smartphone basado en Linux, donde se diseña y se hace una implementación de hardware y software de un dispositivo móvil. El diseño de hardware contiene la mayor parte de las características para ser usados en servicios inalámbricos y la arquitectura de software que es basada en Linux. La figura 2.3 muestra la estructura de hardware del dispositivo móvil que consta de dos procesadores, un procesador de aplicación (PXA225 de Intel, alcanzando de 98 a 400 MHz) y un coprocesador (MQ1188 de nVidia) para aceleración de aplicaciones multimedia, las razones de su uso son para reducir el tráfico en el sistema de bus separando la información de tráfico entre el LCD y el *frame buffer* y para utilizar la característica de aceleración de hardware del coprocesador multimedia que ayuda a mejorar el rendimiento de aplicaciones multimedia. Esta forma de organización de hardware es constantemente utilizada en dispositivos móviles cuyo rendimiento y aplicaciones diversas son características claves. Asimismo, se puede observar que el módulo CDMA es conectado al procesador principal a través de una interfaz UART. El módulo CDMA ejecuta operaciones módem controlados por comandos AT del procesador principal. [14]

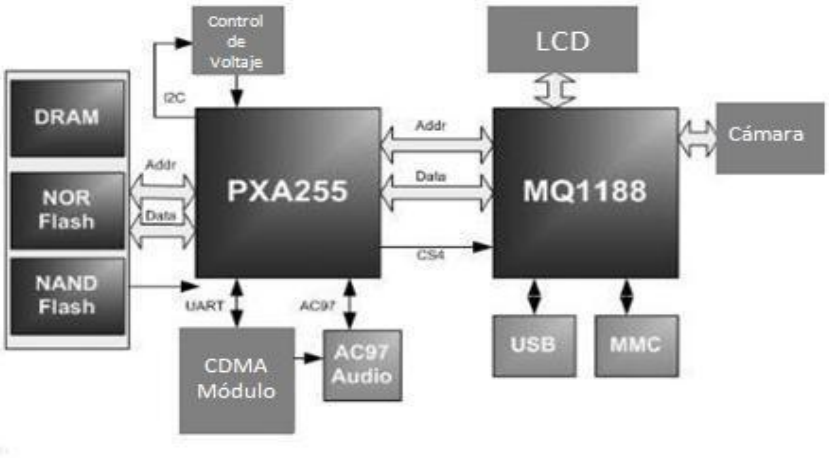


Figura 2.3 La estructura de hardware del teléfono inteligente.

Entre otras arquitecturas sobresalientes en dispositivos móviles se encuentra la del Nokia N900, con conexión a Internet vía Wi-Fi y redes 3G a través de una tarjeta SIM. Gracias a la potencia del Sistema Operativo Maemo 5 de Nokia, el terminal Nokia N900 permite a los usuarios tener docenas de aplicaciones ejecutándose simultáneamente, mientras que al mismo tiempo se puede hacer uso de las funciones de un móvil, con la ventaja de la pantalla táctil de alta resolución WVGA (800x480 píxeles) y el teclado QWERTY completo. Viene equipado con un procesador ARM Cortex-A8 con capacidad de 1GB de memoria RAM. {1}

Otra Arquitectura en dispositivos móviles actuales es el Vodafone 360 H1 de Samsung. Éste viene con una pantalla táctil y de alta resolución OLED ("*Organic Light Emitting Diode*") de 3.5 pulgadas, memoria de 16 GB, Wi-Fi y una batería con tiempo de conversación de 400 minutos, todo basado en un sistema operativo LiMo (Linux Mobile). {2}

Existe también la arquitectura HTC HD2 viene equipado con la nueva versión del sistema operativo para móviles de Microsoft, Windows Mobile 6.5, Es el primer teléfono Windows con una pantalla TFT táctil capacitiva. Viene equipado por el procesador SnapDragon de Qualcomm. Posee conectividad de banda ancha móvil HSDPA 7,2 Mbps, HSUPA 2 Mbps; Wi-Fi 802,11 b/g y receptor GPS integrado con función A-GPS. {3}

El Sistema LG-GW620 cuenta con una pantalla táctil de 3 pulgadas y un teclado QWERTY deslizable, cuenta con el sistema operativo creado por Google, el Android OS v1.5 (Cupcake). [15]

El Sistema DEXT MB220, es el primer móvil de Motorola con el sistema operativo Android. El DEXT incorpora un teclado QWERTY completo, una pantalla TFT capacitiva, 1GB de memoria interna de almacenamiento, conectividad de banda ancha móvil HSDPA, Wi-Fi y Bluetoothy y receptor GPS integrado. {4}

El Sistema Xperia X10 de Sony Ericsson es también basado en Android, viene equipado con una pantalla táctil de cuatro pulgadas e incorpora el chip SnapDragon de Qualcomm, productos y servicios de comunicaciones inalámbricas digitales basados en la tecnología digital CDMA. Dicho chip ya es usado por el TG1 de Toshiba, o el HD2 de HTC Corporation, fabricante de dispositivos portátiles basados en Windows Mobile de Microsoft y Google Android. {5}

2.2 Dispositivo Móvil

Un dispositivo móvil se puede definir como un aparato de dimensiones pequeñas que cuenta con algunas capacidades de procesamiento, conexión permanente o intermitente a una red, memoria suficiente y que ha sido diseñado específicamente para una o pocas funciones, pero que al mismo tiempo puede llevar a cabo otras funciones más generales. De acuerdo con esta definición existe una gran variedad de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores GPS, pasando por los Tablet PC, teléfonos móviles y los PDA (Asistente Personal Digital).

2.2.1 Tipos de Dispositivos Móviles

Ya que el número de niveles de funcionalidad asociado con dispositivos móviles es variado, fue necesario clasificarlos, por esta razón en el año 2005, T38 y DuPont Global Mobility Innovation Team propusieron los siguientes estándares para la definición de dispositivos móviles.

- Dispositivo Móvil de Datos Limitados: Teléfonos móviles clásicos. Se caracterizan por tener una pantalla pequeña de tipo texto y ofrecen servicios de datos limitados a SMS y acceso WAP.
- Dispositivo Móvil de Datos Básicos: Se caracterizan por tener una pantalla de tamaño medio, menú o navegación basada en íconos, ofrece acceso a e-mails, lista de direcciones, SMS y en algunos casos a un navegador Web básico. Un ejemplo de este tipo de dispositivos son los Smartphones.
- Dispositivo Móvil de Datos Mejorados. Se caracterizan por tener pantallas de medianas a grandes (por encima de los 240 x 120 píxeles), navegación de tipo stylus y que ofrecen las mismas características que el "Dispositivo Móvil de Datos Básicos". Además de aplicaciones nativas de Microsoft Office Mobile como Word, Excel, PowerPoint y aplicaciones en versión móvil como SAP, portales intranet, entre otros. Este tipo de dispositivos incluyen Sistemas Operativos como Windows Mobile.

2.2.1.1 PDA

Un PDA es una computadora de mano originalmente diseñada como agenda electrónica con un sistema de reconocimiento de escritura. Actualmente, estos dispositivos pueden realizar muchas de las funciones de una computadora de escritorio, pero con la ventaja de ser portátil. Inicialmente los PDA incluían aplicaciones estrictamente relacionadas con su función, es decir, se reducían a agenda electrónica, calendario, lista de contactos, bloc de notas y recordatorios. Hoy día, ofertan aplicaciones que incluyen juegos, acceso al correo electrónico, reproducción de películas,

crear documentos y navegar por Internet. Cuentan con pantalla sensible al tacto, ranura para tarjeta de memoria, Bluetooth y Wi-Fi.

2.2.1.2 Teléfono Móvil

El teléfono móvil es un dispositivo inalámbrico electrónico basado en la tecnología de ondas de radio que tiene la misma funcionalidad que cualquier teléfono de línea fija. Aunque su principal función es la comunicación de voz, su rápido desarrollo ha incorporado funciones adicionales como mensajería instantánea, agenda, juegos, cámara fotográfica, agenda, acceso a Internet, reproducción de video, mp3 e incluso GPS.

2.2.1.3 Smartphone

Dispositivo que funciona como un teléfono móvil con características similares a las de un ordenador personal, oscila entre el tipo de un teléfono móvil clásico y una PDA ya que permite hacer llamadas y enviar mensajes de texto como un móvil convencional, pero además incluye características cercanas a las de un ordenador personal. Casi todos los teléfonos inteligentes permiten la instalación de programas para incrementar el procesamiento de datos y conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero. Los teléfonos inteligentes cuentan con pantallas táctiles, un sistema operativo, conectividad a Internet y el acceso al correo electrónico. Otras aplicaciones que tienen presentes son las cámaras integradas, la administración de contactos, el software multimedia para reproducción de música, visualización de fotos, video-clips, programas de navegación y en algunos casos la capacidad de manejar documentos en formatos como PDF y Microsoft Office. Asimismo, la mayoría de los teléfonos inteligentes también permiten al usuario instalar programas adicionales. [16]

2.3 Sistemas operativos para dispositivos Móviles

Partiendo de la definición de sistema operativo, capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos a nivel de máquina necesarios para llevarlas a cabo. Se deduce por tanto, que el uso de uno u otro determinarán las capacidades de los dispositivos y su forma de interactuar con el usuario. Existe una multitud de opciones, las más extendidas son Palm OS, Symbian, BlackBerry OS,

Windows Mobile, iPhone OS, Android de Google y los dispositivos con sistema operativo de di

2.3.1 PALM OS

Palm OS es un sistema operativo propietario destinado a dispositivos móviles, específicamente a PDA. En 1996 Palm OS comenzó su desarrollo, un año después Palm Inc. comenzó a licenciarlo con novedosos aparatos como la PalmPilot. Las primeras versiones de este sistema operativo estuvieron basadas en un S.O. multitareas creado por Motorola.

2.3.2 Symbian

Este es el sistema operativo para móviles más extendido entre “smartphones” y por tanto el que más aplicaciones desarrolladas tiene para su sistema. Symbian ocupa más del 65 % del mercado. Nació como resultado de la alianza entre varias empresas multinacionales y de gran renombre en el mercado, tales como Nokia, Sony Ericsson, Samsung, Motorola, entre otras. Sus orígenes provienen del EPOC32, otro sistema operativo para dispositivos móviles, originado a principios de los 90s. Después de algunos años pasaría a llamarse Symbian OS. Posee un núcleo de tiempo real, cuenta con un microkernel y capacidad multithreading, soporta las arquitecturas de los últimos CPU e incluso soporta hardware "single-chip", tiene un sistema de archivos de alta performance que soporta las últimas memorias NOR, NAND, SD (Secure Digital) y MMC (Multimedia Card). Symbian OS supone una plataforma abierta, ésta es la clave que une a las telecomunicaciones y a los estándares globales de Internet que de igual forma lo hace compatible con los estándares de conectividad y redes como Bluetooth, Wi-Fi, GSM/GPRS y WCDMA.

2.3.3 Windows Mobile

Windows Mobile es un sistema operativo de la familia Windows CE, desarrollado por Microsoft. A pesar de llevar el nombre Windows, no es un sistema derivado ni es una versión recortada del mismo, sino un nuevo sistema diseñado específicamente para dispositivos móviles. Este sistema operativo está estrechamente vinculado a otros productos de la misma marca (Office Mobile, Internet Explorer Mobile, etc.) y cuenta con una interfaz gráfica similar a la de los sistemas operativos Windows. Lo anterior, ayuda a disminuir la curva de aprendizaje de los usuarios pues proveen un entorno de trabajo muy similar al que se tiene en el hogar o en la oficina. Asimismo, cuenta con procesadores de arquitecturas x86, ARM, SH4 y MIPS. Una de las ventajas de Windows Mobile es que los programadores pueden

desarrollar aplicaciones para móviles utilizando los mismos lenguajes y entornos que emplean con Windows para PC.

2.3.4 Iphone OS

La historia del IPhone OS comienza con el nacimiento del IPhone en el año 2007. Sin embargo, se ha discutido que el sistema operativo que corre en el Iphone es en realidad una versión adaptada del OS X, por lo cual, en este sentido, este sistema ya tiene años en el mercado y ha sido puesto a prueba. El sistema ha sido adaptado, removiendo los componentes que no son críticos para un dispositivo móvil y se le adicionan funcionalidades que sí están relacionadas con el mundo de la telefonía móvil y tiene una gran inclinación en el desarrollo del sistema operativo a la interfaz del usuario. El iPhone es el sistema operativo para dispositivos móviles que brinda una mejor experiencia de usuario, con un modo de manejo revolucionario basado en su Pantalla táctil *Touch Screen*, implementado mediante el framework Cocoa Touch desarrollado por Apple. Éste ya superó a Windows Mobile en la cuota de mercado que abarca. IPhone OS es una versión reducida de Mac OS X optimizada para los procesadores ARM.

2.3.5 Android

Android OS es el más reciente de los sistemas operativos para móviles del mercado. Android está siendo desarrollado por The Open Handset Alliance, un grupo de más de 30 empresas de tecnología, entre ellas la principal participante es Google. Android es un sistema operativo abierto, multitarea que permite a los desarrolladores acceder a las funcionalidades principales del dispositivo mediante una API (Interfaz de Programación de Aplicaciones). Además, el sistema operativo no diferencia entre las aplicaciones básicas del teléfono y las aplicaciones de terceros, cualquier aplicación puede ser reemplazada libremente, incluso las que trae por defecto. Además, cuenta con soporte para gráfico 2D y 3D basado en la especificación Open GL 1.0 y con conectividad Bluetooth, EDGE (Enhanced Data Rates for GSM Evolution), 3G y Wi-Fi.

Una característica sobresaliente es que se basa en el kernel de Linux versión 2.6 para las principales funciones como manejo de memoria, procesos, networking y modelo de driver. Android dio a conocer un SDK (Software Development Kit) para que los desarrolladores que lo deseen puedan programar aplicaciones que corran en el sistema operativo con el lenguaje de programación Java. Las aplicaciones corren sobre una máquina virtual diseñada para ser usada de forma embebida denominada Dalvik, la cual se ejecuta sobre el kernel de Linux donde cada aplicación en Android corre en su proceso con su propia instancia de la máquina virtual Dalvik.

2.3.6 BlackBerry OS

BlackBerry es un sistema operativo multitarea que está arrasando en la escena empresarial, en especial por sus servicios para correo y teclado QWERTY. BlackBerry apareció en el mercado cuando comenzaba a demandarse un sistema operativo que permitiera utilizar de una forma fácil, cómoda y rápida los servicios de correo electrónico. BlackBerry es un dispositivo que permite redactar, enviar y recibir todo tipo de mensajes de correo electrónico, de la misma forma que en el programa que se utiliza en un ordenador, incluso, es proveedor de servicios de correo electrónico a dispositivos de otras marcas. Además, es posible realizar y contestar llamadas que se emitan a través de la red de telefonía móvil, lo cual permite sustituir el teléfono celular. Asimismo, los dispositivos de este fabricante permiten la navegación por Internet y tienen la capacidad de enviar o recibir mensajes SMS.

2.4 Comparación de Sistemas Operativos para Móviles

El Kernel de un sistema operativo es el núcleo del mismo, es el software responsable de facilitar a los distintos programas acceso seguro al ordenador o en una forma más simple, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Por ello, es importante conocer qué núcleo utiliza cada uno de los sistemas operativos de ésta comparativa.

La principal diferencia entre un kernel de libre distribución y uno propietario es en que los de libre distribución como Linux cuentan con una amplia y experimentada comunidad de desarrolladores, gracias a los cuales se detectan rápidamente agujeros de seguridad, fallos, etc., y se realizan mejoras para lograr solucionar estos problemas, así como para adaptarse a los nuevos tiempos. En los sistemas cerrados o propietarios, el costo para encontrar errores y mejorarlos es más elevado, ya que deben ser los propios desarrolladores del sistema los que detecten y realicen las mejoras, por lo que deben dedicarse más recursos a la investigación.

	Android	BlackBerry OS 4.7	iPhone OS 3.0	S60 5th Edition	Palm WebOS	Windows Mobile 6.5
Kernel	Linux con máquina virtual Dalvik	Propietario	OS X	Symbian	Linux	Windows CE
Conectividad	3G, WiFi, GSM, GPRS	3G, GSM, CDMA, WiFi	3G, GSM, WiFi	3G. GSM, WiFi	3G, GSM, CDMA, WiFi	3G, GSM, CDMA, WiFi

Tabla 2.3 Comparativa entre sistemas operativos para móviles.

La “Adaptabilidad” es la capacidad o facilidad para poder adaptarlo a diferentes terminales o en diferentes máquinas. En este sentido, Android es el que mayor adaptabilidad presenta, ya que cada vez se está empleando en más dispositivos, no sólo en teléfonos móviles, sino también en netbooks y como sistema empotrado. En cambio en el resto de los sistemas operativos el nivel de adaptabilidad es menor y más complicado.

La “Conectividad” es primordial para poder sacar el máximo provecho a las funciones que ofrece cada sistema operativo por lo que es indispensable contar con acceso a Internet. En este sentido, se valora de gran manera el hecho de contar con acceso Wi-Fi a Internet, así como conectividad 3G que permitan el acceso a Internet desde cualquier lugar. [16]

2.5 Conclusiones

En este capítulo se puede concluir que las arquitecturas actuales de dispositivos móviles cuentan con una estructura de hardware que consta de dos procesadores, un procesador de aplicación y un coprocesador para la aceleración de aplicaciones multimedia. Texas Instruments ha dominado en procesadores de aplicación; su procesador OMAP aparece en teléfonos como, el Palm Pre, Motorola Droid, y en muchos de los teléfonos inteligentes de Nokia. El procesador OMAP tiene una arquitectura Dual-Core con un procesador ARM como procesador de aplicación y un DSP como coprocesador, por lo que será el procesador OMAP el usado como componente principal en la arquitectura del dispositivo móvil a desarrollar.

Con respecto a los sistemas operativos para dispositivos móviles se concluyó que Android es el sistema operativo más popular debido a que es el que mayor adaptabilidad presenta, ya que cada vez se está empleando en más dispositivos, no sólo en teléfonos móviles, sino también en netbooks. En cambio en el resto de los sistemas operativos el nivel de adaptabilidad es menor y más complicado, además de que Android cuenta con kernel de Linux de libre distribución. La conectividad en Android es primordial para contar con acceso a Internet, en este sentido, cuenta con soporte para Wi-Fi y conectividad 3G que permiten el acceso a Internet desde cualquier lugar. Por lo tanto, el sistema operativo Android será usado para correrlo en un procesador OMAP y para correr aplicaciones, en este caso una aplicación para hacer más eficiente las comunicaciones.

CAPÍTULO 3

MARCO TEÓRICO

En este capítulo se hace un estudio de las redes Wi-Fi y GSM. Con respecto a la redes Wi-Fi se verán los aspectos más importantes con relación al estándar IEEE 802.11, incluyendo la comunicación desde un móvil con un Punto de Acceso, se describen los diferentes paquetes que maneja el estándar para establecer la comunicación. Se presentan las características de las redes celulares y las características principales de cada una de las generaciones, la arquitectura de las redes GSM, la forma en que trabaja un teléfono celular, etc. Se explica al lector las características básicas del nuevo sistema operativo Android, su diseño, arquitectura y funcionamiento, así como una guía de instalación para el entorno de desarrollo con Eclipse.

3.1 Redes Inalámbricas

Las Redes Inalámbricas facilitan la operación en lugares donde dispositivos no puede permanecer en un sólo lugar. Según su cobertura, se pueden clasificar en: WPAN, WLAN, WMAN y WWAN.

3.1.1 WLAN Wi-Fi

WLAN es una de las áreas de más rápido crecimiento de las tecnologías inalámbricas debido a su flexibilidad y a la velocidad en transferencia de datos. Muchas WLAN están basadas sobre las tecnologías de espectro ensanchado y la conectividad a redes cableadas es suministrada a través de un AP que puede ser conectado a una LAN cableada o a cualquier tipo de red para acceder a Internet o a bases de datos corporativas. Los dispositivos móviles se conectan al AP cuando ellos están dentro del rango (una celda que puede expandirse de 10 a 100 metros). Una vez conectado al AP, el dispositivo móvil puede comunicarse con otros dispositivos en la celda u otros recursos a través del AP.

La figura 3.1 muestra una simple configuración WLAN. Cada dispositivo móvil en la WLAN tiene un adaptador WLAN que opera en ciertos rangos de frecuencia. La conectividad a redes cableadas es suministrada a través de un AP. [17]

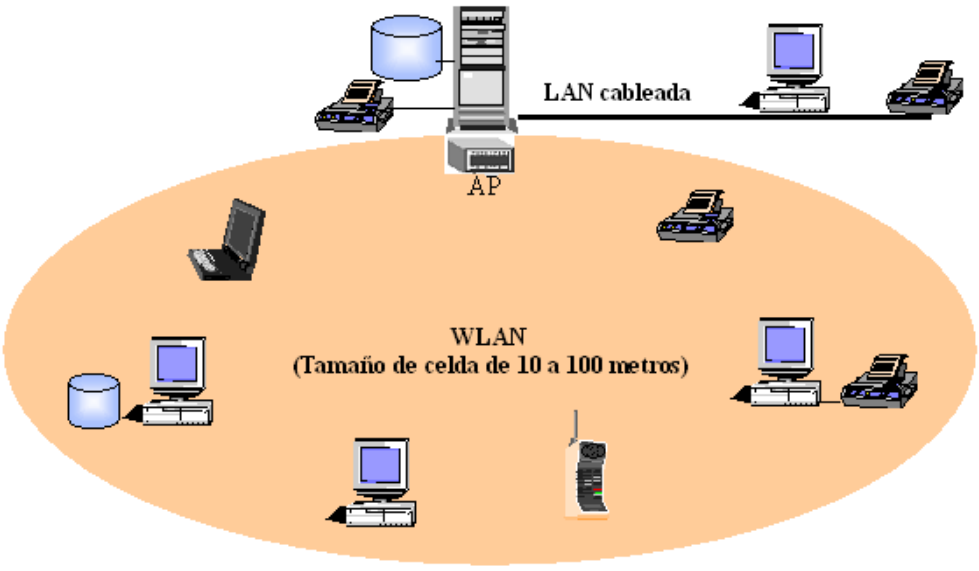


Figura 3.1 Simple configuración WLAN.

El estándar 802.11b es muy popular, aunque 802.11g está ganando terreno. 802.11b conocido como Wi-Fi soporta velocidades de hasta 11 Mbps. Las LAN IEEE 802.11b operan de forma muy similar a las redes cableadas Ethernet LAN, con la diferencia de que no hay cables y los paquetes de datos son enviados sobre ondas de radio. Estas LAN usan la banda de frecuencias de 2.2 – 2.4835 GHz y no se requiere licencia alguna para trabajar en este rango de frecuencias.

IEEE	FRECUENCIA	VELOCIDAD	COMENTARIOS
802.11a	5 GHz	54 Mbps	Esta banda ha sido recientemente habilitada y no existen otras tecnologías que la usen.
802.11b	2.4 GHz	11 Mbps	Mayor alcance comparada con la banda de 5 GHz debido a la frecuencia.
802.11g	2.4 GHz	54 Mbps	Tecnologías como Bluetooth funcionan a esta misma frecuencia por lo que pueden presentar interferencia con Wi-fi.

Tabla 3.1 Versiones del estandard IEEE 802.11.

3.1.1.1 Manera en que trabaja el estándar 802.11

Las LAN 802.11b están estandarizadas por señales de radio DSSS (Espectro ensanchado por secuencia directa), este esquema divide el espectro de frecuencias en 14 canales ligeramente traslapados, cada uno de 22 MHz. De manera que, si cada WLAN es configurada para usar un canal, entonces una oficina de un edificio puede operar 14 WLAN en el mismo espacio físico. Los transmisores en cada canal "expanden" sus señales en los 22 MHz para mejorar la recepción. DSSS es una técnica que utiliza un código de pseudo-ruido para modular directamente una portadora, de tal forma que aumente el ancho de la banda de transmisión y reduzca la densidad de potencia espectral (es decir, el nivel de potencia en cualquier frecuencia dada). La señal resultante tiene un espectro muy parecido al del ruido, de tal forma que a todos los radiorreceptores les parecerá ruido menos al que va dirigida la señal. Con DSSS se genera un patrón de bits redundante para cada uno de los bits que componen la señal, cuanto mayor sea este patrón de bits, mayor será la resistencia de la señal a las interferencias. El estándar IEEE 802.11 recomienda un tamaño de 11 bits, pero el óptimo es de 100. En recepción es necesario realizar el proceso inverso para obtener la información original. [17]

Una característica atractiva de 802.11b es que alrededor de 100 usuarios pueden ser soportados en un canal (cada celda puede soportar arriba de 100 usuarios).

3.1.1.2 Arquitectura del estándar IEEE 802.11

El estándar 802.11 define dos modos operativos, el modo de infraestructura en el que los clientes de tecnología inalámbrica se conectan a un AP y el modo ad-hoc en el que los clientes se conectan entre sí sin ningún AP. [17]

En el modo infraestructura la arquitectura está dividida en:

- Estaciones.
- Puntos de Acceso.
- Conjunto de Servicio Básico (BSS): Estaciones compitiendo por el acceso al medio inalámbrico compartido.
- Sistema de Distribución (DS): Interconecta varios BSS
- Conjunto de Servicio Extendido (ESS): Dos o más BSS interconectados por un DS.

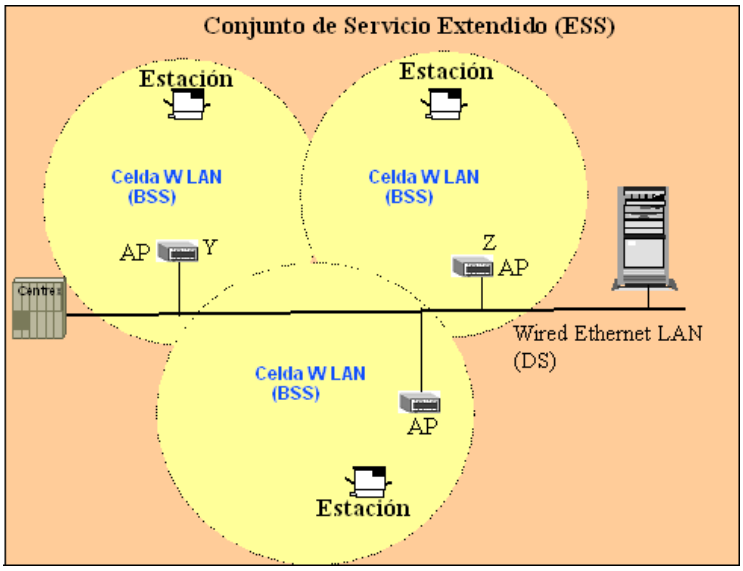


Figura 3.2 Modo infraestructura del estándar 802.11.

3.1.1.3 Handoff

Cuando un usuario va desde un BSS a otro mientras se mueve dentro del ESS, el adaptador de la red inalámbrica de su equipo puede cambiarse de AP, según la calidad de la señal que reciba desde distintos AP. Éstos se

comunican entre sí a través de un SD con el fin de intercambiar información sobre las estaciones y si es necesario también para transmitir datos desde estaciones móviles. Esta característica que permite a las estaciones moverse "de forma transparente" de un AP al otro se denomina “itinerancia”. [17]

3.1.1.4 Servicios del Estándar IEEE 802.11

IEEE 802.11 especifica varios servicios que pueden ser categorizados en servicios relacionados con asociación, distribución y privacidad. [17]

3.1.1.4.1 Relacionados con Asociación

En ambientes móviles, los servicios de asociación son necesitados para identificar que estaciones están asociadas con el AP. Ya que es muy probable la movilidad (por ejemplo, algunas estaciones pueden no moverse mucho mientras otras están siempre en movimiento), el estándar define diferentes tipos de transición basadas en movilidad, éstas son las siguientes:

a) Sin transición. Una estación no se mueve o lo hace sólo en el BS.
b) Transición de BSS. Una estación se mueve de un BSS a otro BSS en el mismo ESS.
c) Transición ESS. Una estación se mueve de un BSS en un ESS a un BSS en otro ESS. Para entregar un mensaje el servicio de distribución necesita conocer la identificación del AP al cual el mensaje debe ser entregado, de manera que se alcance la estación destino, por lo que una estación debe mantener siempre una asociación con un AP. Los siguientes tres servicios soportan este requerimiento:

- Asociación. Establece una asociación inicial entre la estación y el AP.
- Re-asociación. Hace posible transferir de asociación de un AP a otro, permitiendo a la estación moverse de un BSS a otro.
- Des-asociación. Aviso de terminación de asociación de la estación o del AP.

3.1.1.4.2 Relacionados con Distribución

Estos servicios son necesitados para distribuir mensajes en un DS y son de dos tipos:

- Servicio de distribución: Usado para intercambiar tramas MAC de una estación en un BSS a una estación a otra BSS.
- Servicio de Integración: Usado para transferir información entre estaciones en una LAN IEEE 802.11 y estaciones en una LAN

cableada, esto soporta el intercambio de información entre LAN inalámbricas y estructuradas.

3.1.1.4.3 Relacionados con Privacidad y Acceso

Estos servicios tienen requerimientos especiales en el ambiente móvil porque las estaciones pueden moverse alrededor y asociarse con un AP que se encuentre en los rangos de frecuencia del transmisor. Los siguientes 3 servicios son diseñados para adecuar seguridad:

- Autenticación. Servicio usado para establecer identidad de estaciones.
- Des-autenticación. Servicio invocado cuando una autenticación existente es terminada.
- Privacidad. Previene que el contenido de mensajes sean leídos por un receptor no intencionado.

3.1.1.5 Comunicación con Punto de Acceso

Cuando una estación se une a una célula envía una solicitud de sondeo a cada canal. Esta solicitud contiene el ESSID que la célula está configurada para usar y también el volumen de tráfico que su adaptador inalámbrico puede admitir. Si no se establece ningún ESSID, la estación escucha a la red para encontrar un SSID.

Cada AP transmite una señal en intervalos regulares (10 veces por segundo, aproximadamente). Esta señal, que se llama señalización, provee información de su BSSID, de sus características y su ESSID, sí corresponde.

Cuando se recibe una solicitud de sondeo, el AP verifica el ESSID y la solicitud del volumen de tráfico encontrado en la señalización, si el ESSID concuerda con el del AP, éste envía una respuesta con datos de sincronización e información sobre su carga de tráfico. Así, la estación que recibe la respuesta puede verificar la calidad de la señal que envía el AP para determinar que tan lejos se encuentra (mientras más cerca un AP, su capacidad de transferir datos será mayor). Por lo tanto, una estación dentro del rango de muchos AP (que tengan el mismo SSID) puede elegir el AP que le ofrezca la mejor proporción entre capacidad de carga de tráfico y carga de tráfico actual. [17]

La información se transmite por ondas de RF fragmentada en "paquetes" de distintos tamaños. El estándar 802.11 establece los siguientes paquetes:

- Paquetes de Administración: Establecen y mantienen la comunicación.

- Paquetes de control: Ayudan a la entrega de datos y tiene funciones de coordinación.
- Paquete de Datos.

3.1.1.5.1 Paquetes de Administración

- Solicitud de Asociación. Es el que inicia la conexión y se encarga de llevar diversa información, entre ella el SSID del AP al cual nos queremos conectar. Si el AP está interesado en esa comunicación, asigna espacio de memoria para ese cliente.
- Respuesta de Asociación. Paquete que lleva la respuesta a la estación cliente, si acepta o no acepta e incluye informaciones diversas, entre ellas la dirección MAC del cliente.
- Autenticación. Paquete por medio del cual el AP comunica a la estación si la acepta definitivamente o la rechaza (usado cuando en la red Wi-Fi existen medidas de seguridad y se requiere autenticar a los usuarios). Esto se logra intercambiando varios de estos paquetes con "desafíos" y respuestas para verificar la identidad del cliente. En redes "abiertas" donde no existen estas medidas de seguridad no se usa el paquete.
- Des-asociación. Paquete que se usa cuando la estación cliente desea finalizar la conexión, es entonces cuando el AP sabe que puede utilizar los recursos que había asignado a esa estación.
- Beacon. Los AP envían periódicamente "señales" para anunciar su presencia y que todas las estaciones que estén en el rango sepan cuáles AP están disponibles. Estos paquetes contienen varios parámetros entre ellos el SSID del AP y su dirección MAC. Esta información será utilizada por el cliente a la hora de comunicarse.

3.1.1.5.2 Paquetes de Control

- Request to Send (RTS). Es la primera fase antes de comenzar a enviar paquetes de datos y su función es la de evitar colisiones.

- Clear to Send (CTS). Tiene la función de responder a los RTS e Indica que el canal está libre para transmitir. Todas las estaciones que captan el CTS saben que deben esperar un tiempo para transmitir pues alguien está utilizando ya el canal. Existe un tiempo de espera llamado "Slot-time" (distinto para cada estándar, 802.11b, 802.11g, etc.).
- Acknowledgement (ACK). La estación receptora checa el paquete recibido por si tiene algún error, si lo encuentra correcto envía un "ACK", con lo cual el remitente sabe que el paquete llegó bien y si no,

debe ser enviado otra vez. Una vez que las demás estaciones captan el ACK, saben que el canal está libre y que ya pueden intentar enviar sus paquetes.

3.1.2 Red Celular WWAN

La Red Celular consta de muchas celdas, donde cada una es administrada por una Estación Base (BS). Un Centro de Conmutación Móvil (MSC) controla la Red Celular entera. La asignación de frecuencias y localización son dos de los puntos primordiales de redes celulares. En los sistemas celulares, el área de cobertura de un operador es dividida en celdas. Una celda corresponde a una zona cubierta por un transmisor o una pequeña colección de transmisores. El tamaño de la celda depende de la potencia del transmisor, a la banda de frecuencia utilizada, altura y posición de la torre de la antena, el tipo de antena, la topografía del área y la sensibilidad del radio receptor. [18]

3.1.2.1 Evolución de las Redes celulares

Las distintas necesidades y avances en tecnología dieron lugar a generaciones tecnológicas en las redes Celulares bien diferenciadas que se comentan a continuación.

En dicha evolución se aprecia cómo se van cumpliendo las necesidades del mercado para tener acceso múltiple al canal de comunicación, así como la necesaria migración de los sistemas analógicos a sistema digital con el fin de permitir mayor volumen de usuarios y ofrecer los niveles de seguridad que se demandaban. [18]

3.1.2.1.1 Red Celular de 1G

Su capacidad analógica de mantener la comunicación en movimiento es su característica principal. Además, trabaja con sistemas de circuitos conmutados, únicamente ofreciendo servicios de voz. Estas redes son muy lentas menor a 1 Kbps. En esta generación de Red Celular destaca AMPS (Advanced Mobile Phone Systems). Algunos problemas que tiene la Red Celular de 1G es que sus teléfonos celulares analógicos son inseguros y cualquiera con un receptor de radio banda puede entrometerse.

3.1.2.1.2 Red Celular de 2G

En los años 90 surge la telefonía digital o de segunda generación 2G, con diferentes tipos de tecnologías usadas en USA (IS-95) y Europa (GSM).

Los avances en la tecnología de materiales y semiconductores permitieron el salto, traducido en comunicaciones mejores, más rápidas y mas servicios. Así, surge el SMS entre terminales provocando una explosión comercial de celulares a nivel mundial que perdura hasta estos días, pues los 2G dominan ampliamente el mercado mundial. En 2G empiezan a surgir servicios de comunicación de datos pero sobre circuitos conmutados que llegan a transmitir a velocidades de 56 Kbps. En cambio en 2.5 G surgen opciones de transmitir datos a una mayor velocidad.

Los dispositivos 2G usan transmisión digital y son intencionados para voz, sin embargo soportan servicios de fax y correo electrónico a baja velocidad (8 a 9 Kbps). En esta generación de Red Celular destacan redes como la GSM y CDMA.

Entre las ventajas de la comunicación digital podemos encontrar que servicios de voz, datos y fax pueden ser integrados en un sólo sistema, además permite una mejor compresión que facilita la utilización del canal. También algunos códigos de corrección de error pueden ser usados para una mejor calidad. Los sistemas de 2G permiten encriptación y el uso de tipos de acceso al canal que permiten a los canales ser dinámicamente compartidos por usuarios que usen TDMA o CDMA.

3.1.2.1.3 Red Celular de 2.5 G

Sistemas 2.5 son esencialmente sistemas 2G que han evolucionado a una velocidad media (alrededor e los 100 Kbps), un ejemplo de 2.5G es el GPRS, una extensión de GSM que soporta velocidades de 112 Kbps y que permite la transición de servicios de datos a través de conmutación de paquetes.

3.1.2.1.4 Red Celular de 3G

La visión de 3G es crear un conjunto global unificado de estándares que puedan dirigir el despliegue comercial de comunicaciones inalámbricas multimedia avanzadas. El ejemplo mejor conocido de 3G es el UMTS (Universal Mobile Telecommunications System). El 3G es una evolución del 2G, el modo dual de las terminales facilitan la migración de 2G a 3G.

Sus retos principales son donar con una mayor capacidad de ancho de banda y velocidad para la transmisión de servicios de datos así como una calidad de servicio que permita ofrecer tanto servicios de voz como de datos en tiempo real. Los sistemas 3G pueden operar a 2Mbps. Además a los servicios de voz datos y fax convencionales, 3G ofrece video de alta resolución y servicios multimedia en movimiento. La calidad de voz es

comparable a la PSTN (Red Telefónica Publica Conmutada), esto implica una más alta calidad de voz que las redes 2G.

En redes 3G las velocidades de datos van desde 144 Kbps disponibles para usuarios en vehículos de alta velocidad, 384 Kbps disponibles para dispositivos en movimiento lento y puede soportar hasta 2.048 Mbps para uso en oficina. 3G soporta servicios de datos de conmutación de circuitos y conmutación de paquetes. Ya que ahora es el tamaño de los datos lo que modifica las facturas, existe la posibilidad de estar conectado permanentemente. El usuario sólo pagará por aquellos datos que se descargue o transmita, no importa que esté conectado sólo el tiempo justo que dura la operación o durante todo el día. Además con respecto del uso de una red 2.5G y una 3G, no es necesario cambiar la tarjeta SIM. Aunque, sí es necesario cambiar el terminal y las tarjetas SIM son compatibles en ambos modelos (GPRS y UMTS).

3.1.2.1.5 Red Celular de 4G

Su objetivo es alcanzar velocidades de 100 Mbps de recepción y 50 Mbps de transmisión.

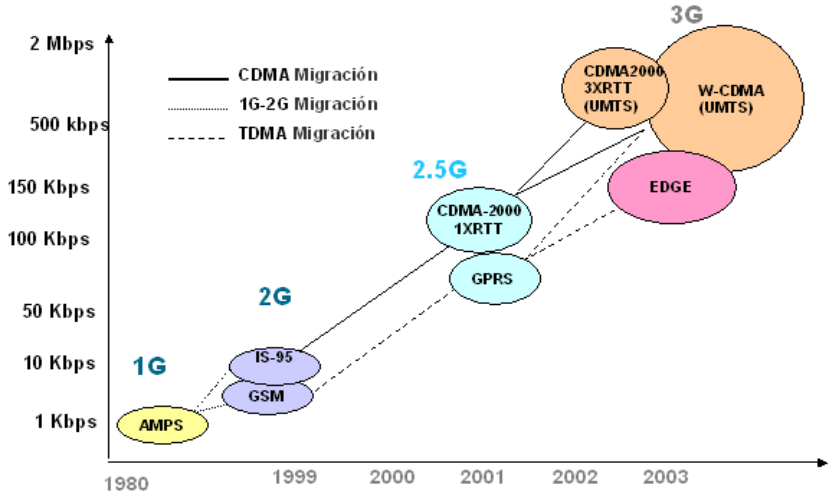


Figura 3.3 Evolución a 3G.

3.1.2.2 Forma en que trabaja un Teléfono Celular

El usuario enciende el teléfono e inmediatamente el dispositivo empieza a escanear para encontrar las señales de control más fuertes desde varias BS. El teléfono detecta y selecciona la BS con la señal de control más fuerte como su celda local. En muchos casos, un usuario es

asignado a la BS más cercana. La BS seleccionada informa al MSC acerca de la localización del usuario. Por su parte, el MSC registra al usuario en un directorio para indicar su localización. [18]

3.1.2.2.1 Handover

Siempre una estación móvil está en una celda bajo el control de un BS, sí la estación móvil deja una celda, por ejemplo, la celda de BS1, la BS1, ésta empezará a detectar una señal débil, entonces BS1 preguntará a las BS a su alrededor si ellos están teniendo una señal más fuerte, de manera que la BS1 transferirá el control de la estación móvil a la BS con la señal más fuerte, por lo que la BS2 informará a la MSC acerca de la localización del usuario. El MSC entonces registra al usuario, asigna un nuevo canal a la estación móvil y la notifica de su nueva BS. Todo el proceso es llevado a cabo sin alguna interrupción.

3.1.2.3 Red GSM

Hoy en día el número de usuarios de telefonía móvil y de Internet han crecido de una manera increíble. Debido a esto, fue inevitable que en algún momento ambos mundos se fusionarán. El sistema GSM es el sistema de comunicación de móviles digital de 2G basado en células de radio. Éste apareció para dar respuestas a los problemas de los sistemas analógicos y fue diseñado para la transmisión de voz por lo que se basa en la conmutación de circuitos, aspecto del que se diferencia del sistema GPRS. Al realizar la transmisión mediante conmutación de circuitos los recursos quedan ocupados durante toda la comunicación y la tarificación es por tiempo. Estas limitaciones hacen ineficiente la transmisión de datos con GSM. La tecnología GPRS lo soluciona. La tecnología GSM trabaja en el rango de frecuencias de 890 a 960 MHz, donde la banda de transmisión de la BS va de 890 a 915 MHz, con un ancho de banda en cada canal de 200 Khz, lo que da como resultado 125 canales de frecuencia, donde cada canal soporta 8 usuarios a través de TDM y alcanza un máximo de 992 usuarios (en práctica aproximadamente 500 debido a que hay canales usados para control de señales). [18]

3.1.2.3.1 Arquitectura de una red GSM

Todas las redes GSM se pueden dividir en tres partes fundamentales y bien diferenciadas.

1.- La Estación Móvil (MS). Consta de dos elementos básicos, por un lado el terminal o equipo móvil y por otro lado el SIM (Módulo de Identidad del Suscriptor). La diferencia entre terminales radica en la

potencia que tiene que va desde los 20 vatios (generalmente instalados en vehículos) hasta los 2 vatios. El SIM es una pequeña tarjeta inteligente que sirve para identificar las características de nuestro terminal, ésta se inserta en el interior del móvil y permite al usuario acceder a todos los servicios que haya disponibles por su operador, sin la tarjeta SIM el terminal no nos sirve de nada porque no podemos hacer uso de la red. Una vez que se introduce el SIM en el terminal, éste buscará redes GSM que estén disponibles y tratará de validarse en ellas, una vez que la red ha validado nuestro terminal, el teléfono queda registrado en la célula que lo valide.

2.- La Estación Base Subsystem (BSS). Sirve para conectar a MS con los NSS, se encargan de la transmisión y recepción. Asimismo, constan de dos elementos diferenciados: La BS y el BSC. La BS consta de transreceptores y antenas usadas en cada célula de la red y suelen estar situadas en el centro de la célula. Generalmente su potencia de transmisión determina el tamaño de la célula. Por su parte, los BSC se utilizan como controladores de los BS y tienen como funciones principales las de estar al cargo de los *handovers*, los saltos de frecuencia y los controles de las frecuencias de radio de los BS. [18]

3.- El Subsistema de Conmutación y Red (NSS). Este sistema se encarga de administrar las comunicaciones que se realizan entre los diferentes usuarios de la red. Para poder hacer este trabajo la NSS se divide en siete sistemas diferentes, cada uno con una misión específica dentro de la red:

- Centro de Conmutación de Servicios Móvil (MSC). Es el componente central del NSS y realiza labores de conmutación dentro de la red, así como de proporcionar conexión con otras redes.
- Centro de Conmutación de Servicios Móvil Gateway (GMSC). Un Gateway es un dispositivo traductor, se encarga de interconectar dos redes provocando que los protocolos de comunicaciones que existen en ambas redes se entiendan. La misión del GMSC es servir de mediador entre las redes de telefonía fija y la red GSM.
- Home Location Registrer (HLR). Es una base de datos que contiene información sobre los usuarios conectados a un determinado MSC. Almacena la localización del usuario y los servicios a los que tiene acceso.
- Visitor Location Registrer (VLR). Contiene la información necesaria sobre un usuario para que éste acceda a los servicios de la red. También forma parte del HLR con quien comparte funcionalidad.

- Centro de Autenticación (AuC). Proporciona los parámetros necesarios para la autenticación de usuarios dentro de la red y soporta funciones de encriptación.
- Equipment Identy Registrer (EIR). También se utiliza para proporcionar seguridad en las redes GSM, pero a nivel de equipos validos. La EIR contiene una base de datos con todos los terminales válidos para ser usados en la red. Esta base contiene los IMEI (International Mobile Equipment Identy) de cada terminal, de manera que si un determinado móvil trata de hacer uso de la red y su IMEI no se encuentra localizado en la base de datos del EIR, simplemente no puede hacer uso de ésta.
- GSM Interworking Unit (GIWU). Sirve como interfaz de comunicación entre diferentes redes para comunicación de datos.

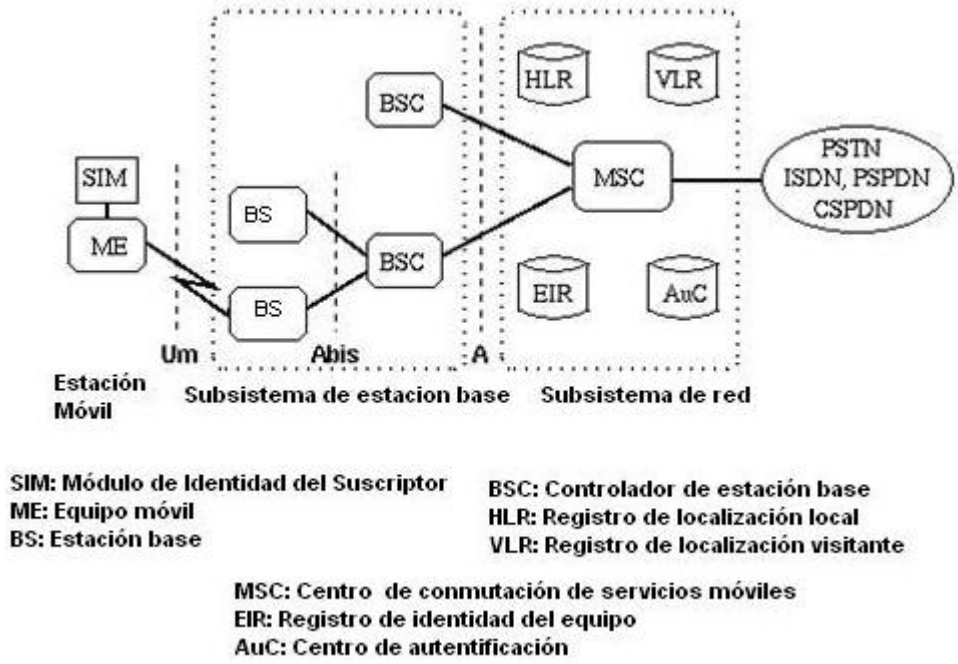


Figura 3.4 Arquitectura de la Red GSM.

Las redes GSM tienen ciertas limitaciones para la transmisión de datos, por ejemplo la velocidad de transferencia es de 9,6 Kbps, el tiempo de establecimiento de conexión es de 15 a 30 segundos, además se paga por tiempo de conexión.

La baja velocidad de transferencia limita la cantidad de servicios que Internet nos ofrece, por ejemplo, a 9,6 Kbps no se puede navegar por Internet de una manera satisfactoria, y si además, tenemos en

cuenta que estamos pagando por tiempo de conexión, los costos se disparan. Ésta es la eterna lucha, pues no se puede comparar una hora de conversación, con una hora de navegar por Internet. La combinación de estos tres factores negativos hace que GSM sea una tecnología mayoritariamente utilizada para la voz y no para datos.

Las tradicionales redes GSM no se adaptan adecuadamente a las necesidades de transmisión de datos con terminales móviles. Por ello surge una nueva tecnología portadora denominada GPRS que unifica el mundo IP con el mundo de la telefonía móvil, creándose toda una red paralela a la red GSM y orientada exclusivamente a la transmisión de datos. Al sistema GPRS se le conoce también como GSM-IP ya que usa la tecnología IP para acceder directamente a los proveedores de contenidos de Internet. [18]

3.1.2.4 Red GPRS

GPRS es una nueva tecnología que comparte el rango de frecuencias de la red GSM utilizando una transmisión de datos por medio de paquetes. La conmutación de paquetes es el procedimiento más adecuado para transmitir datos, hasta ahora los datos se habían transmitido mediante conmutación de circuitos, procedimiento más adecuado para la transmisión de voz.

En GSM, cuando se realiza una llamada se asigna un canal de comunicación al usuario, que permanecerá asignado aunque no se envíen datos. En GPRS los canales de comunicación se comparten entre los distintos usuarios dinámicamente, de modo que un usuario sólo tiene asignado un canal cuando realmente se están transmitiendo datos. Para utilizar GPRS se precisa un teléfono que soporte esta tecnología. La mayoría de estos terminales soportarán también GSM, por lo que podrá realizar llamadas de voz utilizando la red GSM de modo habitual y sus llamadas de datos (conexión a Internet, WAP, etc.) tanto con GSM como con GPRS.

La tecnología GPRS o 2.5G, representa un paso más hacia los sistemas inalámbricos de 3G o UMTS. Su principal característica radica en la posibilidad de disponer de un terminal permanentemente conectado, tarificando únicamente por el volumen de datos transferidos (enviados y recibidos) y no por el tiempo de conexión.

Tradicionalmente la transmisión de datos inalámbrica se ha venido realizando utilizando un canal dedicado GSM a una velocidad máxima de 9.6 Kbps. Con el GPRS no sólo la velocidad de transmisión de datos

aumenta de 40 Kbps a un máximo de 115 Kbps por comunicación, sino que además la tecnología utilizada permite compartir cada canal por varios usuarios, mejorando así la eficiencia en la utilización de los recursos de red. La tecnología GPRS permite proporcionar servicios de transmisión de datos de una forma más eficiente. GPRS es una evolución de la actual red GSM, no conlleva grandes inversiones y reutiliza parte de las infraestructuras actuales de GSM. Por este motivo, GPRS tiene desde sus inicios la misma cobertura que la actual red GSM y subsana las deficiencias de ésta. [18]

3.1.2.4.1 Características de GPRS

Entre las características más importantes de la Tecnología GPRS destacan su velocidad de transferencia de hasta 144 Kbps. El aumento en la velocidad de transmisión se debe a que, comparando con GSM sólo se puede tener un canal asignado (un "time slot"), sin embargo, en GPRS, se pueden tener varios canales asignados, tanto en el sentido de transmisión del móvil a la BS como viceversa. La velocidad de transmisión aumentará con el número de canales asignados. Además, GPRS permite el uso de esquemas de codificación de datos que permiten una velocidad de transferencia mayor que en GSM.

Otra característica es su conexión permanente con un tiempo de establecimiento de conexión inferior al segundo. La habilidad de *"Always connected"* la cual significa que un usuario GPRS puede estar conectado todo el tiempo que desee, ya que no hace uso de recursos de red y por tanto no paga mientras no esté recibiendo, ni transmitiendo datos. Los siguientes casos ejemplifican algunos de los tamaños de información que descargaríamos: Envío de un e-mail de 5 líneas de texto con un anexo (documento tipo de Word de 4 páginas), consumiría alrededor de 95 kbytes. Acceder a un buscador, buscar un término (ej. viajes) y recibir una pantalla de respuesta podría ocupar aproximadamente 100 kbytes.

Los terminales GPRS presentan características comunes como una capacidad dual en donde los terminales GPRS están adaptados para aprovechar la cobertura existente GSM para la voz y en GPRS para la transmisión de datos. Para la velocidad de transferencia los terminales GPRS utilizan varios canales simultáneos (slots), en donde el número de canales depende de cada terminal, variando de 1 a 4 para la recepción de datos y de 1 a 2 para el envío. Cada canal representa una velocidad teórica de 13.4 kbits (en GSM sólo 9 Kbits). Además, la tarjeta SIM es la misma que para GSM, por tanto no es necesario cambiar de tarjeta para usar GPRS.

El principal componente de una Red GPRS es el GSN (GPRS Support Node) que recibe el paquete de datos y lo transfiere a Internet u otras redes GPRS. Para suministrar los servicios GPRS sobre GSM, los operadores de red necesitaron agregar unos pocos GSN y hacer una mejora de software a los BSC. [18]

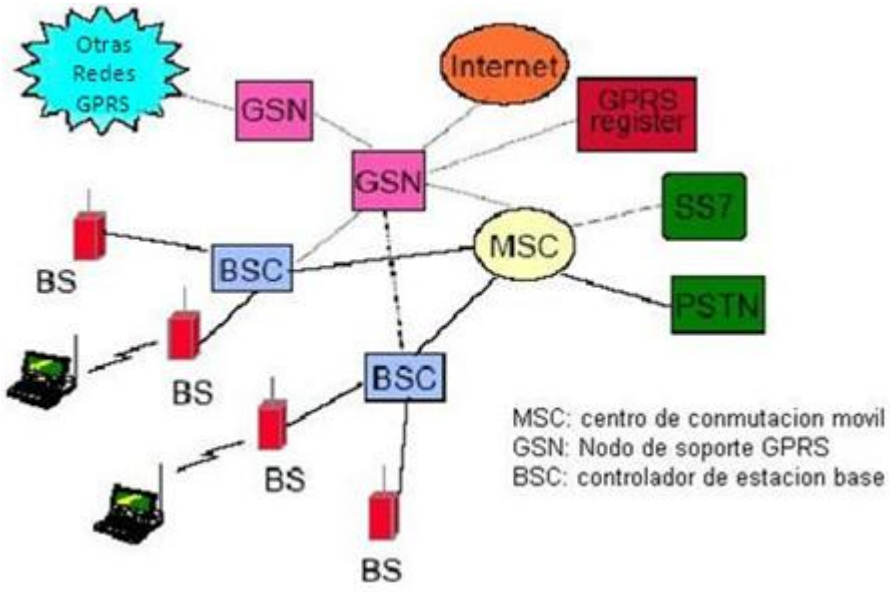


Figura 3.5 Arquitectura de Red GPRS.

3.1.2.5 Red EDGE

La fase después de GPRS es llamada EDGE, generalmente considerada una tecnología 2.75G que introduce nuevos métodos en la capa física, incluyendo una nueva forma de modulación (8 PSK) y diferentes formas de codificar información para proteger contra errores, pero las capas superiores son básicamente las mismas. De manera que EDGE puede generar velocidades máximas de información arriba de 500 Kbps usando la misma infraestructura GPRS.

Mientras que GPRS es basado en una técnica de modulación conocida como GMSK (Gaussian minimum-shift keying), EDGE es basado en modulación 8-PSK que permite una elevación en la velocidad de transmisión a través de la interfaz de aire.

La versión 3G de GSM es WCDMA y está basada en la tecnología CDMA. Esta versión se desvía de los estándares Americanos, aunque usa el mismo principio de espectro ensanchado. [18]

3.2 La plataforma Android

Android constituye una pila de software pensada especialmente para dispositivos móviles, que incluye un sistema operativo como *middleware* y diversas aplicaciones de usuario.

Todas las aplicaciones para Android se programan en lenguaje Java y son ejecutadas en una máquina virtual especialmente diseñada para esta plataforma, que ha sido bautizada con el nombre de Dalvik. El núcleo de Android está basado en Linux 2.6.

La licencia de distribución elegida para Android ha sido Apache 2.0, lo que lo convierte en software de libre distribución. A los desarrolladores se les proporciona de forma gratuita un SDK y la opción de un *plug-in* para el entorno de desarrollo Eclipse, así como un emulador integrado para su ejecución.

El proyecto Android está dirigido por Google y otras empresas tecnológicas agrupadas bajo el nombre de *Open Handset Alliance* (OHA). El objetivo principal de esta alianza empresarial (que incluye a fabricantes de dispositivos y operadores, con firmas como Samsung, LG, Telefónica, Intel, Texas Instruments, entre muchas otras) es el desarrollo de estándares abiertos para la telefonía móvil como medida para incentivar su desarrollo y para mejorar la experiencia del usuario.

Con Android se busca reunir en una misma plataforma todos los elementos necesarios que permitan al desarrollador controlar y aprovechar al máximo cualquier funcionalidad ofrecida por un dispositivo móvil (llamadas, mensajes de texto, cámara, agenda de contactos, conexión Wi-Fi, Bluetooth, videojuegos, etc.), así como crear aplicaciones que sean verdaderamente portables, reutilizables y de rápido desarrollo. En otras palabras, Android quiere mejorar y estandarizar el desarrollo de aplicaciones para cualquier dispositivo móvil.

Android cuenta con su propia máquina virtual, Dalvik, que interpreta y ejecuta código escrito en Java, permite la representación de gráficos 2D y 3D, posibilita el uso de bases de datos, soporta un elevado número de formatos multimedia, servicio de localización GSM, controla los diferentes elementos hardware como Bluetooth, Wi-Fi, cámara, GPS, entre otros. Siempre y cuando el dispositivo móvil lo contemple. También ofrece un *plug-in* para uno de los entornos de desarrollo más populares, Eclipse, y un emulador integrado para ejecutar las aplicaciones. {6}

3.2.1 Arquitectura empleada en Android

La figura 3.6 muestra una visión global por capas de la arquitectura empleada en Android. Cada una de estas capas utiliza servicios ofrecidos por las anteriores y ofrece a su vez los suyos propios a las capas de niveles superiores.

Figura 3.6 Arquitectura empleada en Android.

La capa más inmediata es el núcleo de Android, el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los dispositivos móviles. Esta capa contiene los drivers necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes. Siempre que un fabricante incluya un nuevo elemento de hardware, lo primero que se debe realizar para que pueda ser utilizado desde Android es crear las librerías de control o drivers necesarios dentro del kernel de Linux embebido en el propio Android. {6}

La elección de Linux 2.6 fue debido a:

- Su naturaleza de código abierto y libre se ajusta al tipo de distribución que se buscaba para Android.

- Este kernel de Linux incluye numerosos drivers, además de contemplar la gestión de memoria, gestión de procesos, módulos de seguridad, comunicación en red y muchas otras responsabilidades propias de un sistema operativo.

La siguiente capa corresponde a las librerías utilizadas por Android. Éstas han sido escritas utilizando C/C++ y proporcionan a Android la mayor parte de sus capacidades y características. Junto al núcleo basado en Linux, estas librerías constituyen el corazón de Android. {6}

Entre las librerías más importantes se pueden mencionar las siguientes:

- Librería *libc.* Incluye todas las cabeceras y funciones según el estándar del lenguaje C. Asimismo, las demás librerías se definen en este lenguaje.
- Librería *Surface Manager.* Es la encargada de componer los diferentes elementos de navegación de pantalla y gestiona las ventanas pertenecientes a las aplicaciones activas en cada momento.

- *OpenGL/SL y SGL.* Representan las librerías gráficas y, por tanto, sustentan la capacidad gráfica de Android. *OpenGL/SL* maneja gráficos en 3D y permite utilizar, en caso de que esté disponible en el propio dispositivo móvil, el hardware encargado de proporcionar gráficos 3D. Por otro lado, *SGL* proporciona gráficos en 2D, por lo que es la librería comúnmente utilizada por la mayoría de las aplicaciones.
- Librería *Media Libraries.* Proporciona todos los códecs necesarios para el contenido multimedia soportado en Android (video, audio, imágenes, etc.).
- Librería *SSL (Secure Sockets Layer).* Posibilita la utilización de dicho protocolo para establecer comunicaciones seguras.
- Librería *SQLite*. A través de ésta, Android ofrece la creación y gestión de bases de Datos.
- Librería *WebKit.* Proporciona un motor para las aplicaciones de tipo navegador y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.

Al mismo nivel que las librerías de Android, se sitúa el entorno de ejecución. Éste lo constituyen las *Core Libraries*, que son librerías con multitud de clases de Java y la máquina virtual Dalvik.

Los dos últimos niveles de la arquitectura de Android están escritos en Java. El framework de aplicaciones representa el conjunto de herramientas de desarrollo para cualquier aplicación. Toda aplicación que se desarrolle

para Android, ya sean las propias del dispositivo, las desarrolladas por Google o incluso las que el propio usuario cree, utilizan el mismo conjunto de API y el mismo *framework*, representado por este nivel.

Entre las API más importantes ubicadas aquí, se pueden encontrar las siguientes:

- *Activity Manager*. Este es un importante conjunto de API que gestiona el ciclo de vida de las aplicaciones en Android (del que se hablará más adelante).
- *Window Manager*. Gestiona las ventanas de las aplicaciones y utiliza la librería ya vista *Surface Manager*.
- *Telephone Manager*. Incluye todas las API vinculadas a las funcionalidades propias del teléfono (llamadas, mensajes, etc.)
- *Content Providers*. Permite a cualquier aplicación compartir sus datos con las demás aplicaciones de Android. Por ejemplo, la información de contactos, agenda, mensajes, etc., será accesible para otras aplicaciones.
- *View System*. Proporciona un gran número de elementos para poder construir interfaces de usuario (GUI), como listas, mosaicos, botones, *check-boxes*, tamaño de ventanas, control de las interfaces mediante tacto o teclado, etc.
- *Notification Manager*. Mediante el cual las aplicaciones, usando un mismo formato, comunican al usuario eventos que ocurran durante su ejecución, ya sea una llamada entrante, un mensaje recibido, conexión Wi-Fi disponible, ubicación en un punto determinado, etc. Si llevan asociada alguna acción, en Android denominada *Intent*, (por ejemplo, atender una llamada recibida) ésta se activa mediante un simple clic.

El último nivel de la arquitectura de Android son las aplicaciones. Éste nivel incorpora tanto las incluidas por defecto de Android como aquellas que el usuario vaya añadiendo posteriormente. Todas estas aplicaciones utilizan los servicios, las API y librerías de los niveles anteriores. {6}

3.2.2 Componentes de una Aplicación en Android

Todas las aplicaciones en Android pueden descomponerse en cuatro tipos de componentes principales. Cada aplicación será una combinación de uno o más de estos componentes, que deberán ser declarados de forma explícita en un fichero con formato XML denominado “AndroidManifest.xml”, junto a otros datos asociados como valores globales, clases que implementa, datos que puede manejar, permisos, etc. Este fichero es básico en cualquier aplicación de Android y permite al sistema desplegar y ejecutar correctamente la aplicación.

A continuación se exponen los cuatro tipos de componentes en los que puede dividirse una aplicación para Android. {6}

3.2.2.1 Activity

Es el componente más habitual de las aplicaciones para Android. Un componente *Activity* refleja una determinada actividad llevada a cabo por una aplicación y que lleva asociada típicamente una ventana o interfaz de usuario. Es importante señalar que no contempla únicamente el aspecto gráfico, sino que éste forma parte del componente Activity a través de vistas representadas por clases como *View* y sus derivadas. Este componente se implementa mediante la clase de mismo nombre *Activity*.

La mayoría de las aplicaciones permiten la ejecución de varias acciones a través de la existencia de una o más pantallas. Por ejemplo, en una aplicación de mensajes de texto, en ella, la lista de contactos se muestra en una ventana, mediante el despliegue de una segunda, el usuario puede escribir el mensaje al contacto elegido y en otra tercera puede repasar su historial de mensajes enviados o recibidos. Cada una de estas ventanas debería estar representada a través de un componente Activity, de forma que navegar de una ventana a otra implica lanzar una actividad o dormir otra. Android permite controlar por completo el ciclo de vida de los componentes Activity.

Muy vinculado a este componente se encuentran los Intents. Un Intent consiste en la voluntad de realizar alguna acción, generalmente asociada a unos datos. Lanzando un Intent, una aplicación puede delegar el trabajo en otra, de forma que el sistema se encarga de buscar qué aplicación entre las instaladas es la que puede llevar a cabo la acción solicitada. Por ejemplo, abrir una URL en algún navegador Web, o escribir un correo electrónico desde algún cliente de correo.

3.2.2.2 Broadcast Intent Receiver

Un componente *Broadcast Intent Receiver* se utiliza para lanzar alguna ejecución dentro de la aplicación actual cuando un determinado evento se produzca (generalmente, abrir un componente *Activity*). Por ejemplo, una llamada entrante o un SMS recibido no tiene interfaz de usuario asociada, pero puede utilizar el API Notification Manager, mencionada anteriormente, para avisar al usuario del evento producido a través de la barra de estado del dispositivo móvil. Este componente se implementa a través de una clase de nombre BroadcastReceiver. Para que Broadcast Intent Receiver funcione, no es necesario que la aplicación en cuestión sea la

aplicación activa en el momento de producirse el evento, el sistema lanzará la aplicación si es necesario cuando el evento monitorizado tenga lugar.

3.2.2.3 Service

Service representa una aplicación ejecutada sin interfaz de usuario, y que generalmente tiene lugar en segundo plano mientras otras aplicaciones (con interfaz) son las que están activas en la pantalla del dispositivo.

Un ejemplo de este componente es un reproductor de música. La interfaz del reproductor muestra al usuario las distintas canciones disponibles, así como los típicos botones de reproducción, pausa, volumen, etc. En el momento en el que el usuario reproduce una canción, ésta se escucha mientras se siguen visionando todas las acciones anteriores, e incluso puede ejecutar una aplicación distinta sin que la música deje de sonar. La interfaz de usuario del reproductor sería un componente *Activity*, pero la música en reproducción sería un componente *Service* porque se ejecuta en *background*. Este elemento está implementado por la clase de mismo nombre Service.

3.2.2.4 Content Provider

Con el componente Content Provider, cualquier aplicación en Android puede almacenar datos en un fichero, en una base de datos SQLite o en cualquier otro formato que considere. Además, estos datos pueden ser compartidos entre distintas aplicaciones. Una clase que implemente el componente *Content Provider* contendrá una serie de métodos que permite almacenar, recuperar, actualizar y compartir los datos de una aplicación.

3.2.3 Ciclo de vida de una aplicación en Android

En Android, cada aplicación se ejecuta en su propio proceso. Esto aporta beneficios en cuestiones básicas como seguridad, gestión de memoria, o la ocupación de la CPU del dispositivo móvil. Android se ocupa de lanzar y parar todos estos procesos, gestionar su ejecución y decidir qué hacer en función de los recursos disponibles y de las órdenes dadas por el usuario.

Android lanza tantos procesos como permitan los recursos del dispositivo. Cada proceso, correspondiente a una aplicación estará formado por una o varias actividades independientes (componentes *Activity*) de esa aplicación. Cuando el usuario navega de una actividad a otra, o abre una nueva aplicación, el sistema duerme dicho proceso y realiza una copia de su estado para poder recuperarlo más tarde. El proceso y la actividad siguen existiendo en el sistema, pero están dormidos y su estado ha sido guardado.

Es entonces cuando crea, o despierta si ya existe, el proceso para la aplicación que debe ser lanzada, asumiendo que existan recursos para ello.

Cada uno de los componentes básicos de Android tiene un ciclo de vida bien definido; esto implica que el desarrollador puede controlar en cada momento en qué estado se encuentra dicho componente. El componente *Activity*, probablemente el más importante, tiene un ciclo de vida como el mostrado en la Figura 3.7. {6}

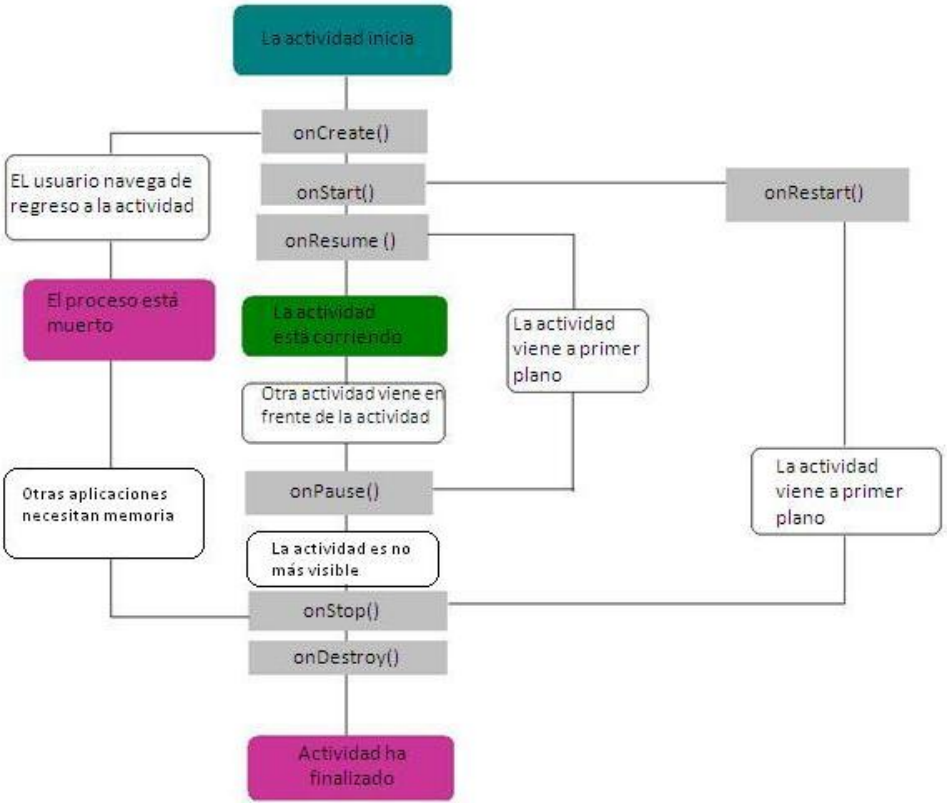


Figura 3.7 Ciclo de vida del componente Activity.

- **onCreate()**, **onDestroy()**. Abarcan todo el ciclo de vida y cada uno de estos métodos representan el principio y el fin de la actividad.
- **onStart()**, **onStop().** Representan la parte visible del ciclo de vida, desde onStart() hasta onStop(), la actividad será visible para el usuario, aunque es posible que no tenga el foco de acción por existir otras actividades superpuestas con las que el usuario está interactuando. Pueden ser llamados múltiples veces.
- **onResume()**, **onPause()**. Delimitan la parte útil del ciclo de vida, desde onResume() hasta onPause(), la actividad no sólo es visible,

sino que además tiene el foco de la acción y el usuario puede interactuar con ella.

Como se ve en la Figura 3.7, el proceso que mantiene a esta *Activity* puede ser eliminado cuando se encuentra en onPause() o en onStop(), es decir, cuando no tiene el foco de la aplicación. Android nunca elimina procesos con los que el usuario está interactuando en ese momento. Una vez que se elimina el proceso, el usuario desconoce dicha situación y puede incluso volver atrás y querer usarlo de nuevo, entonces el proceso se restaura gracias a una copia y vuelve a estar activo como si no hubiera sido eliminado. Además, la *Activity* puede haber estado en segundo plano de manera invisible y entonces es despertada pasando por el estado onRestart().

Como los recursos son siempre limitados. En el momento en el que Android detecta que no hay los recursos necesarios para poder lanzar una nueva aplicación, analiza los procesos existentes en ese momento y elimina los procesos que sean menos prioritarios para poder liberar sus recursos.

Cuando el usuario regresa a una actividad que está dormida, el sistema simplemente la despierta. En este caso, no es necesario recuperar el estado guardado porque el proceso todavía existe y mantiene el mismo estado. Sin embargo, cuando el usuario quiere regresar a una aplicación cuyo proceso ya no existe porque se necesitaba liberar sus recursos, Android lo crea de nuevo y utiliza el estado previamente guardado para poder restaurar una copia fresca del mismo. {6}

3.2.4 Seguridad en Android

La mayoría de las medidas de seguridad entre el sistema y las aplicaciones deriva de los estándares de Linux 2.6, cuyo kernel constituye el núcleo principal de Android.

Por defecto, ninguna aplicación tiene permiso para realizar ninguna operación o comportamiento que pueda impactar negativamente en la ejecución de otras aplicaciones o del sistema mismo. Por ejemplo, acciones como leer o escribir ficheros privados del usuario (contactos, teléfonos, etc.), leer o escribir ficheros de otras aplicaciones, acceso de red, habilitación de algún recurso hardware del dispositivo, etc., no están permitidas. La única forma de poder saltar estas restricciones impuestas por Android, es mediante la declaración explícita de un permiso que autorice a llevar a cabo una determinada acción habitualmente prohibida. Para establecer un permiso para una aplicación, es necesario declarar en el manifiesto uno o más elementos <uses-permission> donde se especifica el tipo de permiso que se desea habilitar.

En la clase android.Manifest.permission se especifican todos los posibles permisos que se pueden conceder a una aplicación: utilización de Wi-Fi, Bluetooth, llamadas telefónicas, cámara, Internet, mensajes SMS y MMS, vibrador, etc.

3.2.5 El emulador

El SDK de Android incluye un completo emulador que permite probar y depurar eficientemente las aplicaciones que se van desarrollando. Este emulador representa gráficamente un dispositivo móvil real, con pantalla, teclado y botones de función, así como aplicaciones de usuario preinstaladas.

Puede ser lanzado tanto a través de una ventana de comandos, con el ejecutable de nombre “emulator” de la carpeta “\tools” o, la que probablemente será la mejor, a través del *plug-in* para Eclipse. En este último caso, cada vez que se ejecute una aplicación, se hará automáticamente a través del emulador incluido en el SDK. Las opciones de lanzamiento desde Eclipse se pueden revisar en *Debug/Run -> Target*.

Este emulador simula un procesador ARM y permite tener varias instancias corriendo al mismo tiempo, cada uno con una emulación distinta. {6}

Figura 3.8 Emulador de Android.

3.2.6 Instalación del Entorno para desarrollar Aplicaciones

Se expondrán los pasos necesarios para empezar a desarrollar y a entender aplicaciones para Android. Las instrucciones de instalación descritas se basan en el sistema operativo Linux. {6}

3.2.6.1 SDK de Android

Para descargarlo, basta con visitar la Web de Android y asegurarse de acceder a la última versión publicada. Una vez descargado el SDK, es necesario descomprimirlo.

3.2.6.2 Eclipse

La descarga de Eclipse no es muy diferente al SDK de Android. La Web de Eclipse ofrece multitud de versiones de este entorno de desarrollo según las necesidades del desarrollador. Finalizada la descarga, se debe descomprimir los ficheros y pulsar el ejecutable para abrir la aplicación. La primera vez que se inicie Eclipse, pide al usuario una localización para el *workspace*, donde se ubicarán por defecto todos los proyectos desarrollados.

3.2.6.3 Plug-in de Android

Lo siguiente es instalar un *plug-in* específico de Android para la plataforma Eclipse. Esta herramienta, llamada ADT (Android Development Tools), facilita la creación de proyectos, su implementación, depuración y ejecución, por lo que es recomendable si se quiere trabajar con Android.

Para instalar el *plug-in* ADT en Eclipse, es necesario seguir las siguientes indicaciones:

1. Iniciar Eclipse
2. Seleccionar la pestaña *Help* > *Software Updates*. Se abrirá una nueva ventana llamada *Software Updates and Add-ons*.
3. Pinchar en la pestaña *Available Software* y pulsar el botón *Add Site*.
4. Introducir la siguiente URL y pulsar *OK*: https://dl-ssl.google.com/android/eclipse/
5. Marcar la casilla correspondiente a *Developer Tools* y pulsar el botón *Install*. Se abrirá una nueva ventana.
6. Cerciorarse de que las opciones *Android Developer Tools* y *Android Editors* están marcadas y pulsar el botón *Finish*.

El proceso de instalación dará comienzo. Con el fin de que los cambios tengan efecto, es necesario reiniciar Eclipse.

3.2.6.4 Referenciar el SDK de Android

Tras abrir de nuevo Eclipse, debe indicarse en las preferencias de Eclipse la localización del SDK a utilizar para los proyectos de Android:

1. Seleccionar la pestaña *Window* > *Preferences*, lo que abrirá una nueva ventana.
2. Elegir *Android* en el panel izquierdo.
3. Pulsar el botón *Browse* e indicar la ruta del SDK de Android.
4. Pulsar el botón *Apply* y después *OK*.

3.2.7 Contenido de un proyecto de Android

Al crear un proyecto Android, en el *workspace* de Eclipse se genera una nueva carpeta con el nombre de dicho proyecto, esta carpeta contiene una serie de subcarpetas y de ficheros que constituyen la anatomía completa de un proyecto Android. En el workspace podrás encontrar una jerarquía similar a la mostrada en la siguiente Figura 3.9:

Figura 3.9 "Workspace" de Eclipse.

Las carpetas y ficheros de "HolaMundo" han sido creadas por el *plug-in* ADT, y sus características dependerán siempre del tipo de proyecto que se esté construyendo. Los principales elementos que podemos encontrar se describen a continuación.

3.2.7.1 Carpeta \src

Contiene los archivos fuente .java del proyecto, utiliza la misma jerarquía de carpetas que la indicada por el nombre del paquete que se haya

asignado, por ejemplo, en el caso de la aplicación “HolaMundo” se especificó el nombre de paquete“com.android.hola”, por lo que se podrá encontrar la siguiente jerarquía:

Figura 3.10 Carpeta “src”.

La última carpeta tiene los ficheros fuente en java. El primero de ellos,“HolaMundo.java”, aloja la única clase que se creó para la aplicación; el segundo fichero, “R.java”, es un archivo que siempre se adjunta por defecto a cualquier proyecto Android y que declara una serie de índices o referencias a los recursos externos que se utilizan en el proyecto actual.

3.2.7.2 Carpeta \res

La carpeta “\res” alberga los recursos utilizados en el proyecto. Por recurso se entiende cualquier fichero externo que contenga datos o descripciones referentes a la aplicación y que debe ser compilado junto a los ficheros fuente. Esta compilación permite al recurso ser accedido de forma más rápida y eficiente.

Android contempla muchos tipos de recursos, como XML, JPEG o PNG, entre otros. En la mayoría de los casos, un recurso viene acompañado de un fichero XML que lo describe y/o configura. Por ejemplo, un caso típico de recurso en Android son las traducciones de una aplicación a distintos idiomas. A pesar de ser cadenas de texto, se representan como recurso a través de un documento XML.

Cada uno de los recursos utilizados en una aplicación para Android ha de estar localizado en la carpeta adecuada, en función de su naturaleza. De esta forma, dentro de la carpeta de recursos “\res” se pueden encontrar las siguientes subcarpetas:

- \anim: Ficheros XML que describen una animación para un determinado elemento.

- \drawable: Contiene recursos que pueden ser dibujados en la pantalla, como imágenes.
- \layout: Contiene diseños que pueden ser usadas para construir interfaces.
- \values: Esta carpeta contendrá ficheros XML que declaran valores de diferentes tipos, por ejemplo, cadenas de texto, colores predefinidos, arrays de elementos, dimensiones, estilos, etc.

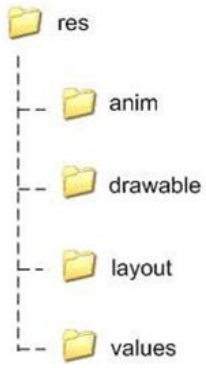


Figura 3.11 Carpeta “res”.

Todos los recursos que aquí se declaren quedarán reflejados en el fichero fuente “R.java”. Este fichero se genera y sincroniza de forma automática en tiempo de compilación según el contenido de la carpeta “\res”.

3.2.7.3 Carpeta \bin

Esta carpeta contiene los archivos binarios del proyecto generados a partir de los archivos fuente, al igual que la carpeta “\src" se mantiene la misma jerarquía de subcarpetas que la representada en el nombre del paquete.

3.2.7.4 Archivo AndroidManifest.xml

Este archivo es generado de forma automática por el *plug-in* de Eclipse, que representa un manifiesto escrito en XML y describe de forma genérica cada uno de los componentes que forman la aplicación (*Activity*, *Broadcast Intent Receiver*, *Service* y *Content Provider*).

Esta descripción abarca aspectos como sus capacidades y requisitos, las clases que los implementan, los datos que pueden manejar o cuándo deben ser lanzados. En este fichero también pueden concretarse permisos y políticas de seguridad que afectan a toda la aplicación.

3.3 Conclusiones

Gracias al estudio de las redes Wi-Fi en su modo infraestructura, en el que el dispositivo móvil se conecta a un AP, se estudió la forma en que estos dispositivos establecen una comunicación. Es decir, el intercambio de los paquetes que el estándar establece, entre estos paquetes están los de Administración que establecen y mantienen la comunicación.

El paquete de Administración de “Solicitud de Asociación” que es enviado por el que inicia la conexión, se encarga de llevar diversa información, entre ellas el SSID del AP al cual nos queremos conectar, si el AP está interesado en esa comunicación, asigna espacio de memoria para ese cliente

.

El paquete de Administración de “Beacon” es enviado periódicamente por los AP para anunciar su presencia y que todas las estaciones que estén en el rango sepan cuales AP están disponibles, estos paquetes contienen varios parámetros entre ellos el SSID del AP y su dirección MAC.

Se explicó también al lector las características básicas del sistema operativo Android, su diseño, arquitectura y funcionamiento, así como una guía de instalación para el entorno de desarrollo Eclipse.

Esta información de las redes Wi-Fi será utilizada por el cliente móvil a la hora de comunicarse y estos paquetes serán los usados en la aplicación de Android para hacer posible la comunicación inalámbrica y poder realizar las comunicaciones de forma eficiente.

CAPÍTULO 4

DISEÑO

En este capítulo se explica el sistema a implementar, el cual consiste en una combinación de hardware y software. El hardware es principalmente una arquitectura de cómputo móvil con conexión a Internet vía Wi-Fi y Red Celular GSM/GPRS. El software tiene como objetivo hacer posible la conexión a Internet por cualquiera de los dos modos Wi-Fi o Red Celular, éste será desarrollado como una Aplicación para el sistema operativo Android.

Por otra parte, se propondrá la integración de Hardware y Software, donde como componente principal para la integración se usará el módulo de evaluación OMAP35x de Texas Instruments, el modulo correrá la aplicación para hacer más eficiente las comunicaciones sobre el sistema operativo Android y contará con interfaces a Internet vía Red Celular y Wi-Fi.

4.1 Diseño de la Arquitectura para dispositivo móvil con conectividad Wi-Fi y GPRS

La arquitectura de cómputo móvil debe contar con módulos de hardware que le permitan lograr una conexión a Internet, además de contar con una pantalla LCD y una ranura de inserción de tarjetas SD/MMC (Secure Digital / Multimedia Card) a través de la cual se le cargará el sistema operativo. A continuación se da una explicación más detallada de la arquitectura y sus requerimientos.

4.1.1 Descripción de la arquitectura

La arquitectura debe de contar necesariamente con los siguientes módulos:

a) Procesador OMAP.
   Es un procesador de Texas Instruments con capacidades para aplicaciones móviles. Este procesador contiene una arquitectura dual-core que consiste de un procesador ARM (Advanced RISC Machines) de propósito general y un DSP (Procesador Digital de señales). [19]

b) Interfaz LCD
   Interfaz LCD es un dispositivo de entrada/salida que nos mostrará los resultados de las aplicaciones que se estén corriendo en el procesador y nos servirá como periférico para introducir datos.

c) Conector SD/ MMC
   Esta interfaz será usado para iniciar el Sistema Operativo desde una tarjeta SD/MMC, esta tarjeta tendrá instalado el Sistema Operativo Android.

d) Interfaz para conexión a un Módem GSM/ GPRS
   El Módem GSM/ GPRS nos permitirá hacer una conexión a Internet vía Red Celular, estará conectado al procesador vía una interfaz UART.

e) Interfaz para conexión Wi-Fi

El módulo Wi-Fi del estándar 802.11 nos permitirá hacer una conexión a un Punto de Acceso para poder conectarnos a Internet. Este módulo estará conectado al procesador vía una interfaz UART.

La figura 4.1 muestra el diagrama de bloques de la arquitectura de cómputo móvil a implementar.

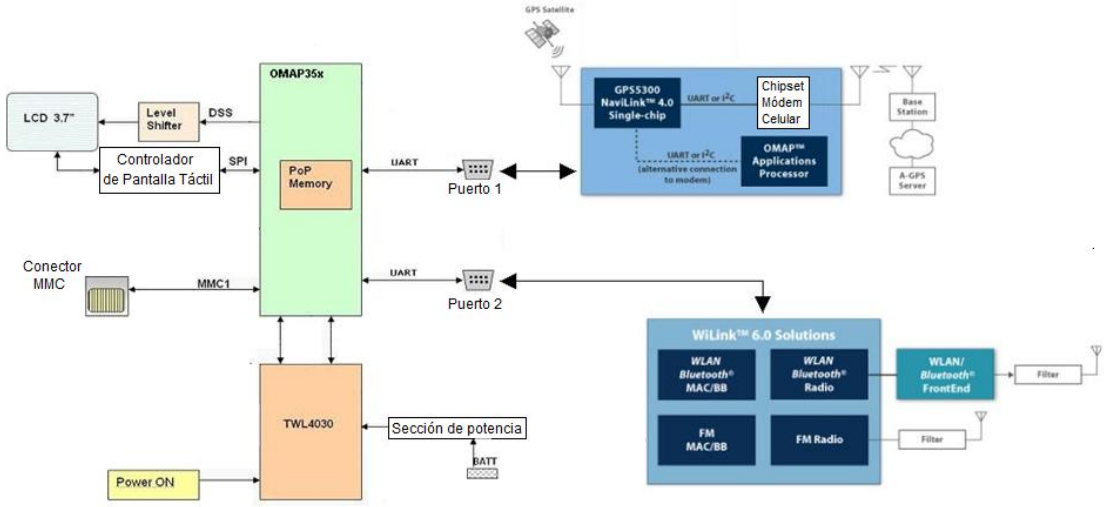


Figura 4.1 Diagrama de bloques de la arquitectura propuesta.

4.2 Módulos para la Integración de Hardware

Como se mencionó anteriormente, para la Integración de Hardware y Software se usa el módulo de evaluación OMAP35x de Texas Instruments, un módem GSM/GPRS de Enfora, que será conectado al módulo de evaluación vía puerto serial, y una tarjeta WL1271, tarjeta hija del OMAP fabricada por la misma empresa, que nos proporcionará la conexión Wi-Fi.

4.2.1 Módulo de evaluación OMAP35x EVM

La figura 4.2 muestra la parte superior del OMAP35x EVM con etiquetas que señalan módulos y conectores que serán usados para experimentación.

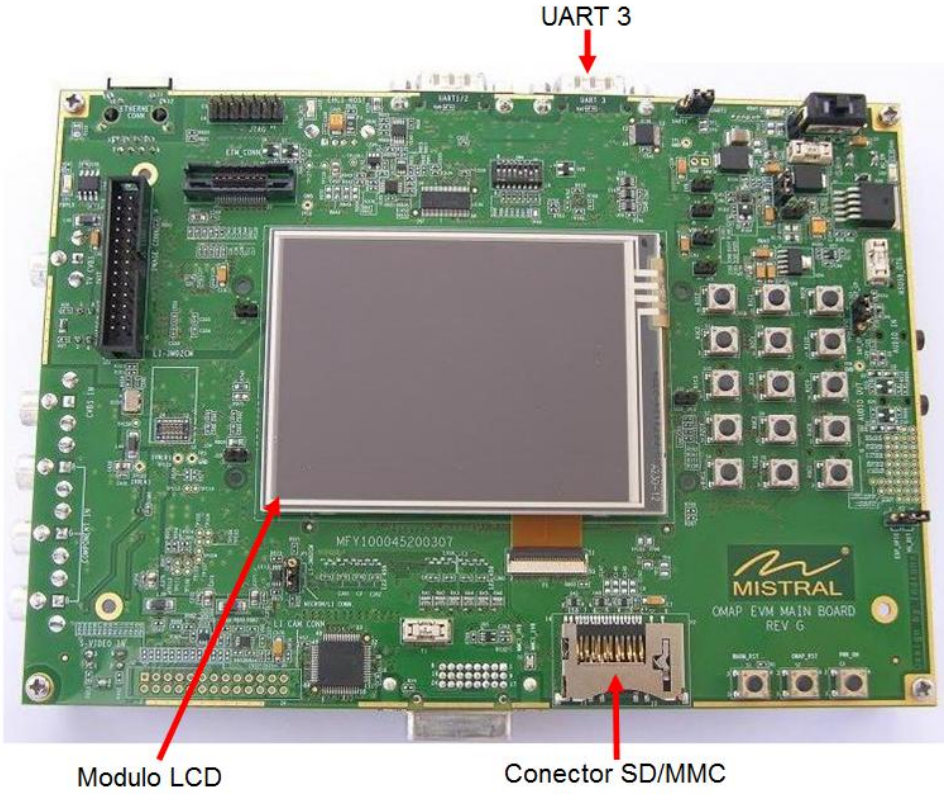


Figura 4.2 Parte superior del OMAP 35x EVM.

- ✓ UART3: Interfaz vía puerto serial en el OMAP35 EVM, el cual será usado para conectar un módem GSM/GPRS.
- ✓ Conector SD/ MMC: Este interfaz será usado para iniciar el Sistema Operativo desde una tarjeta SD que tendrá instalado el Sistema Operativo Android.
- ✓ Módulo LCD: Pantalla LCD de 3.7 pulgadas modo dual (QVGA/ VGA) con soporte Touch Screen.

La figura 4.3 muestra la parte superior del OMAP35x EVM con etiquetas que señalan módulos y conectores que serán usados para experimentación. [19]

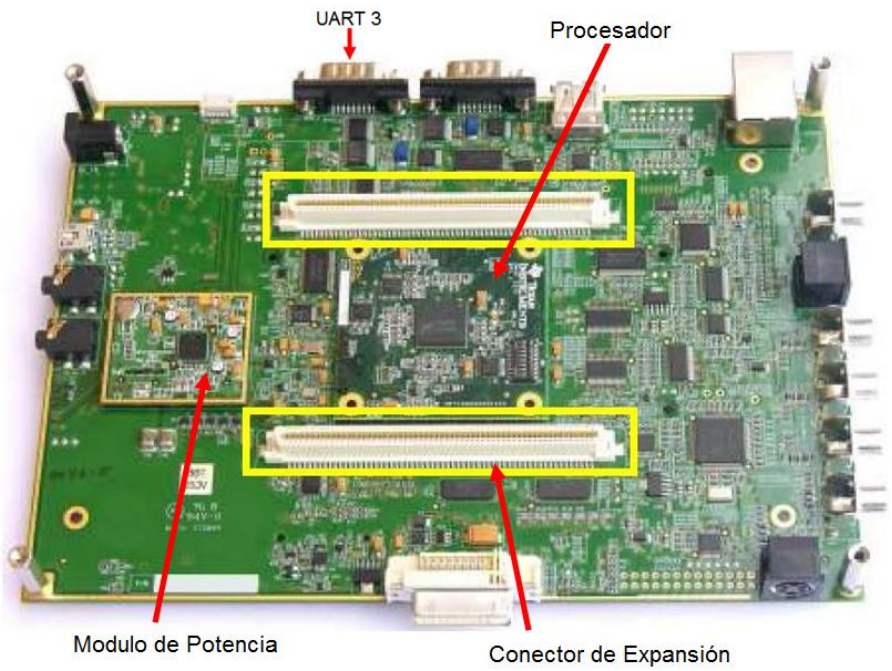


Figura 4.3 Parte inferior del OMAP 35x EVM.

- ✓ Conector de Expansión: Este conector de expansión usa una interfaz UART y será usado para conectar la tarjeta WL1271 que nos dará la conexión a Internet, vía WI-Fi
- ✓ Procesador: El Módulo del procesador consiste de un OMAP 3530.

4.2.2 Tarjeta WL1271

Tarjeta hija del OMAP, fabricada por la misma empresa que el módulo OMAPEVM, que nos proporcionará la conexión Wi-Fi. [20]

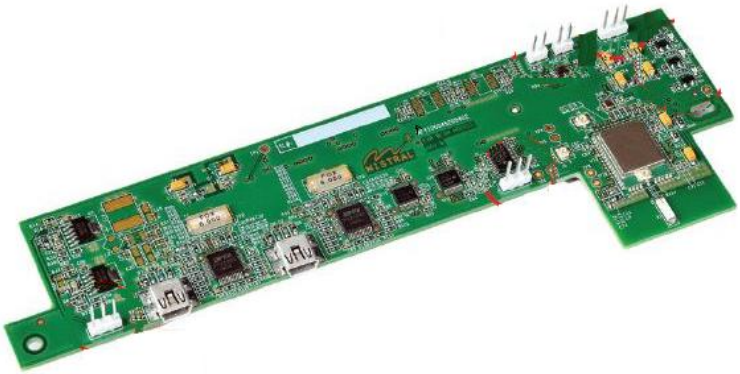


Figura 4.4 Parte superior de la tarjeta hija WL1271.

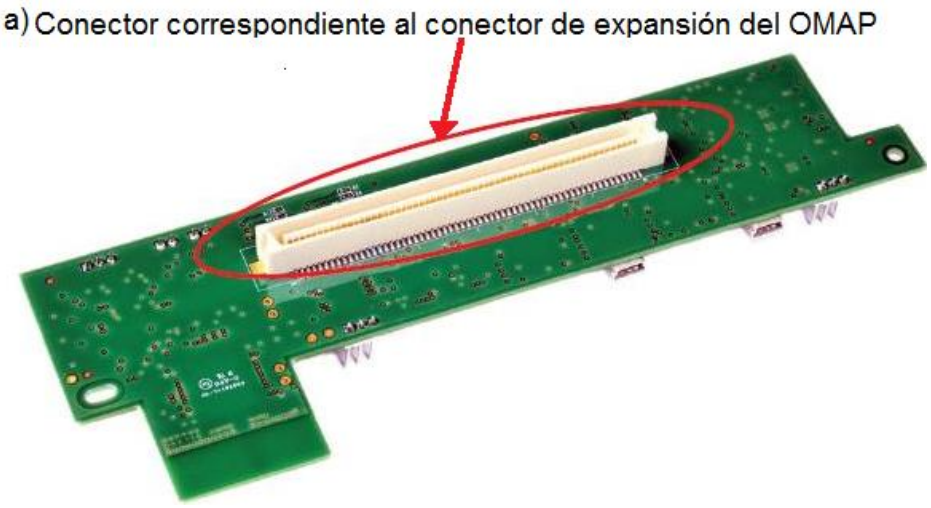


Figura 4.5 Parte inferior de la tarjeta hija WL1271.

a) Conector de la Tarjeta WL1271 que será conectado al conector de expansión de módulo OMAP.

4.2.3 Modem Enfora GSM/GPRS EDG0308-11

Módem GSM/GPRS de Enfora, que será conectado al módulo de evaluación vía puerto serial. [21]

Figura 4.6 Módem GSM/GPRS de Enfora.

- Interfaz SIM: Interfaz para insertar el Módulo de Identidad del Suscriptor
- Antena: usada para enviar o recibir información esta internamente conectada a la señal de RF del módulo GPRS
- Puerto Serie: La interfaz serial es intencionada para la comunicación entre el módulo y el módulo OMAP

4.3 Software para hacer más eficiente las comunicaciones

Como se mencionó anteriormente, se pretende tener siempre una conexión a Internet, dándole prioridad a una red Ethernet Inalámbrica abierta, en caso de que el dispositivo móvil encuentre redes inalámbricas Ethernet con un tipo de seguridad habilitada que le impida tener la conexión, inmediatamente hará la conexión a Internet mediante la Red Celular.

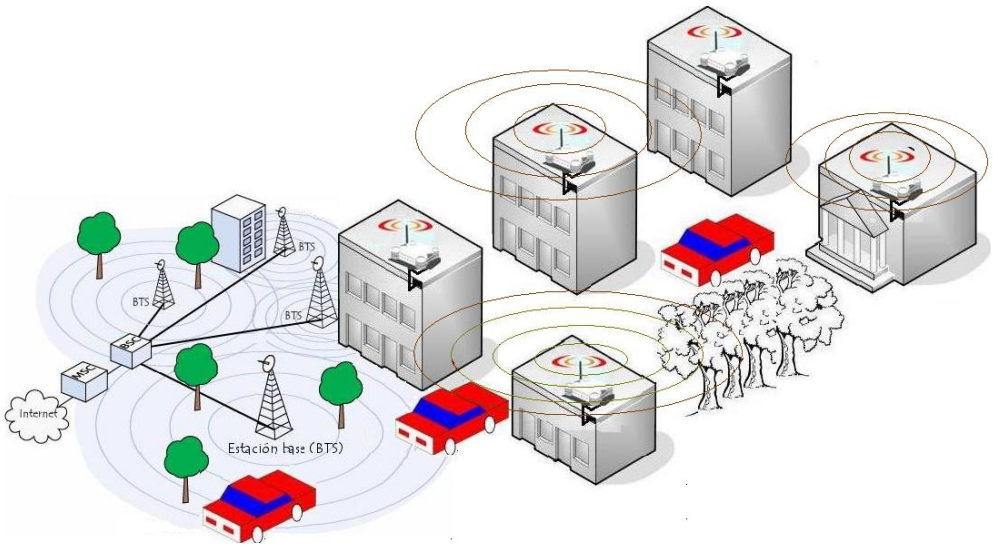


Figura 4.7 Como hacer más eficiente las comunicaciones.

Otro caso a tratar es el siguiente: cuando el dispositivo móvil encuentre más de una red Ethernet inalámbrica sin seguridad habilitada, es decir abierta, el dispositivo móvil elegirá al punto de Acceso que le ofrezca mayor potencia, ya que mientras más potencia le ofrezca al dispositivo móvil, la transferencia de datos entre los dispositivos será más veloz.

El siguiente diagrama de flujo muestra el problema a tratar. En primera instancia el dispositivo debe escanear redes inalámbricas Ethernet a su alcance, en caso de que no las halle, realizará una conexión vía Red celular, en caso contrario, es decir, que detecte redes inalámbricas Ethernet

disponibles, verificará si el dispositivo móvil tienen algún tipo de seguridad habilitada, en caso de que todas las redes escaneadas tengan un tipo de seguridad habilitada, otra vez hará una conexión vía Red celular, pero en caso de que halle redes Ethernet inalámbricas sin seguridad habilitada o abiertas, tomará la que le ofrezca mayor potencia.

Sin embargo, ¿qué pasa si la red que encontró es la misma a la que está conectado? Entonces, el dispositivo móvil no hará ningún tipo de asociación para evitar la pérdida de datos, pero en caso que la red inalámbrica Ethernet sea diferente a la red que está conectado actualmente, hará la conexión a este punto de acceso. En otro caso, si hubiera una conexión GPRS, el dispositivo móvil se desasociará de esta conexión. De esta forma el Dispositivo móvil finaliza el proceso y puede volver a escanear nuevamente redes inalámbricas, buscando siempre las redes Ethernet inalámbricas abiertas y de mayor potencia.

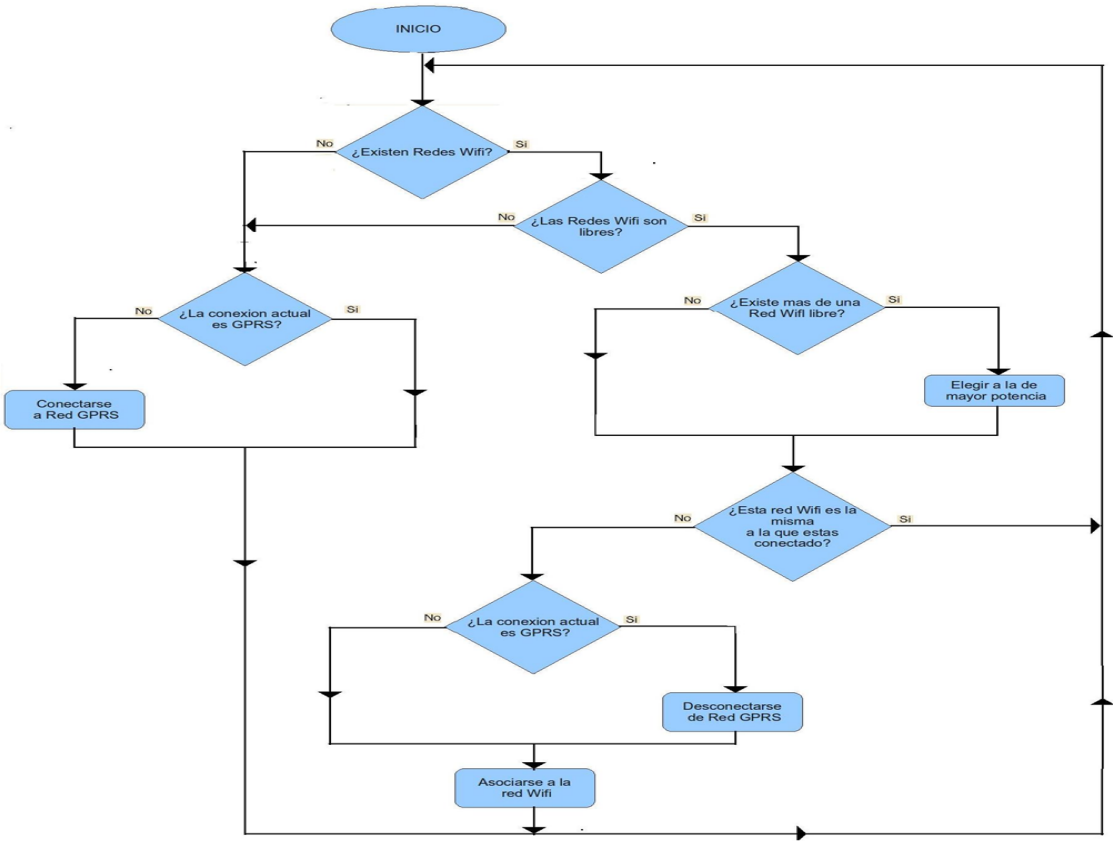


Figura 4.8 Diagrama de flujo que muestra el problema a tratar.

4.3.1 Análisis y diseño de la aplicación para una Comunicación Eficiente

Esta parte del capítulo explica paso a paso el desarrollo completo de la aplicación “ComEficiente”, cuyo objetivo es mantener al dispositivo móvil conectado a Internet vía Red Celular o red Wi-Fi.

4.3.1.1 Introducción a “ComEficiente”

El objetivo básico de “ComEficiente” es mantener al dispositivo móvil conectado a Internet vía Red Celular o red Wi-Fi, habilitando, deshabilitando o haciendo conexiones a redes Wi-Fi. Para ello utiliza fundamentalmente la tarjeta de red Inalámbrica Wi-Fi del propio dispositivo móvil, para saber la existencia de puntos de acceso de redes Inalámbricas Wi-Fi.

La aplicación “ComEficiente” debe conocer las características de cada una de las redes Inalámbricas detectadas, tales como su potencia y tipo de encriptación, pudiendo de este modo elegir la red Wi-FI sin ningún tipo de encriptación y la que le ofrezca más potencia para hacer una conexión a Internet o en caso de la ausencia de al menos una red Wi-Fi libre hacer la conexión a Internet vía Red Celular.

4.3.1.1.1 Casos de uso

En este apartado se utilizan los casos de uso como forma de acercar al lector las funcionalidades que la aplicación “ComEficiente” va a desempeñar frente al usuario. Como recordatorio se dirá que, en ingeniería del software, un caso de uso representa un uso típico que se le da al sistema. La técnica de los casos de uso permite capturar y definir los requisitos que debe cumplir una aplicación, y describe las típicas interacciones que hay entre el usuario y esta. Dicha técnica es utilizada con frecuencia por los ingenieros del software para mostrar al cliente de forma clara y sencilla qué tipo de acciones podrá realizar su futuro sistema, entre otras cosas.A continuación se muestra un diagrama con los casos de uso asociados a “ComEficiente”.

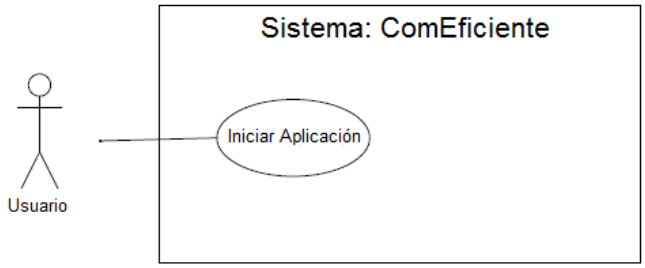


Figura 4.9 Caso de uso asociado a el usuario.

En el diagrama mostrado en la Figura 4.9, la aplicación "*ComEficiente"* está representado por una caja que contiene los casos de uso, en este caso sólo uno. Cada caso de uso consiste en un óvalo con un nombre descriptivo en su interior. Fuera del sistema se encuentra los actores que pueden interactuar con él, en este caso, existe un único actor de nombre "usuario" que es el que realiza el caso de uso.

El cometido del caso de uso mostrado en el diagrama "Iniciar Aplicación" es correr la aplicación "ComEficiente".

Siguiendo el mismo criterio, el siguiente diagrama expresa los casos de uso asociados a la aplicación para gestionar la información de conexión de red:

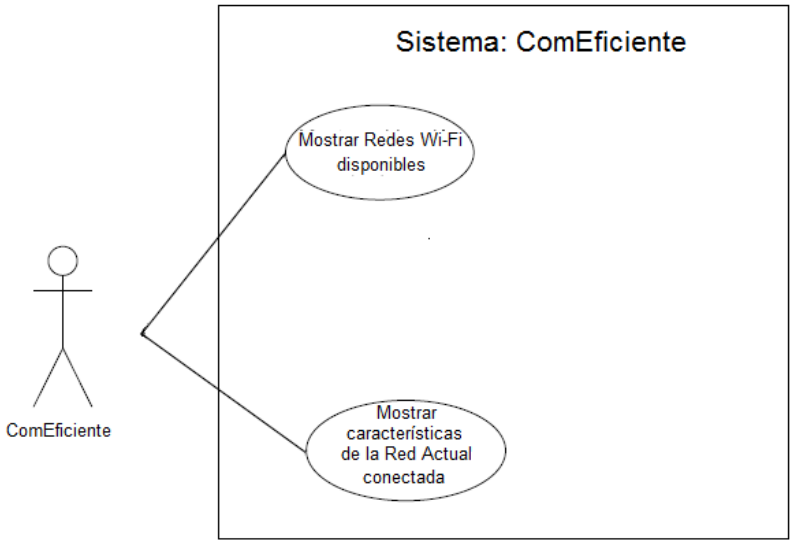


Figura 4.10 Caso de uso asociado a la aplicación.

En el diagrama de la Figura 4.10, el único actor que realiza los casos de uso es la propia aplicación *"ComEficiente"*, ya que es la que establece la conexión con las redes. Los casos de uso considerados son:

- Mostrar Redes Wi-Fi disponibles: Desplegar todas las redes Wi-Fi disponibles.
- Mostrar características de la Red Actual conectada: Obtener los datos de la red actual conectada, ya sea Red celular o Red Wi-Fi, como potencia, velocidad y encriptación.

Es importante recalcar que los casos de uso simplemente expresan el punto de vista del usuario sobre cómo debe funcionar la aplicación, qué

puede realizar a través de ella y también como una forma de facilitar su comprensión. No tiene por qué existir ninguna correspondencia entre los casos de uso y las clases finalmente implementadas, más allá de que las clases en su conjunto, como sistema completo, realizan aquello que los casos de uso expresan.

4.3.1.1.2 Modelo de clases

Las clases utilizadas para resolver un determinado problema, sus atributos, métodos y la visibilidad que de estos tienen las demás clases, así como las relaciones que existen entre ellas y sus colaboraciones, constituyen el modelo de clases. Mediante este tipo de modelos se expresa, con mayor detalle, la futura implementación del sistema y permite dar una idea bastante acertada a la forma en la que se ha abordado el problema.

En el presente apartado se ofrece al lector una breve descripción de cuál es el modelo de clases utilizado en la aplicación “ComEficiente”. Nuevamente, el objetivo principal es ayudar a comprender cómo se soluciona el problema planteado y pretende servir de anticipo a la explicación más detallada de la implementación de ComEficiente, que se da más adelante. Por todo ello, el modelo de clases ofrecido no incluye más que clases, atributos, métodos, y la visibilidad de éstos, con vistas a no perder al lector con aspectos demasiado complejos.

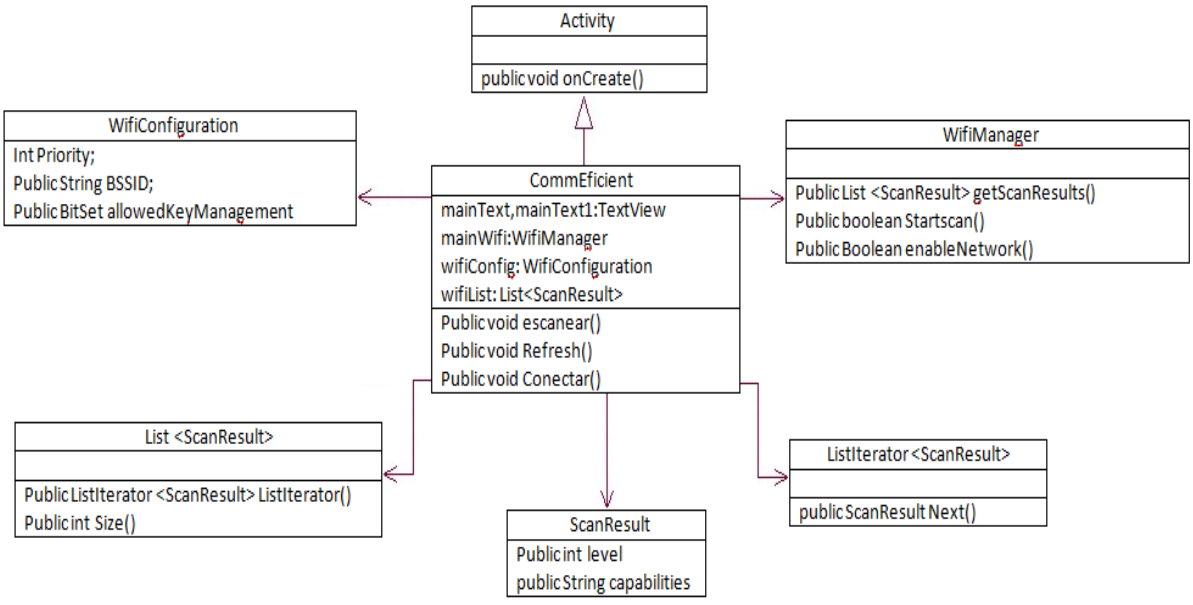


Figura 4.11 Modelo de clases.

Clase	Descripción
Activity	Clase que refleja una determinada actividad llevada a cabo por una aplicación, y que lleva asociada típicamente una ventana o interfaz de usuario.
WifiConfiguration	Clase representando una red Wi-Fi configurada, incluyendo configuración de seguridad.
List<E>	Un List es una colección la cual mantiene un ordenamiento para sus elementos.
ScanResult	Clase que describe información acerca de un Punto de Acceso Detectado.
WifiManager	Esta clase suministra la API principal para administrar todos los aspectos de conectividad Wi-Fi.
ComEfiiente	Clase principal que hereda métodos y atributos de la clase Activity, aquí se hace la instancia de objetos de las demás clases agregadas.

Tabla 4.5 Clases usadas en el diseño.

4.3.1.1.3 Arquitectura

Una vez descrito el funcionamiento general de *"ComEficiente"* y las funcionalidades ofrecidas al usuario, a continuación se ofrece un diagrama con la arquitectura general del sistema completo:

- ✓ **Dispositivo móvil**: Módulo OMAP con el sistema operativo Android corriendo y donde está instalada la aplicación "ComEficiente". A través de su interfaz Wi-Fi escanea Puntos de Acceso libres y de mayor potencia que le ofrezcan una conexión a Internet y por medio de su interfaz GSM/ GPRS hará una conexión a Internet vía Red Celular cuando no haya disponible ningún Punto de Acceso libre.

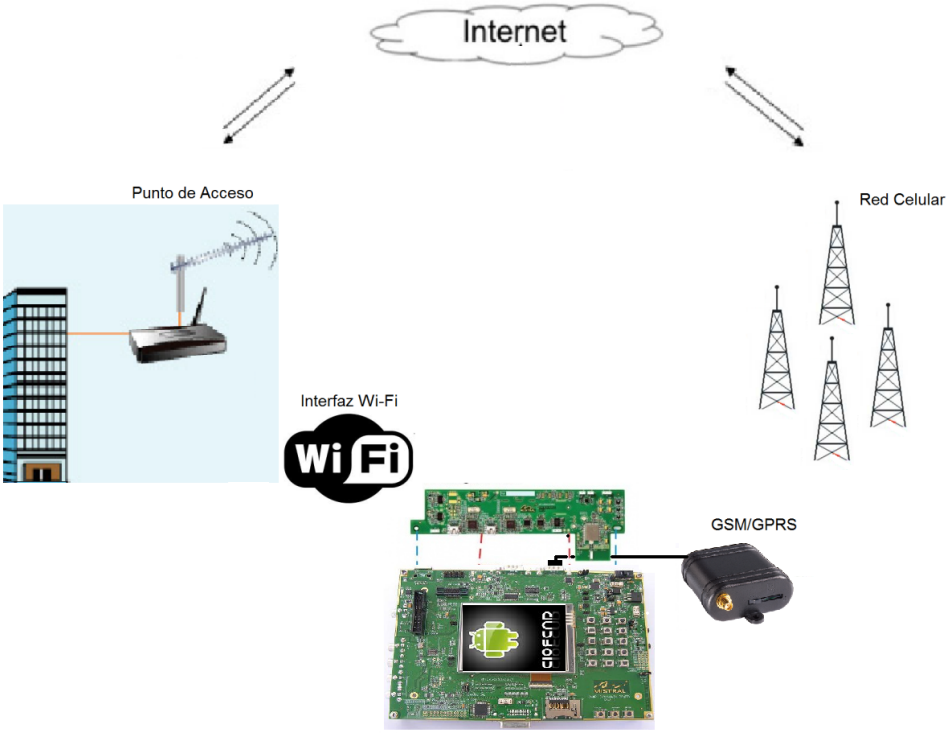


Figura 4.12 Arquitectura general del sistema completo.

4.4 Conclusiones

Se propuso una arquitectura de computo móvil, la cual utiliza como componente principal el procesador OMAP con interfaz para LCD, interfaz para SD/ MMC, interfaz para conexión a un Módem GSM/ GPRS e interfaz para conexión Wi-Fi. A través de la interfaz SD/ MMC se instaló el sistema operativo Android. También se desarrolló una aplicación para eficientar las comunicaciones, la cual es llamada “ComEficente”, a través de un diagrama de flujo se explica el problema a resolver, se muestran los casos de uso de la aplicación, así como también el diagrama de clases.

Para la Integración de Hardware y Software se usará el módulo de evaluación OMAP35x de Texas Instruments, el cual correrá la aplicación sobre el sistema operativo Android que estará montado sobre el procesador OMAP. El módulo de evaluación contará con un módem GSM/GPRS de Enfora, que será conectado al módulo de evaluación vía puerto serial, y una tarjeta WL1271, tarjeta hija del OMAP fabricada por la misma empresa, que nos proporcionará la conexión Wi-Fi.

CAPÍTULO 5

PRUEBAS Y RESULTADOS

Se desarrollará la arquitectura de cómputo móvil propuesta en el capítulo anterior, hasta la creación del PCB. Se mostrará el desarrollo de una aplicación básica en Android para concluir con la creación de la aplicación “ComEficiente”. Se configurará el módulo de evaluación OMAP35x para que sea capaz de correr un sistema operativo y se instalará el sistema operativo en una tarjeta SD, la cual será insertada en el módulo de evaluación a través de la interfaz SD/MMC, también se hará la instalación del sistema operativo Android sobre un ordenador portátil con procesador ATOM y en la plataforma TAO. Se mostrará la forma de correr aplicaciones de Android tanto en dispositivos virtuales como físicos y se expondrá a detalle la compilación del Kernel para el sistema operativo Android.

5.1 Desarrollo de la arquitectura de cómputo móvil

Esta parte de capitulo explica el desarrollo completo de la arquitectura propuesta hasta llegar a un PCB (Printed Circuit Board). Se pretende desarrollar la arquitectura propuesta en el Capitulo 4, Figura 4.1.

Como recordatorio, la arquitectura debe contar con las siguientes interfaces:

a) Procesador OMAP
b) Interfaz LCD
c) Conector SD/ MMC
d) Interfaz para conexión a un Módem GSM/ GPRS
e) Interfaz para conexión Wi-Fi

El desarrollo de la arquitectura se encuentra dividido:

- En la creación del circuito esquemático.
- En la creación del PCB a partir del circuito esquemático.

El trabajo a realizar será desarrollado con el software para diseño de PCB Altium Designer.

5.1.1 Creación Del PCB

En esta parte se desarrollará el circuito esquemático y a partir de éste se creará el PCB de la arquitectura propuesta. El diseño de cada uno de los componentes esquemáticos así como los *footprints* de cada componente (representación del componente en el PCB) son creados, debido a la ausencia de estos componentes en el software.

5.1.1.1 Conectores para el procesador OMAP

En esta parte de desarrollo se crean los *footprint* conectores donde irá montado el procesador. Los conectores mostrados a continuación son usados para obtener las señales del procesador OMAP. [19]

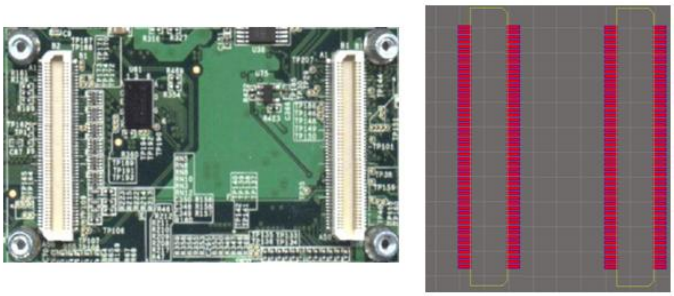


Figura 5.1 Conectores del módulo del procesador OMAP y *footprint*.

El esquemático del módulo del procesador se puede ver en el anexo 1.

5.1.1.2 Conector SD/MMC

En esta parte de desarrollo se crean los conectores y componentes que integran a la interfaz. El esquemático del módulo del conector SD/MMC se puede ver en el anexo 2.

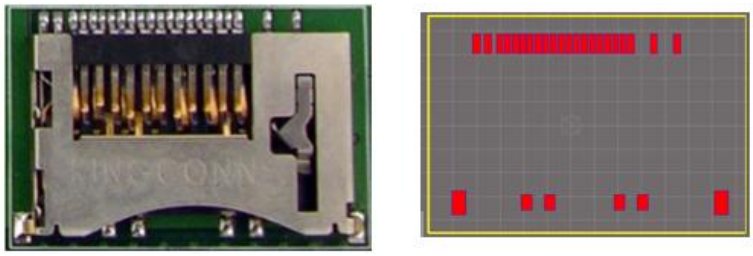


Figura 5.2 Conector SD/MMC y footprint. [19]

5.1.1.3 Interfaz LCD

En esta parte de desarrollo se crean los conectores y componentes que integran a la interfaz LCD.

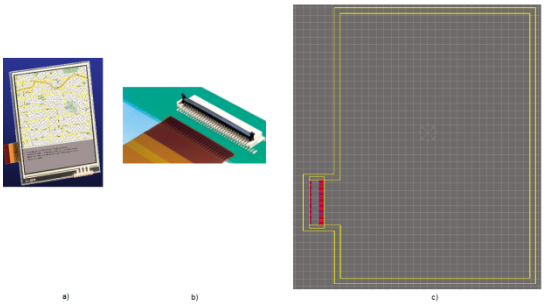


Figura 5.3 Conector para el LCD y su footprint.

El esquemático del módulo del LCD se puede ver en el anexo 3.

5.1.1.4 Distribución de potencia

En esta parte de desarrollo se crea la distribución de potencia que estará presente en el PCB. El esquemático del módulo de distribución de potencia se puede ver en el anexo 4.

5.1.1.5 Conectores del TWL4030 (Modulo Administrador de Potencia**)**

Conectores utilizados para obtener la señal del modulo de potencia. El esquemático del módulo de los conectores TWL4030 se puede ver en el anexo 5.

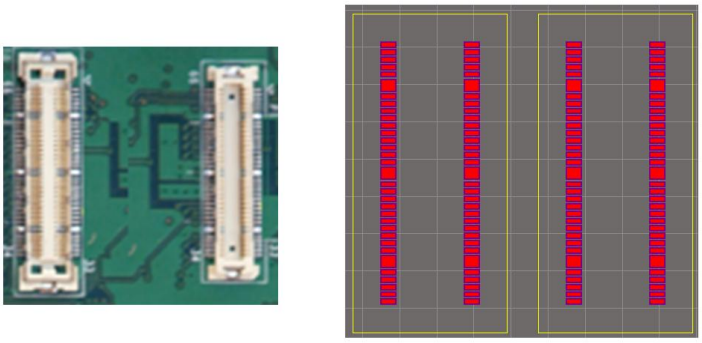


Figura 5.4 Conectores del Módulo TWL4030 y su footprint.

5.1.1.6 Interfaz GSM/GPRS

Interfaz que habilita la conexión a una red GSM/GPRS, el esquemático del módulo GSM/GPRS se puede ver en el anexo 6

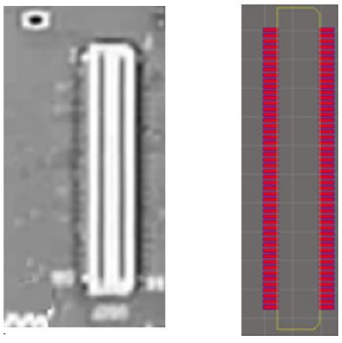


Figura 5.5 Conector del Módem GSM/GPRS y su footprint.

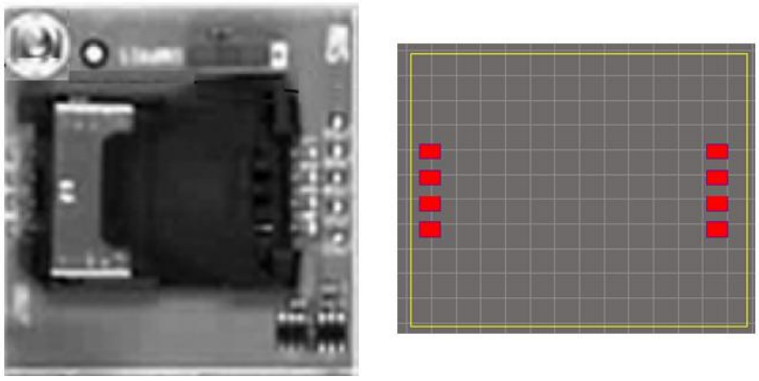


Figura 5.6 Socket del SIM y su footprint.

5.2 Aplicación básica en Android

5.2.1 Creación de un nuevo proyecto en Android

1. Ejecuta Eclipse:

Figura 5.7 Ejecución del Ambiente de desarrollo Eclipse.

2. Desde Eclipse, selecciona File>New>Project y después “Android Project”.

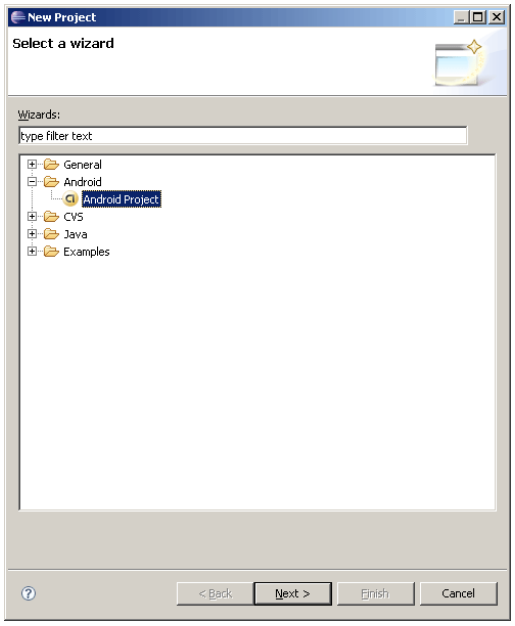


Figura 5.8 Creación de un nuevo proyecto.

3. En la siguiente ventana agrégale los siguientes valores:

   - Project name: HelloAndroid
   - Application name: Hello, Android
   - Package name: com.example.helloandroid
   - Create Activity: HelloAndroid
   - Min SDK Version: 2

Nota: El valor SDK está asociado con una versión del sistema operativo. Como se muestra en la siguiente figura, el valor del SDK=2 está asociado con la versión del sistema operativo Android 1.1, de manera que el valor en "Min SDK=2" significa que la aplicación necesita que el dispositivo cuente con la versión mínima 1.1 de Android para poder ejecutarse.

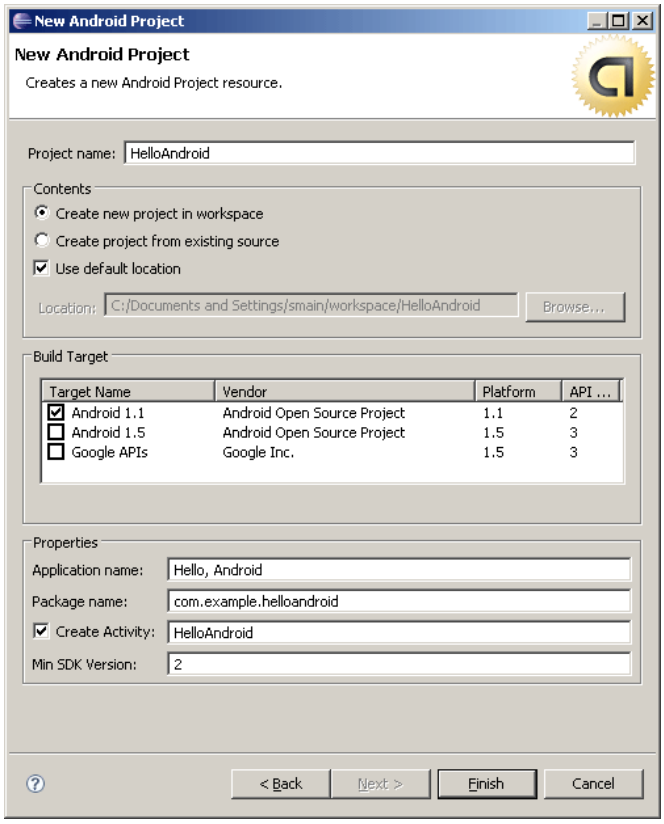


Figura 5.9 Configuración del nuevo proyecto.

4. El proyecto Android está listo y está visible a la izquierda de la pantalla. Abre el archivo "HelloAndroid.java", localizado en "HelloAndroid >src >com.example.helloandroid":

```
package com.android.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       TextView tv = new TextView(this);
       tv.setText("Hello, Android");
       setContentView(tv);
   }
}
```

Código 5.1 Clase principal "HelloAndroid".

Como se dijo en el capítulo 3, un “View” es un objeto dibujable usado como un objeto en la interfaz de usuario como: botón, imagen o texto. Cada uno de estos objetos es una clase de la clase View. La subclase que controla el texto es TextView.

En el código anterior se crea un TextView, y se define el contenido del texto con “setText”. Finalmente se pasa el TextView a setContentView() para mostrarlo como contenido en la interfaz de usuario.

5. El plugin de Eclipse hace sencillo correr las aplicaciones:
    a. Selecciona Run > Run
    b. Selecciona “Android Application”

Tu aplicación aparecerá en el simulador después de unos instantes:

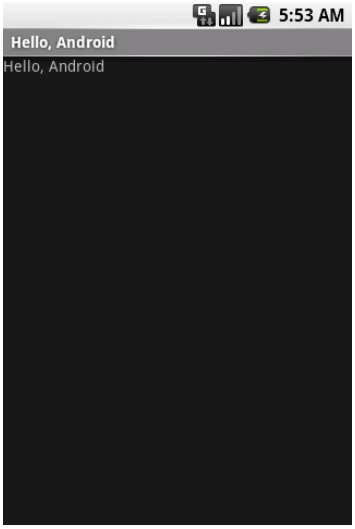


Figura 5.10 Aplicación “Hello, Android” corriendo en el simulador.

En el ejemplo anterior se usó un esquema de interfaz de usuario programática, se construyó la interfaz de usuario directamente con código fuente. Esta declaración del archivo XML está en: /res/layout/main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:text="Hello, Android"/>
```

Código 5.2 Archivo XML creado directamente con código fuente.

El formato del fichero consiste en construir un árbol de elementos, de tal forma que cada elemento corresponda con el nombre de una clase View válida (como TextView).

En el código anterior se puede comprobar como está definido, en primer lugar, el documento XML como un documento válido. A continuación, se indica un único elemento <TextView> que corresponde con la clase TextView utilizada ya anteriormente. {6}

Observamos los siguientes atributos:

- xmlns:android: Declara el espacio de nombres Android válido.
- android:layout_width: Indica qué cantidad del ancho de la pantalla ha de ser ocupado por éste elemento View. En este caso, se declara que la caja de texto ocupe toda la pantalla.
- android:layout_height: Declara qué cantidad del alto de la pantalla va a ocupar el elemento View al que hace referencia. Al igual que antes, en este caso se desea que ocupe todo el alto de la pantalla.
- android:text: Indica el texto que se desea mostrar en este elemento View.

5.2.2 Aplicación básica usando el código XML.

Android suministra un modelo de construcción de interfaz de usuario alternativo basado en código XML, como se puede ver a continuación:

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:text="@string/hello"/>
```

Código 5.3 Código XML alternativo de la aplicación anterior.

Ahora en el campo “android:text” del código anterior se usa una cadena de recurso y no un valor de cadena. La cadena de recurso “@string/hello” definida en res/values/strings.xml:

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello, Android! I am a string resource!</string>
    <string name="app_name">Hello, Android</string>
</resources>
```

Código 5.4 Contenido del archivo "strings.xml".

Abre y modifica tu clase "HelloAndroid" para que use el esquema XML, como sigue:

```java
package com.android.helloandroid;

import android.app.Activity;
import android.os.Bundle;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Código 5.5 Modificación de la clase "HelloAndroid".

Ahora en lugar de pasar un objeto de vista en setContentView(), se da una referencia al esquema de recurso, identificado como "R.layout.main" el cual está definido en "/res/layout/main.xml". Al correr la aplicación nos mostrará la misma interfaz. Después de todo, el punto era mostrar que los dos métodos producen los mismos resultados, pero ahora con un tipo de programación más libre de errores al no estar trabajando directamente con código fuente.

5.3 Instalación del Hardware

Ésta sección preparará el Hardware con el propósito de correr software sobre la plataforma OMAP.

5.3.1 Instalación de un programa terminal

Un programa terminal puerto serial es usado para comunicarse con la consola del puerto serie de la plataforma OMAP. Para los usuarios de

Windows Hyperterminal y de Linux Minicom es el recomendado. En ambos casos los ajustes del puerto serie son los mismos:

Bits por segundo: 115200
Bits de datos: 8
Paridad: Ninguno
Bits de parada: 1
Control de flujo Ninguno

5.3.1.1 Instalación de Minicom

Para crear una nueva configuración para el puerto serie ejecuta el siguiente comando como súper usuario:

minicom –c on

Desde el menú de instalación configura el nombre del puerto serie y parámetros - “Serial port setup” - > “A” para el dispositivo serial (/dev/ttyS0 o /dev/ttyS1) y “E” para la instalación de ”Bps/Par/Bits”, la cual es “115200 8N1” por default.

Una vez listos los ajustes anteriores guarda la configuración. Ahora puedes empezar minicom con la configuración guardada.

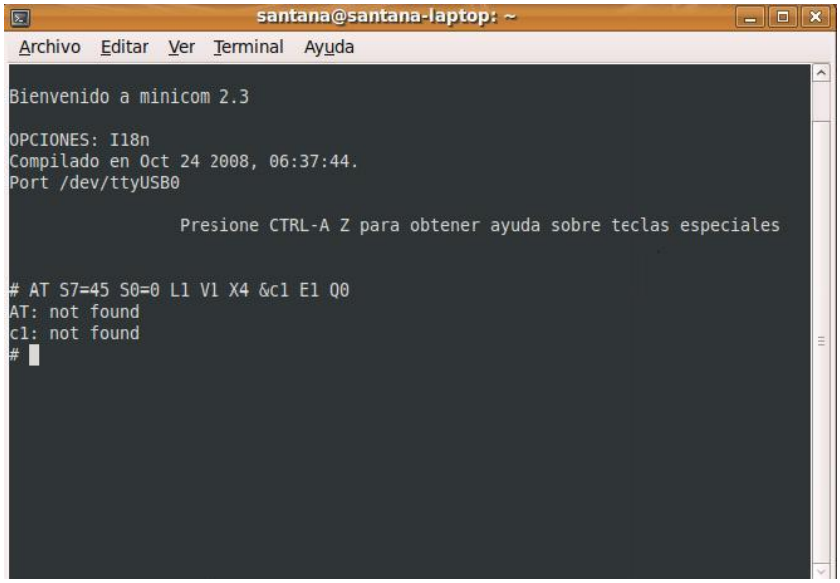


Figura 5.11 Consola corriendo el programa terminal Minicom.

Una vez hechos los ajustes, conecta el OMAP al ordenador vía el puerto serie UART1/2 eligiendo el puerto serial UART1 como el default como se muestra en la siguiente figura:

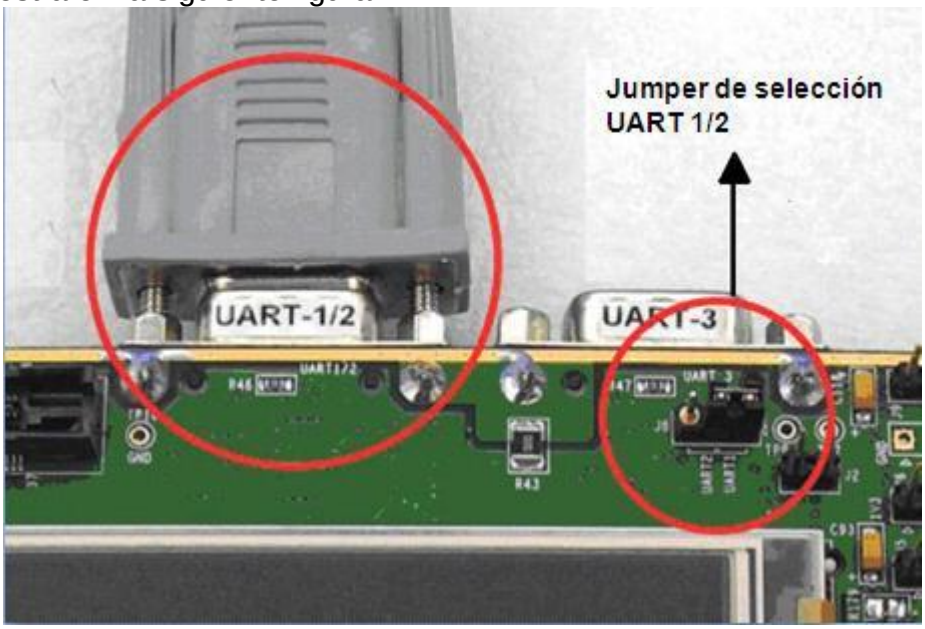


Figura 5.12 Selección del puerto serie 1 en la tarjeta OMAP.

Enciende la Tarjeta OMAP conectándola al adaptador de corriente AC. Inmediatamente en consola debes obtener algo similar a lo siguiente:

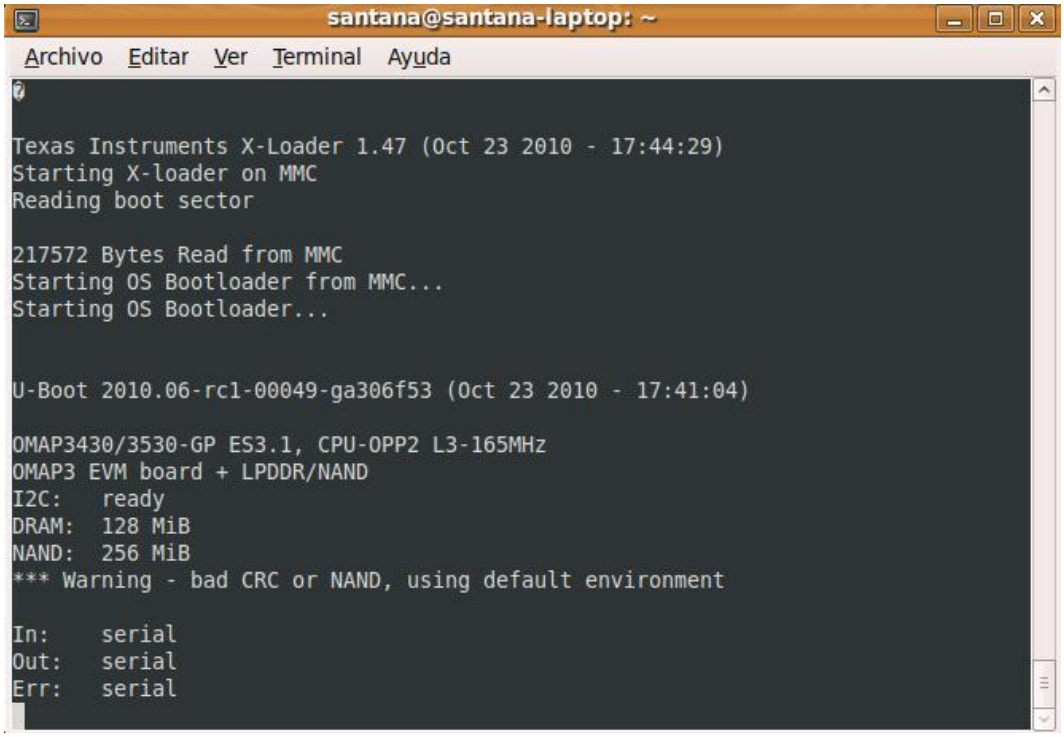


Figura 5.13 Mensajes de inicialización del OMAP en consola usando Minicom.

5.3.2 Modificaciones del OMAP

Para poder descargar las aplicaciones al OMAP se encontró que la tarjeta venía con defectos de fábrica, las aplicaciones se descargan vía el puerto USB-OTG en el cual el OMAP de fábrica tiene una resistencia que lo manda a tierra e impide la detección de la tarjeta al ordenador vía el puerto. La documentación nos dice que se debe remover la siguiente resistencia de

la tarjeta:

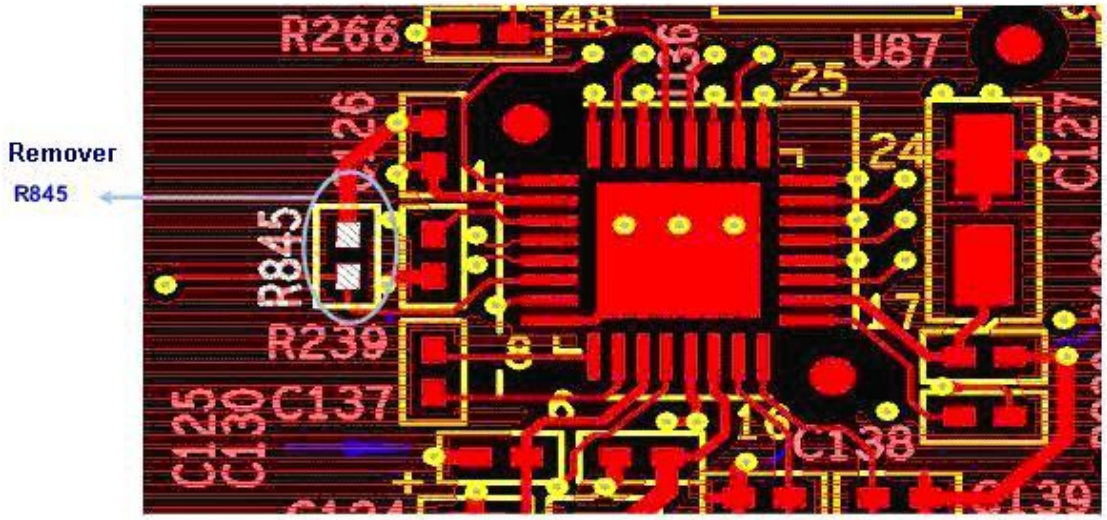


Figura 5.14 Parte inferior de la tarjeta OMAP Rev-G.

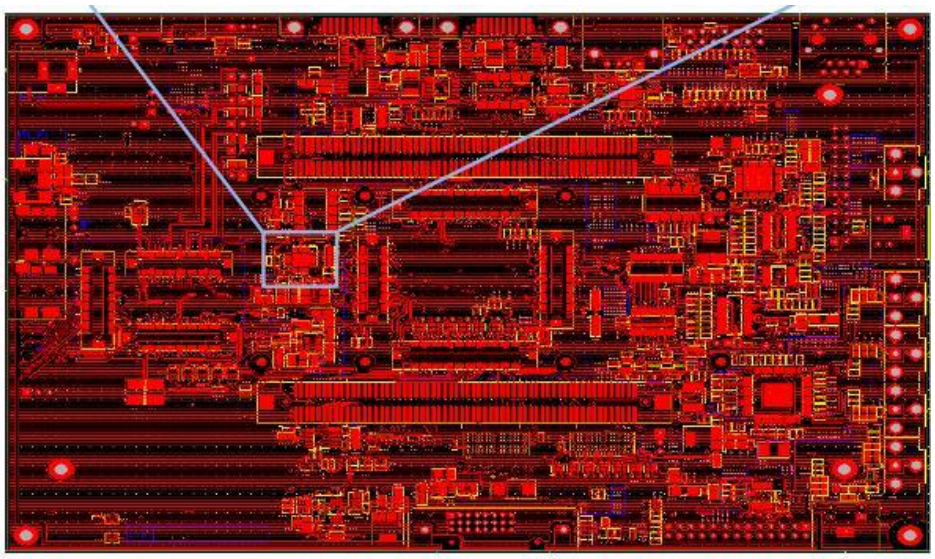


Figura 5.15 Parte inferior de la tarjeta OMAP Rev-G.

5.3.3 Preparando la tarjeta OMAP para correr el sistema operativo

El Módulo OMAP debe correr el Sistema Operativo Android a través de la interfaz MMC/SD. Esta interfaz será usada para iniciar el Sistema Operativo desde una tarjeta SD. La figura 5.16 muestra la interfaz MMC/SD y la tarjeta SD usada en el OMAP.

Figura 5.16 Tarjeta SD e Interfaz de la tarjeta OMAP usada para cargar el sistema operativo.

El modo de inicio puede ser seleccionado usando el DIP switch 4 instalado en la parte inferior de la tarjeta OMAP. [19] Esta selección identifica la localización de donde los archivos de inicio son ejecutados. El switch se muestra en la figura 5.17:

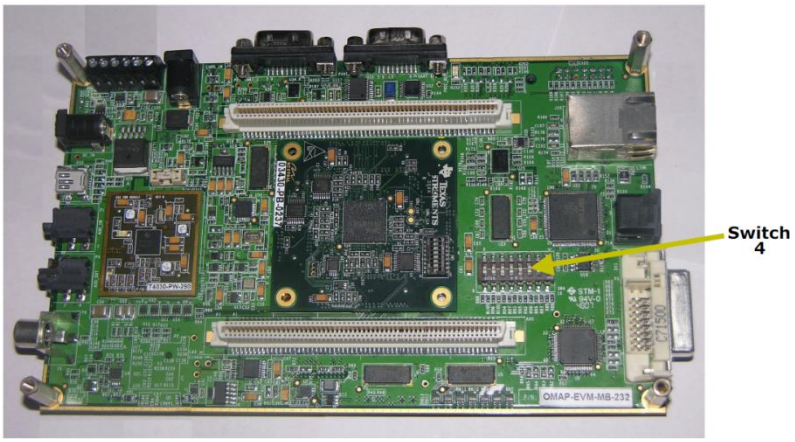


Figura 5.17 DIP Switch 4 usado para controlar el modo de inicio del procesador.

Con la siguiente configuración del DIP switch 4, el módulo del procesador iniciará el sistema operativo desde la interfaz MMC/SD:

Modo de inicio	SW-1	SW-2	SW-3	SW-4	SW-5	SW-6
MMC	ON	ON	ON	OFF	OFF	OFF

Tabla 5.1 Posición del DIP switch 4 para iniciar desde la interfaz MMC.

5.3.4 Preparando la tarjeta SD para iniciar el sistema operativo

Para hacer posible el inicio desde la MMC/SD, la tarjeta SD debe ser particionada en dos, esto es, una partición llevara los archivos de inicio y la imagen del sistema operativo, y la otra partición llevará el sistema de archivos del sistema operativo.

Los archivos mostrados en la figura 5.18, son los archivos necesarios para correr con la tarjeta SD [22]:

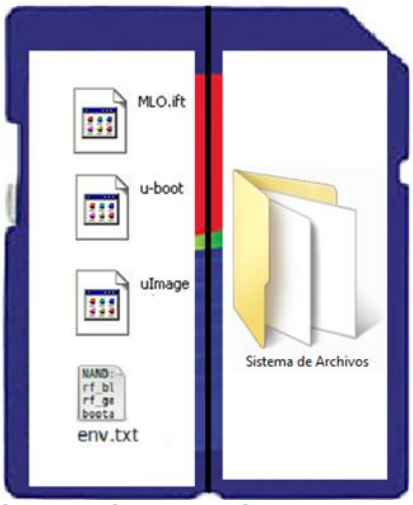


Figura 5.18 Archivos en la memoria SD para correr un sistema operativo.

En la parte izquierda de la figura se muestran los archivos de inicio y la parte derecha el Sistema de archivos del Sistema Operativo.

La descripción de estos archivos es la siguiente:

- U-boot: Es un cargador de arranque disponible para un amplio rango de arquitecturas embebidas.
- MLO: Este archivo depende del archivo u-boot, es usado para inicializar la memoria.
- uImage: Imagen del Kernel del sistema operativo.
- env.txt: Argumentos de inicio que le indican al procesador donde se encuentran los archivos para iniciar el Sistema Operativo.
- Sistema de Archivos: Estructura la información guardada en una unidad de almacenamiento que luego será representada ya sea textual o

gráficamente usando un gestor de archivos. La mayoría de los sistemas operativos manejan su propio sistema de archivos.

Actualmente existe una gran comunidad que le da soporte al Sistema Operativo Android, cada cierto tiempo se liberan archivos pre-compilados que pueden ser descargados desde la página de soporte {9} con versiones más actuales del sistema operativo para ser instalados en una tarjeta SD.

Figura 5.19 Sistema operativo Android sobre la plataforma OMAP.

5.4 Descargar aplicaciones al OMAP

Si quieres correr las aplicaciones directamente en el OMAP desde el ambiente de desarrollo Eclipse debes hacer algunos ajustes [23], que a continuación se muestran:

Ya instalado el SDK de Android y el Plugin ADT de Eclipse, crea el siguiente archivo:

/etc/udev/rules.d/51-android.rules

Se deberá agregar la siguiente línea para ¿?:

SUBSYSTEM=="usb", SYSFS{idVendor}=="0bb4", MODE="0666"

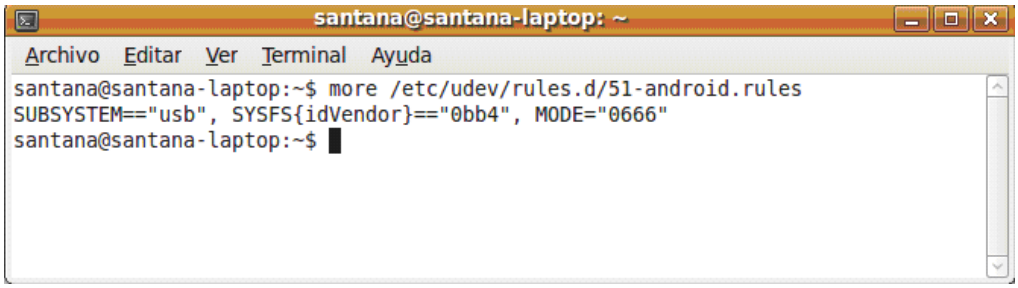


Figura 5.20 Primer ajuste para correr las aplicaciones en el OMAP.

.

Lo siguiente es encontrar el “ldVendor” de la tarjeta OMAP, para encontrarlo:

Con el puerto USB-OTG del OMAP desconectado del ordenador, escribe el siguiente comando para listar todos los dispositivos conectados al ordenador vía USB:

lsusb

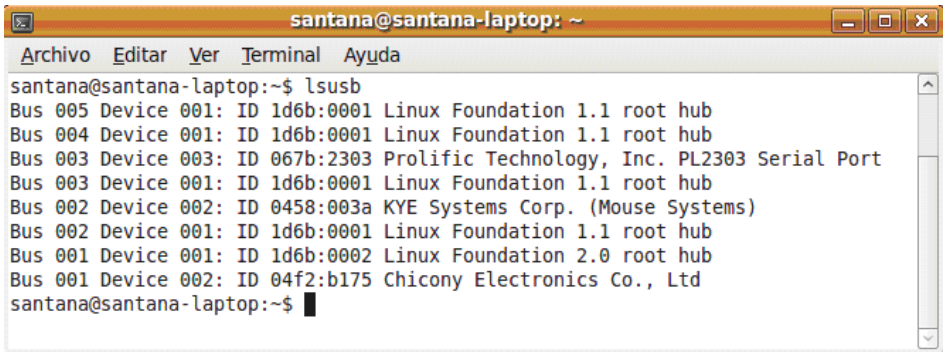


Figura 5.21 Salida del comando “lsusb”

Con el OMAP corriendo el sistema operativo Android, configúralo para que el puerto USB-OTG sea usado para descargar las aplicaciones. En el OMAP ve a:

.

Settings>applications>development

Selecciona:

USB debugging, como se muestra en la figura 5.22:

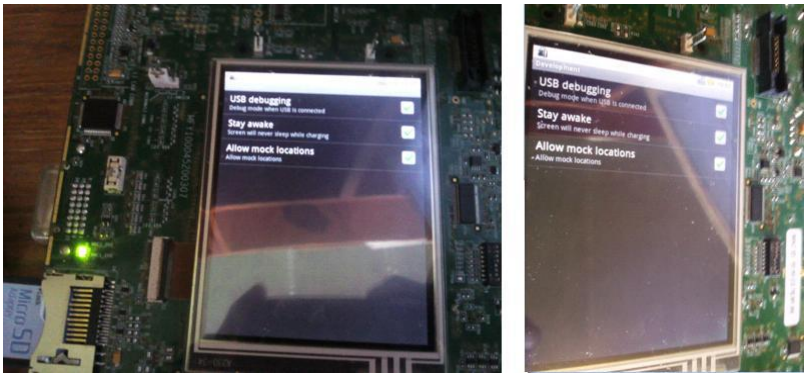


Figura 5.22 Configuración del sistema operativo Android para correr aplicaciones en el OMAP.

Posteriormente se conecta el OMAP al ordenador vía el puerto USB-OTG como se muestra a continuación:

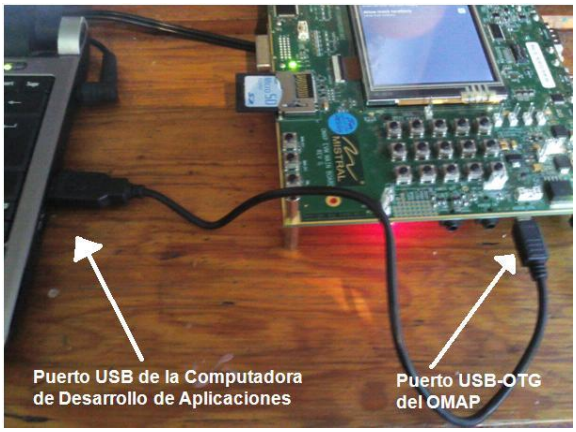


Figura 5.23 Computadora de desarrollo y el OMAP conectados vía el puerto USB y USB-OTG.

Vuelve a escribir el comando “lsusb”. Ésta vez debes ver una pequeña diferencia en la lista:

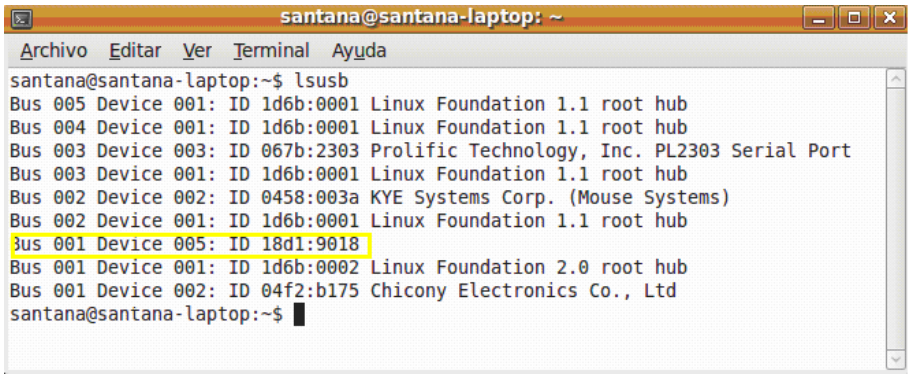


Figura 5.24 Ejecución del comando “lsusb”.

La diferencia está en la línea subrayada como se muestra en la figura anterior: “18d1:9018”, el numero antes de los “:” es: “18d1” y éste es el “IdVendor” del OMAP.

En seguida se crea un archivo con el siguiente comando:

sudo gedit /etc/udev/rules.d/55-beagle.rules

Y escribe la siguiente línea en el nuevo archivo:

SUBSYSTEM=="usb", ATTRS{idVendor}=="18d1", MODE="0666"

Guarda y cierra gedit, y después ejecuta el siguiente comando:

chmod a+rx /etc/udev/rules.d/55-beagle.rules

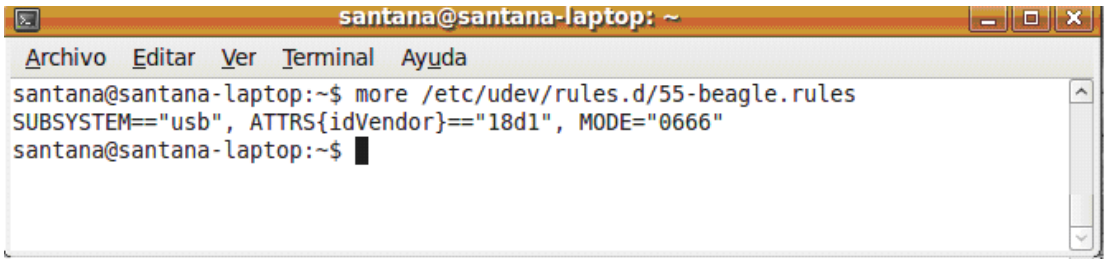


Figura 5.25 Segundo ajuste para correr las aplicaciones en el OMAP.

Desconecta el OMAP y conéctalo de nuevo, ahora desde la carpeta del SDK de Android lista los dispositivos disponibles, como se muestra en la siguiente figura:

Figura 5.26 Muestra que el OMAP ha sido reconocido por el ambiente de desarrollo Eclipse.

Lo anterior significa que el OMAP ha sido reconocido por AVD (Android Virtual Device) y puedes usarlo en el ambiente de desarrollo Eclipse para descargar las aplicaciones.

5.5 Corriendo una aplicación en el OMAP

Corre Eclipse como se muestra en la figura 5.27. En Eclipse se harán ajustes para que se puedan correr las aplicaciones en el OMAP, ahora ve a:

Run>Run configurations>Target

Y selecciona: ”Manual” para elegir el OMAP como el dispositivo default:

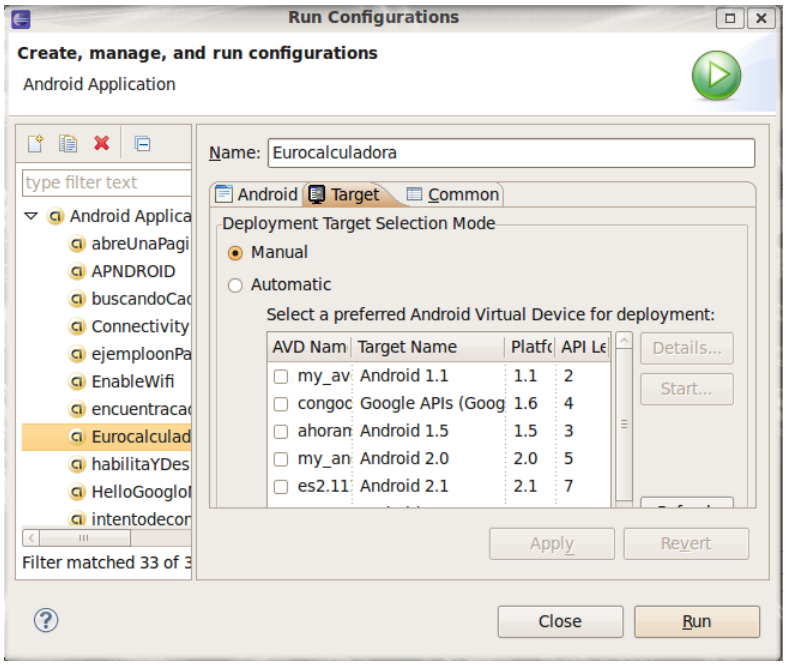


Figura 5.27 Elección manual del dispositivo donde se correrá la aplicación.

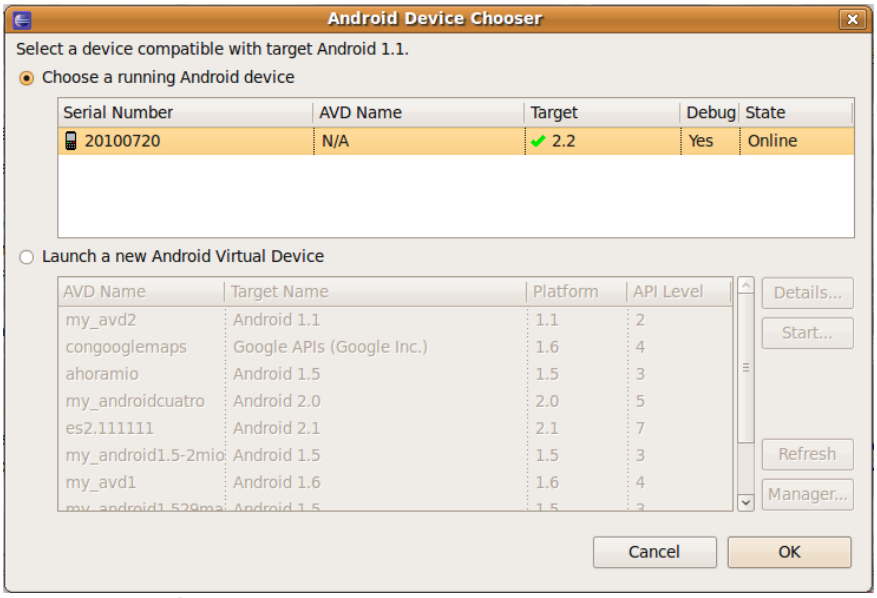


Figura 5.28 Elección del OMAP como dispositivo para correr la aplicación.

Después de los ajustes ya mencionados, mostrados en las figuras anteriores, la aplicación estará corriendo en el OMAP.

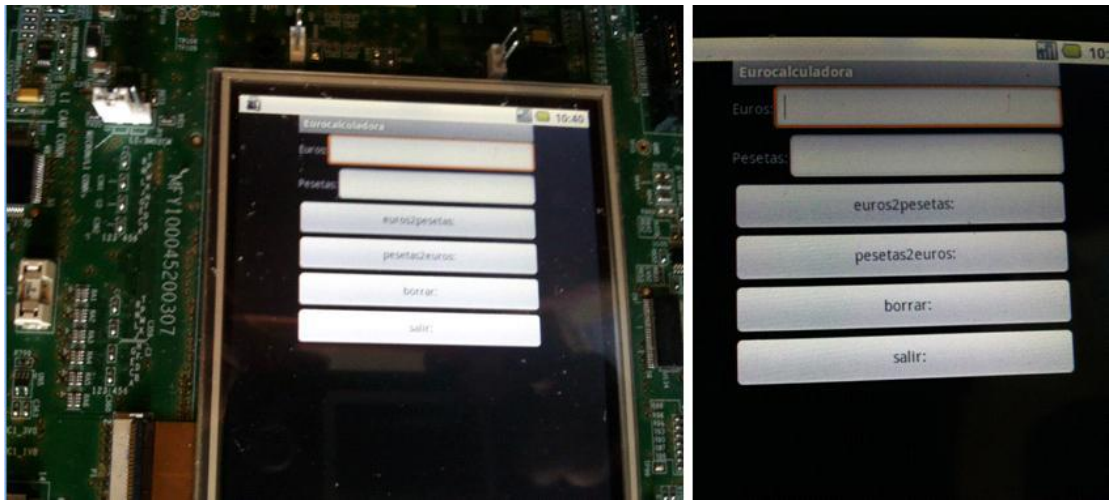


Figura 5.29 Aplicación corriendo en el OMAP.

A continuación se mostrará que la aplicación mencionada anteriormente también se puede correr en el simulador de Android.

Para esto es necesario elegir un dispositivo simulador AVD en:

run>Run configurations

Escoge un simulador de dispositivo de los integrados en el ambiente de desarrollo para correr la aplicación:

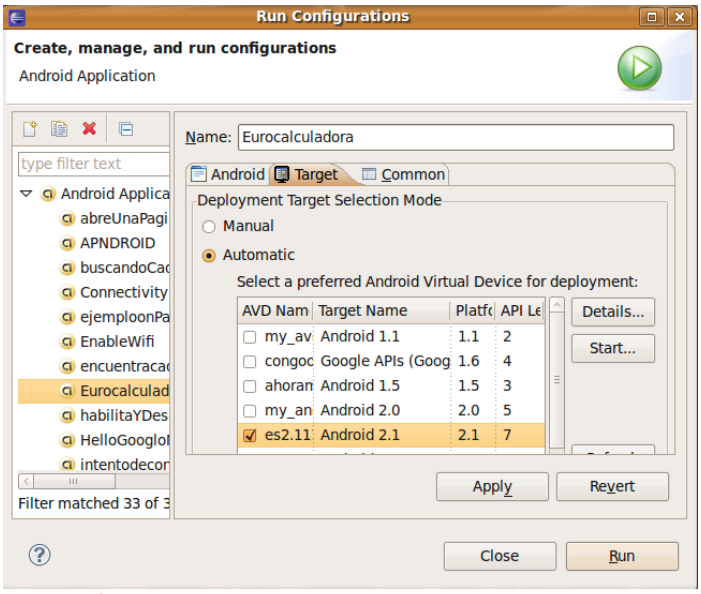


Figura 5.30 Elección de un simulador del sistema operativo Android para correr la aplicación.

La aplicación ahora estará corriendo en el dispositivo virtual como se muestra en la siguiente figura:

Figura 5.31 Aplicación corriendo en el simulador de Android.

Se puede concluir que las aplicaciones se pueden correr de dos formas diferentes, tanto virtual como físicamente, siguiendo los pasos mostrados anteriormente. Se tiene como ventaja poder ejecutar las aplicaciones sobre el simulador cuando no se cuenta con el equipo físico, pero como desventaja que es imposible hacer uso del hardware que proporciona el simulador tal como la tarjeta de red Wi-Fi.

5.6 Compilando el Kernel

Para construir los componentes de software [22] usando las fuentes del Kernel de Linux se requiere una herramienta de compilación cruzada ARM, esta herramienta ARM puede ser descargada desde el repositorio de imágenes precompiladas de Android {7}:

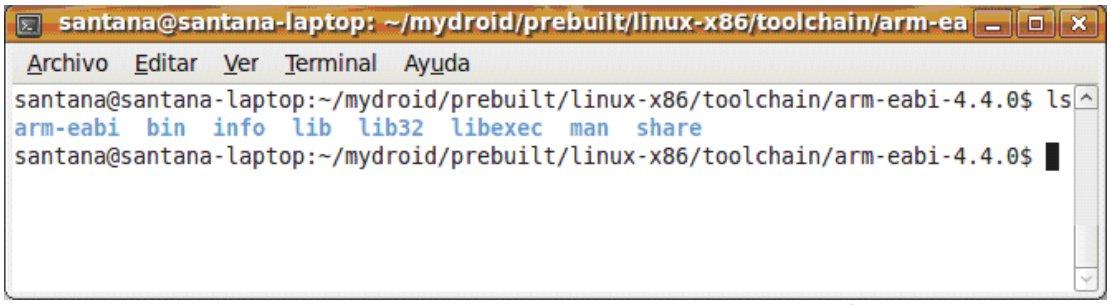


Figura 5.32 Directorio de la herramienta de compilación cruzada.

Ahora crea una variable de entorno que apunte a la carpeta “bin” de la herramienta de compilación cruzada, como se muestra a continuación:

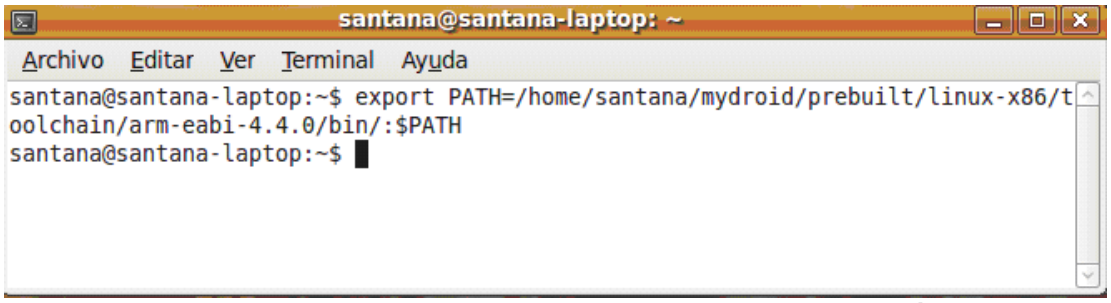


Figura 5.33 Variable de entorno de la herramienta de compilación cruzada.

De la página del proyecto de Android {8}, descarga las fuentes del Kernel más recientes, la última versión llamada en el momento de redacción de esta tesis es:

Android_Linux_Kernel_2_6_32.tar.gz

Descomprime las fuentes del Kernel y dentro del folder de éstas ejecuta los siguientes comandos:

#> make CROSS_COMPILE=arm-eabi- distclean

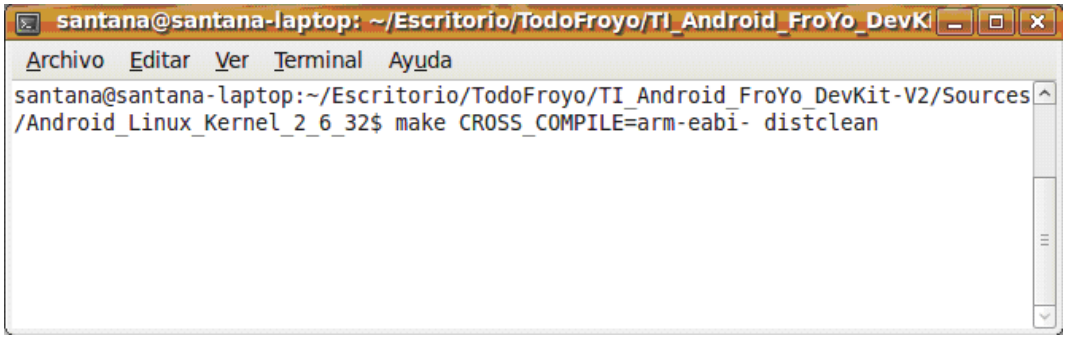


Figura 5.34 Salida del comando "distclean".

El comando anterior deja las fuentes del Kernel limpias, borrando archivos como el ".config" (archivo que contiene la configuración del kernel de Linux, a partir de éste mismo se crea la imagen del mismo) ubicado en la carpeta raíz de las fuentes del kernel.

Se crea la configuración que por defecto viene en las fuentes del Kernel con el siguiente comando:

#> make CROSS_COMPILE=arm-eabi- omap3_evm_android_defconfig

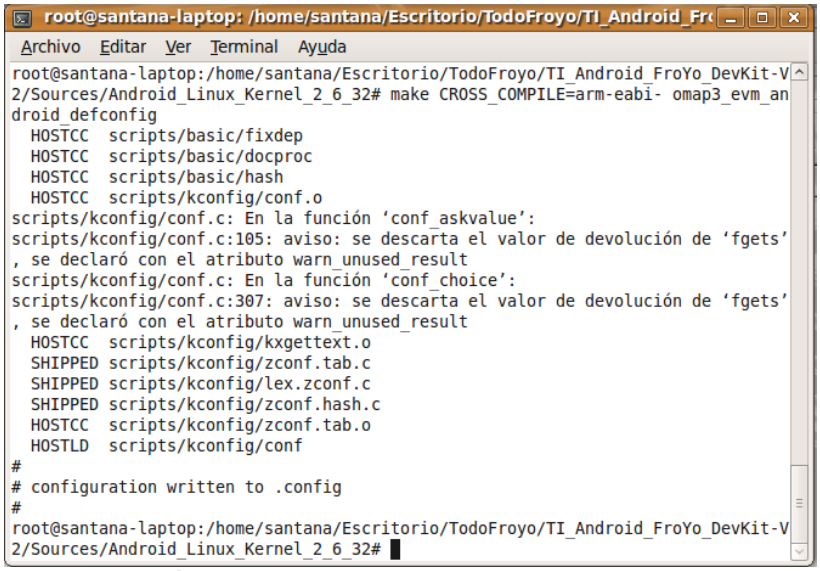


Figura 5.35 Ejecución en consola del comando para crear las fuentes del Kernel.

Ahora se ha creado en el directorio principal de las fuentes del Kernel un archivo llamado ".config" con las características mínimas para correr el sistema operativo Android sobre la plataforma OMAP.

Este archivo debe ser modificado para agregarle los diferentes drivers para que el sistema operativo sea capaz de detectar diferentes módulos de hardware que en un futuro sean integrados al OMAP.

Con el siguiente comando se configurarán las fuentes del Kernel de Linux:

make CROSS_COMPILE=arm-eabi- menuconfig

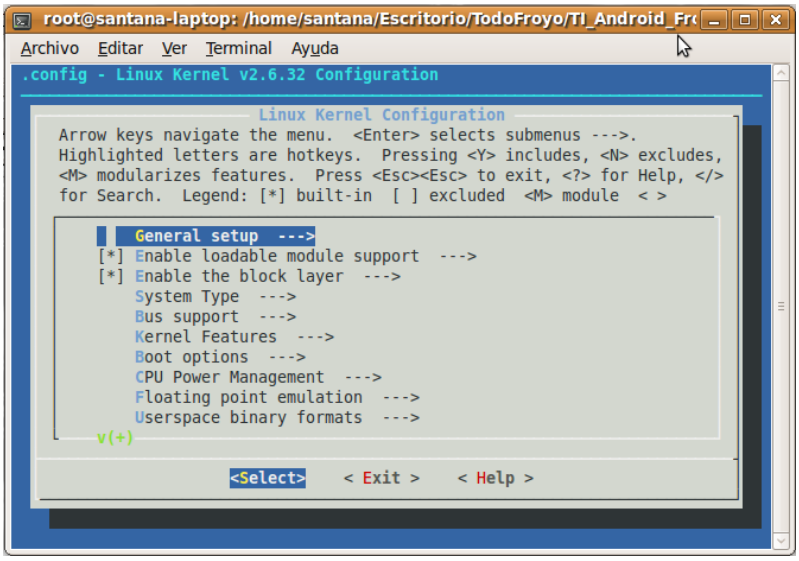


Figura 5.36 Salida del comando "menuconfig".

Aquí es donde se encuentran configuradas las fuentes del Kernel. Con "menuconfig" cargas el archivo ".config" creado por la configuración default que viene con las fuentes del Kernel, este archivo será el que se modificará para agregarle características nuevas al Kernel o soporte para una gran cantidad de dispositivos. Ahora a continuación se procede a crear la imagen del Kernel de Linux con el siguiente comando:

make CROSS_COMPILE=arm-eabi- uImage

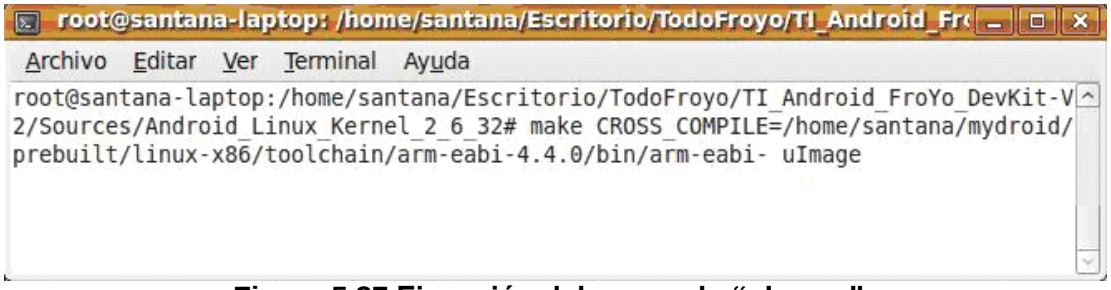


Figura 5.37 Ejecución del comando "uImage".

El comando anterior construirá la imagen del Kernel de Linux en el siguiente directorio: arch /arm/boot/uImage

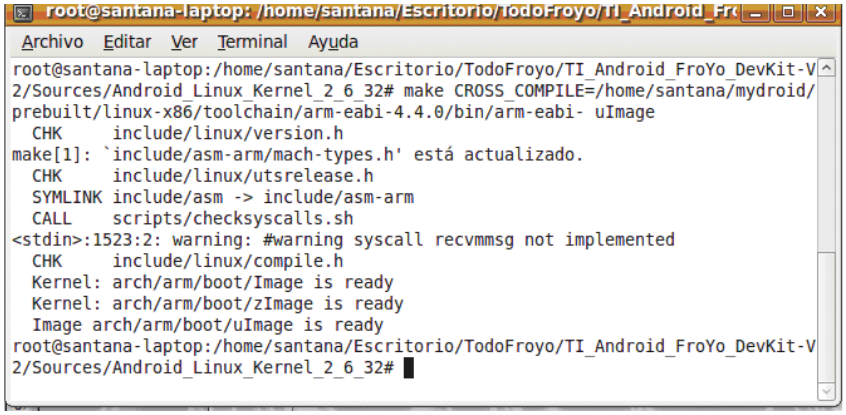


Figura 5.38 Directorio donde se creó la imagen del Kernel.

La imagen del Kernel "uImage" creada, es el archivo que irá guardado en la partición de inicio de la tarjeta SD del sistema operativo Android.

5.7 Instalación del sistema operativo Android en una computadora portátil

Debido a que con el simulador instalado en Eclipse para correr aplicaciones para el sistema Operativo Android no permite controlar los módulos de hardware en este caso el modulo Wi-Fi y además no se cuenta con el modulo de hardware Wi-Fi para plataforma OMAP, y ya que se debe de probar el software "ComEficiente" se buscaron otras alternativas para poder correr esta aplicación en un sistema Operativo Android.

La instalación del sistema Operativo Android en un ordenador portátil nos ayuda a usar el modulo de hardware Wi-Fi con el que cuenta éste para correr nuestra aplicación.

Esta versión x86 de sistema operativo Android está basada en Android 2.2 (Froyo), el *target* es el procesador ATOM.

http://www.thinq.co.uk/2010/6/25/intel-building-x86-compliant-android-os/

La forma de instalación es la siguiente:

- Descargar la imagen ISO del sistema operativo Android de la página de desarrollo. {10}

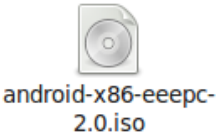


Figura 5.39 Imagen ISO.

- Usar un programa que permita crear imágenes de disco en dispositivos USB (el recomendado es “UNetbootin” para UBUNTU).

Figura 5.40 Herramienta UNetbootin.

- Ahora ya con la USB con la imagen del sistema operativo introducirla en un puerto USB de un ordenador con procesador ATOM.

- Configurar el ordenador para que inicie desde el puerto USB y de esta forma cargar el sistema operativo. Al iniciar nos mostrara la siguiente pantalla:

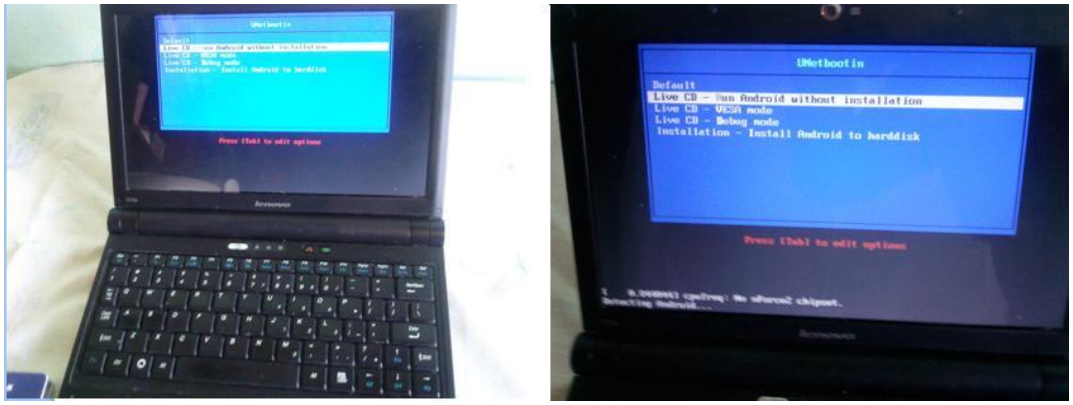


Figura 5.41 Ordenador iniciando Android.

1. Elegir la opción default (correr Android sobre el ordenador sin instalar) y en seguida iniciara Android en el ordenador.

Figura 5.42 Android corriendo en un ordenador con procesador ATOM.

Ya con esta versión del sistema operativo sobre este tipo de arquitectura de computadoras podemos hacer uso del hardware Wi-Fi que incluye.

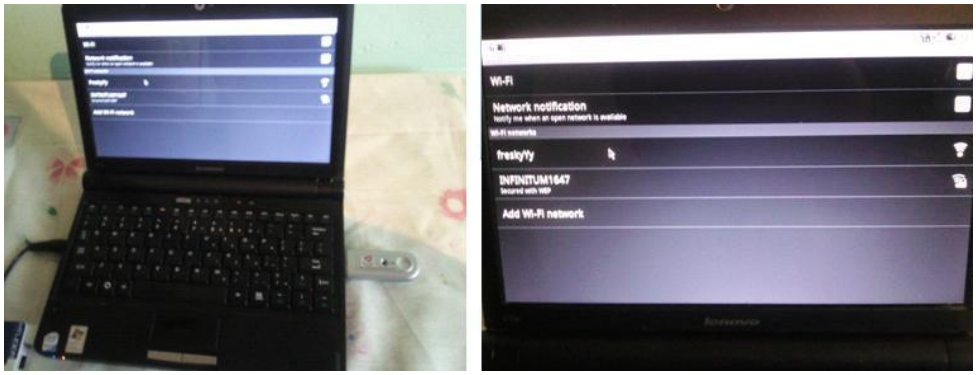


Figura 5.43 Usando el hardware Wi-Fi del ordenador.

5.8 Plataforma TAO

La tarjeta “Thunder” de Technexion, al igual que la tarjeta de desarrollo OMAP, cuenta con periféricos de entrada y salida como una Interfaz LCD, puerto USB/OTG y el interfaz para conectar un módulo CPU TAO-3530. El módulo TAO-3530 está basado en el procesador de aplicación OMAP 3530 de TI, que tiene integrado un procesador ARM y un DSP.

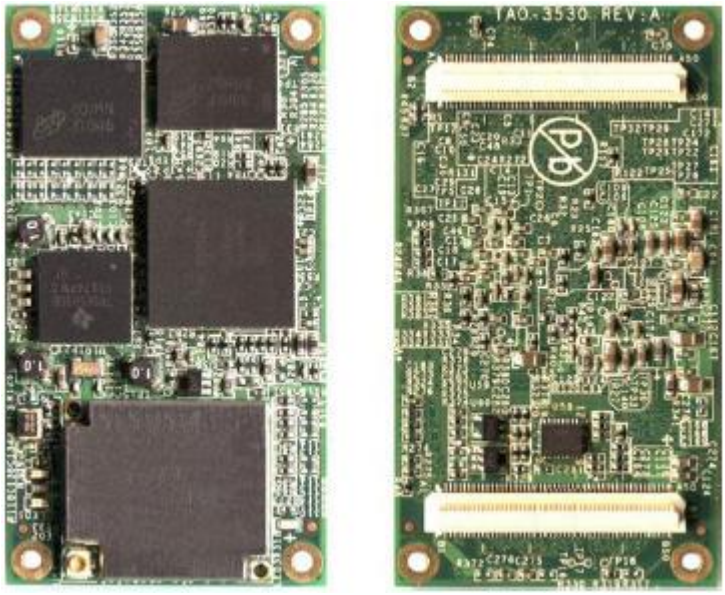


Figura 5.44 Procesador TAO,

Los conectores B1 y B2 son usados para que el Procesador sea montado sobre la tarjeta “Thunder”.

Figura 5.45 Vista superior de la tarjeta “Thunder”

La figura 5.45 muestra los conectores B1 y B2 los cuales son utilizados para montar el Procesador TAO-3530. Entre las características importantes del módulo TAO-3530-W, además del Procesador ARM y el DSP, tiene integrado una interfaz Wireless LAN [22].

La instalación del Sistema Operativo Android para esta plataforma TAO es la misma que para la tarjeta OMAP, al igual se necesita iniciar el Sistema Operativo desde una tarjeta SD particionada, pero con la ventaja de que esta Plataforma TAO tiene integrada la interfaz Wi-Fi.

Figura 5.46 Tarjeta "Thunder" con procesador TAO corriendo el Sistema Operativo Android.

5.9 Desarrollo de la aplicación "ComEficiente"

Una vez expuestos los objetivos y características principales de la aplicación y las decisiones tomadas en cuanto a su diseño, se desarrollarán en este apartado los aspectos relacionados con su implementación. El fin perseguido en las siguientes líneas es mostrar el código más relevante de *ComEficient.*

5.9.1 Pruebas antes de la aplicación final

```
wifiManager = (WifiManager) this.getSystemService(Context.WIFI_SERVICE);

if(wifiManager.isWifiEnabled())
{
    wifiManager.setWifiEnabled(false);

}
else
{
wifiManager.setWifiEnabled(true);
}
```

Código 5.6 Código que prende la interfaz Wi-Fi si está apagada y la apaga si está encendida.

Figura 5.47 Muestra que la interfaz Wi-Fi ha sido habilitada

5.9.2 Captura de paquetes “Beacon” de los Puntos de Acceso

Los Puntos de Acceso envían periódicamente "señales", para anunciar su presencia y que todas las estaciones que estén en el rango sepan cuáles Puntos de Acceso están disponibles. Estos paquetes contienen varios parámetros entre ellos el SSID del AP y su Dirección MAC. Esta información será utilizada por el cliente a la hora de comunicarse.

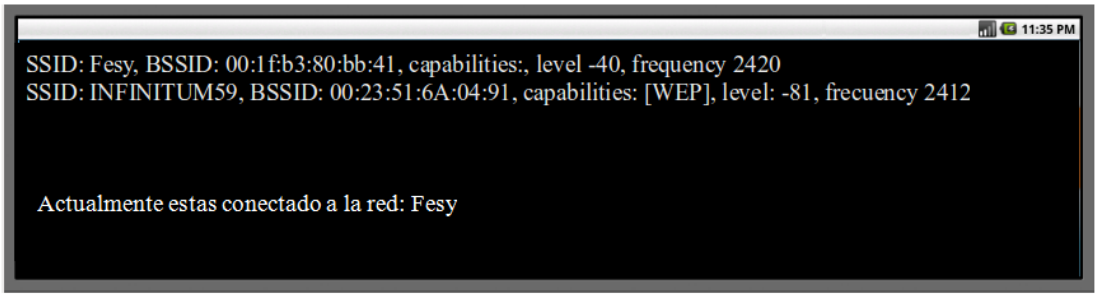


Figura 5.48 Paquetes “Beacon”.

Como se puede observar en la figura 5.48, la información proporcionada por los paquetes de señalización, contiene información sobre el SSID, el BSSID, el tipo de seguridad de la red, la frecuencia a la que se transmite y la potencia de transmisión del Punto de Acceso.

```
public class ComEficiente extends Activity
{

private WifiManager mainWifi;
List<ScanResult> wifiList

public void onCreate(Bundle savedInstanceState)
{

mainWifi=(WifiManager) getSystemService (Context.WIFI_SERVICE);
..      refresh.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v)
{
// TODO Auto-generated method stub
mainWifi.startScan();
escanear();
}
});
}


private void escanear ()
{
.
.
.

wifiList = mainWifi.getScanResults();
ListIterator<ScanResult> results = wifiList.listIterator();
.
.
.
}
}
```

Código 5.7 Muestra el uso de la clase " WifiManager" de Android.

En el código 5.7 se puede ver que se instancia "mainWifi" de la clase "WifiManager" (clase que administra todos los aspectos de conectividad Wi-Fi). Más adelante a partir de esta instancia se manda a llamar al método "getScanResults()", el cual almacenará la información de las redes escaneadas en una lista, a partir de esta lista, podemos obtener y desplegar en pantalla cada uno de los parámetros mostrados en la figura anterior.

Como se mencionó anteriormente en caso de que no se halle una red disponible abierta la conexión a Internet se hará vía Red Celular, por medio del módem GSM/ GPRS.

5.9.3 Conexión a un Punto de Acceso

Recordando lo dicho en el capítulo tres, entre los paquetes de Administración está el paquete de Solicitud de Asociación, este es el que inicia la conexión. Lleva diversa información, entre ella el SSID del Punto de Acceso al cual nos queremos conectar.

Una vez ya capturada la información de cada uno de los Puntos de Acceso al alcance, lo siguiente es revisar cada uno de estos paquetes obtenidos de los diferentes Puntos de acceso, descartando aquellos paquetes con algún tipo de seguridad. Ahora de los paquetes restantes hay que elegir el paquete que nos ofrezca más potencia de transmisión. Ya con este paquete elegido se obtiene el nombre de la red a la que nos queremos conectar (el SSID) y la dirección Mac del punto de acceso (BSSID).

```
conectar.setOnClickListener(new View.OnClickListener()
{
@Override
  public void onClick(View v) {

          WifiConfiguration wifiConfig = new WifiConfiguration();
          wifiConfig.BSSID=cadenaBBSSID;
         wifiConfig.priority = 1;

          wifiConfig.allowedKeyManagement.set(WifiConfiguration.KeyMgmt.NONE);

          mainWifi.enableNetwork(mainWifi.addNetwork(wifiConfig), true);
 }
});
```

Código 5.8 Muestra el uso de la clase "WifiConfiguration" de Android.

En el código 5.8 se instancia "wifiConfig" de la clase "WifiConfiguration" (clase representando una red Wi-Fi configurada). Al haber elegido la Red Wi-Fi que se hará la conexión, se prepara el paquete de administración "Solicitud de Asociación" para hacer la petición de conexión con el Punto de Acceso. En el código se ven los parámetros de SSID y BSSID usados para hacer la petición y el método "enableNetwork()" de la clase WifiManager, para enviar el paquete de solicitud de Asociación. Si el Punto de Acceso está interesado en esa comunicación, asigna espacio de memoria para ese cliente y le envía una Respuesta de Asociación, el cual es un paquete que lleva la respuesta a la estación cliente, incluye informaciones diversas, entre ellas el MAC Address del usuario.

Figura 5.49 Tarjeta "Thunder" con procesador TAO corriendo la aplicación "ComEficiente".

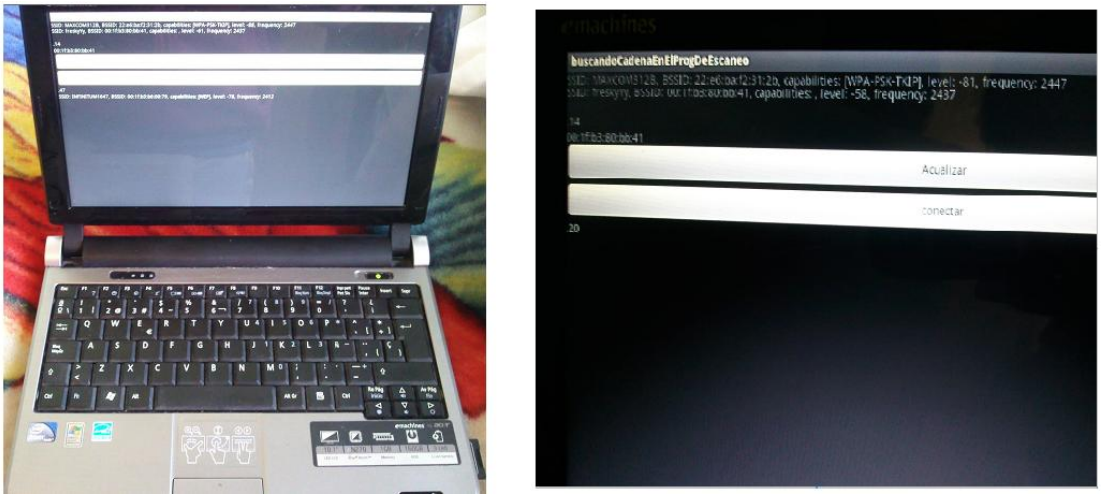


Figura 5.50 Ordenador corriendo la aplicación "ComEficiente".

5.10 Conclusiones

En este capítulo se desarrolló la arquitectura de computo móvil propuesta en el capítulo anterior, se desarrolló el diagrama esquemático y a partir de éste, la creación del PCB.

Se desarrollaron las primeras aplicaciones para el sistema operativo Android, siendo ejecutadas en los simuladores de Android que se instala en el ambiente de desarrollo de Eclipse.

Se realizó la configuración del módulo de evaluación OMAP35x y de la tarjeta SD para que se pudiera iniciar el sistema operativo. Para descargar al módulo las aplicaciones realizadas en Eclipse para el sistema operativo Android, se hace vía el puerto USB-OTG que trae el módulo de desarrollo, se detectó que éste venía con *"bugs"* de ensamble que impedían usar el puerto USB-OTG por lo que se tuvieron que corregir removiendo una resistencia, para habilitar el puerto USB-OTG, y así poder hacer la configuración tanto de hardware y software.

En este capítulo se mostró la forma detallada de como hacer una compilación del Kernel del sistema operativo Android. El cual se encuentra entre los archivos que están en la tarjeta SD para iniciar el sistema operativo Android.

Se describe detalladamente el completo desarrollo de la aplicación "ComEficiente" para Android desde su diseño y funcionalidad, La aplicación "ComEficiente" propuesta en el capítulo anterior usa los paquetes "Beacon" y "Solicitud de Asociación" que establece el estándar 802.11 para la comunicación con el punto de acceso. Dicha aplicación de alguna forma debió ser probada, ya que el simulador del sistema operativo Android no permite manejar los módulos virtuales de hardware y no se cuenta con la interfaz Wi-Fi en el módulo de evaluación OMAP35x, se tuvieron que ver alternativas para correr la aplicación "ComEficiente". Por lo tanto se realizó la instalación del sistema operativo Android en un ordenador x86 con procesador ATOM de Intel y en la plataforma "Thunder" con el Módulo TAO, esta versión del sistema operativo permite manipular la tarjeta Wi-Fi con la que cuenta el ordenador y la plataforma "Thunder", de esta forma se probó el funcionamiento de la aplicación. Con respecto a la conexión a Internet vía Red Celular, el sistema operativo Android está configurado para usar las redes celulares para transmitir datos por default [26], pero hay que tomar en cuenta que cuando una conexión de datos Wi-Fi está activa, todo el tráfico de datos usará esta conexión {6}.

CAPÍTULO 6

CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se hace un repaso global al proyecto, presentando las conclusiones finales en comparación con los objetivos marcados inicialmente y los posibles trabajos futuros.

6.1 Conclusiones finales

Una vez concluidas las principales tareas que forman este proyecto, es el momento en el que se puede hacer balance y crítica de los resultados obtenidos. Así pues, repasando los objetivos inicialmente marcados pueden sacarse las siguientes conclusiones:

6.1.1 Conclusión sobre los Objetivos Específicos: Software

- Se estudiaron las redes WLAN y GSM, así como los protocolos para comunicación inalámbrica Ethernet y GPRS.
- Se estudiaron los diferentes tipos de acceso al canal su configuración para el sensado y conexión a puntos de acceso.

A lo largo de todo el presente proyecto se ha conseguido obtener un conocimiento amplio de las redes inalámbricas WLAN y GSM. Su arquitectura, sus componentes y características así como la forma en que trabajan, incluyendo los estándares y protocolos de comunicación que las rigen.

Con respecto a las redes de tecnología GSM, dependiendo del grado de evolución, como se mencionó en el tercer capítulo, fueron clasificadas en sus generaciones, lo que las diferencian unas de las otras es la forma en que tienen acceso al canal, dando como resultados más sobresalientes, la capacidad de usuarios y la seguridad de la red. Por otro lado, las redes WLAN al igual que las redes GSM fueron estudiadas a profundidad, ya que gran parte de este proyecto está encaminado a tener una conexión a puntos de acceso. El estudio de las redes inalámbricas WLAN fue enfocado en su modo operativo infraestructura, donde los clientes sensan para hacer una conexión a un punto de acceso intercambiando paquetes para lograr la conexión.

6.1.2 Conclusión sobre los Objetivos Específicos: Hardware

- Se estudiaron y analizaron los dispositivos módem GSM/GPRS, y de dispositivos de interfaz de red inalámbrica Wi-Fi 802.11g.

Después de haber cumplido este objetivo, nos ha permitido conocer las características de las redes Inalámbricas WLAN y GSM, se procedió a analizar los dispositivos que nos permiten hacer una conexión a este tipo de redes. El presente proyecto está enfocado a hacer conexiones a redes inalámbricas WLAN y GSM vía módulos de hardware GSM/GPRS y una

interfaz de red inalámbrica 802.11, por lo que es indispensable conocer cómo trabajan este tipo de dispositivos.

Se estudió y se analizó el módulo de hardware que será conectado a la tarjeta OMAP, la tarjeta hija WL1271 de Mistral, desde conocer las interfaces con las que cuenta la tarjeta, hasta el método usado para ser empotrada sobre la tarjeta OMAP. También se analizó el funcionamiento de módems de fabricantes tales como: SIEMENS y Wavecom; la forma en que trabajan todos los módems de telefonía es bastante similar.

6.1.3 Conclusión sobre el Objetivo General:

- Integrar y desarrollar hardware y software de infraestructura para comunicaciones Cliente-Servidor utilizando sistemas de comunicación inalámbrica GSM/GPRS/Ethernet para aplicaciones móviles embebidas.

Ahora después de cumplir con los objetivos anteriores se desarrolló software para dispositivos móviles tomando como sistema operativo a Android. Una aplicación para realizar el enlace a Internet que hemos llamado **"ComEficiente"** controla el módulo Wi-Fi del dispositivo móvil para hacer las conexiones con puntos de acceso. Todo el diseño e implementación de la aplicación ha sido documentado.

El desarrollo de Hardware fue hecho con ayuda del software para la creación de PCB, **Altium Designer**. Se creó cada símbolo esquemático de cada elemento que compone el hardware, también se creó cada símbolo footprint para su representación en el PCB de cada componente. El PCB lleva módulos de interfaz LCD, MMC/SD, Wi-Fi, GSM/GPRS y los conectores donde el procesador irá montado.

La integración del software y Hardware no fue implementada debido a la ausencia de los módulos de hardware que irán conectados al OMAP, donde se pensaba ejecutar la aplicación "ComEficiente", la aplicación se ejecutó en el ordenador con el sistema operativo Android instalado y en la plataforma "Thunder" con el procesador TAO, ya que esta versión del sistema operativo permite controlar el módulo Wi-Fi del ordenador y de la plataforma "Thunder".

6.2 Trabajos Futuros

Como líneas de trabajos futuros se proponen los siguientes puntos:

- Integrar el mecanismo de comunicación eficiente **“ComEficiente”** al Kernel de Linux, para hacer la comunicación eficiente sin intervención del usuario.

- Realizar la integración del procesador OMAP35x, Android con el Módem WiFi, GPRS, GPS y probar su correcto funcionamiento en el diseño del circuito impreso.

Referencias Bibliográficas

[1] CISCO, Certified Network Associate, 5ta Edicion Tood Lammle, Editorial Sibex.

[2] 802.11 Wireless Networks, The Definitive Guide
Matthew S. Gast
O´REILLY,
April 2002

[3] 3G Wireless Demysified
Lawrence Harte, Richard Levine
McGraw-Hill, 2002

[4] Adding modular HSxPA cellular broadband connectivity to laptop computers and portable consumer equipment, author: Tom Yemington, Marketing Director, Icera Inc. 2007.

[5] Rootkits on Smart Phones: Attacks, Implications and Opportunities, Jeffrey Bickford, Department of Computer Science, Rutgers University, 2010.

[6] A Guide to Mobile Processors, PUBLISHED DECEMBER 2009, Authors: Linley Gwennap and Joseph Byrne, The Linley Group.

[7] Estudio cualitativo de arquitecturas orientadas a computación móvil
Mónica Aydé Vallejo Velásquez, José Edinson Aedo Cobo
Grupo de Microelectrónica y Control, Universidad de Antioquia
Diciembre 2005.

[8] Augmented Smartphone Applications Through Clone Cloud Execution, Byung-Gon Chun, Petros Maniatis, Intel Research Berkeley, This paper appears in HotOS 2009.

[9] Paul J.M. Havinga, “Mobile Multimedia Systems”, Ph.D. thesis University of Twente, February 2000, ISBN 90-365-1406-1.

[10] TA305 Technical Article “Nomadik - Open multimedia platform for next generation mobile devices”, 2004, STMicroelectronics.

[11] “The Freescale Semiconductor I.MX21 Processor”, Synchromesh Computing. 2004.

[12] S. Sohoni, Z. Xu, R. Min, and Y. Hu, “A Study of Memory System Performance of Multimedia Applications”, I n Proceedings of the ACMJoint International Conference on Measurement& Modeling of Computer System, (Cambridge, Massachusetts), June 16- 21 2001.

[13] Zheng Yuan and Talal Arnaout, “Architectural Level Power Consumption of Network on Chip”, Seminar: Network on chip. 2003.

[14] Design and Implementation of a Linux-based Smartphone Dong-Yun Shin and Sung-Soo Lim, December 3, 2005, School of Computer Science Kookmin University Seoul, Korea.

[15] LG’s Android Phone Makes Social Networking Even Friendlier
LG Electronics Inc.

Global Marketing Strategy
Jet Chang, Seoul, November 10, 2009
[16] Arturo Baz Alonso, Irene Ferreira Artime, María Álvarez Rodríguez, Rosana García Baniello, “Dispositivos móviles”, E.P.S.I.G: Ingeniería de Telecomunicación Universidad de Oviedo.
[17] Mobile Computing and Wireless Communications, chapter Wireless LANs – 802.11 and Mobile Ad Hoc Networks, AMJAD UMAR, Publication Date: July 2004.
[18] Mobile Computing and Wireless Communications, Chapter Cellular Networks -- From 1G to 5G, AMJAD UMAR, Publication Date: July 2004.
[19] Hardware User Guide, OMAP35x Evaluation Module, Copyright 2008 Mistral Solutions Pvt. Ltd.
[20] Hardware User Guide, WL1271Daughter card, Copyright 2009 Mistral Solutions Pvt. Ltd.
[21] Enfora Enabler IIIE Modem Integration Guide, Version 1.00, 06/26/08.
[22] TI Android Froyo DevKit V2 User Guide, October 25, 2010, Texas Instruments.
[23] Using Android in Industrial Automation, Technical Report, University of Applied Sciences Northwestern Switzerland, Manuel Di Cerbo, January 29, 2010.
[24] EVM Rework Instruction, HSUSB-OTG VBUS Fault Detection Feature, Texas Instruments.
[25] TAO-3530 System on module, USER´S GUIDE, TechNexion, OCTOBER 27, 2010
[26] Android 2.2.1 User´s Guide, Android TM Mobile Technology Platform 2.2.1, Google, October 23, 2010.

Referencias Electrónicas

{1} http://maemo.nokia.com/n900/

{2}http://www.samsungmobile.es/soporte/softwaremanuals/software.do?phone_model=GT-I8320

{3} http://www.htc.com/europe/product/hd2/specification.html

{4} http://www.motorola.com/Consumers/ES-ES/Consumer-Products-and-Services/Mobile-Phones/ci.Motorola-DEXT-ES-ES.alt

{5}http://www.sonyericsson.com/cws/products/mobilephones/overview/xperia x10?cc=gb&lc=en#view=features_specifications

{6} http://developer.android.com/guide/index.html

{7} http://android.git.kernel.org/?p=platform/prebuilt.git;a=tree;f=linux-x86/toolchain/arm-eabi-4.4.0;h=0f1763c115040381493278dde810a69e4bc37a76;hb=HEAD

{8} http://software-dl.ti.com/dsps/dsps_public_sw/sdo_tii/TI_Android_DevKit/02_00_00/index_FDS.html

{9}http://code.google.com/p/rowboat/

{10} http://www.android-x86.org/download

Abreviaciones

Abreviación	Descripción
ADT	Herramientas de desarrollo de Android
AP	Punto de Acceso
API	Interfaz de programación de Aplicaciones
BS	Estación Base
BSC	Controlador de Estación Base
BSS	Conjunto de Servicio básico
CDMA	Acceso Múltiple por División de Código
DHCP	Protocolo de configuración de Host Dinámico
DS	Sistema de Distribución
DSP	Procesador digital de señal
DSSS	Espectro ensanchado por secuencia directa
ELF	Frecuencia extremadamente baja
ESS	Conjunto de Servicio Extendido
EST	Estación
ETSI	Instituto Europeo de los Estándares de las Telecomunicaciones
GPRS	General Packet Radio Service
GPS	Sistema de Posicionamiento Global
GSM	Sistema Global para las Comunicaciones Móviles
IEEE	Instituto de Ingenieros en Electricidad y Electrónica
IP	Protocolo de Internet
LAN	Red de Área Local
PCI	Interconexión de Componentes Periféricos
MAN	Red de Área Metropolitana
MAC	Control de Acceso al Medio
MID	Dispositivos de Internet Móvil
MS	Estación Móvil
MSC	Centro de Conmutación Móvil
NIC	Controlador de Interfaz de Red
NSS	Subsistema de Conmutación y Red

OS	Sistema Operativo
PDA	Asistente Personal Digital
RF	Radio Frecuencia
SD	Sistema de Distribución
SD/MMC	Secure Digital / Multi Media Card
SIM	Módulo de Identidad del Suscriptor
SSID	Identificador del conjunto de servicio
TDMA	Acceso Múltiple por División de Tiempo
UHF	Frecuencia Ultra Alta
UMTS	Sistema Universal de Telecomunicaciones Móviles
WAN	Red de Área Extendida
WiMax	Interoperabilidad Mundial para Acceso con Microondas
WLAN	LAN inalámbrica
WMAN	MAN inalámbrica
WPAN	Red de Área Personal Inalámbrica
WWAN	WAN Inalámbrica

Abreviaciones en ingles

Abreviación	Descripción
ARM	Advanced RISC Machines
EDGE	Enhanced Data Rates for GSM Evolution
HIPERLAN	High Performance Radio Local Area Network
RISC	Reduced Instruction Set Computer
SDK	Software Development Kit
SoC	System on Chip

Términos

Arquitectura Harvard: Tipo de arquitectura que utiliza dispositivos de almacenamiento de instrucciones y de datos separados.

Cargador de Arranque: Programa sencillo diseñado exclusivamente para preparar todo lo que necesita el sistema operativo para funcionar. Un cargador embebido mínimo suministra características, como: inicializar el hardware. Suministra parámetros de inicio para el kernel de Linux y empieza el kernel de Linux

Comunicaciones Móviles: El uso de la tecnología que nos permite estar en diferentes lugares sin el uso de cables.

Estándar AMBA-AHB: Esquema de buses que permite realizar transferencias de datos entre módulos del sistema sin la intervención de la CPU en el flujo de datos.

Framework: Estructura conceptual y tecnológica de soporte definida, con módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Half Dúplex: Modo de comunicación donde la comunicación se puede realizar en los dos sentidos pero no al mismo tiempo.

Kernel de Linux: Es el encargado de que el software y hardware del ordenador puedan trabajar juntos. Las funciones más importantes son: Administración de memoria para los programas y procesos, Administración del tiempo de procesador que los programas y procesos utilizan y es el encargado de que podamos acceder a los periféricos de nuestro ordenador.

Middleware: Software que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones sobre plataformas heterogéneas.

Plug-in: Conjunto de componentes de software que agregan capacidades especificas a una aplicación de software.

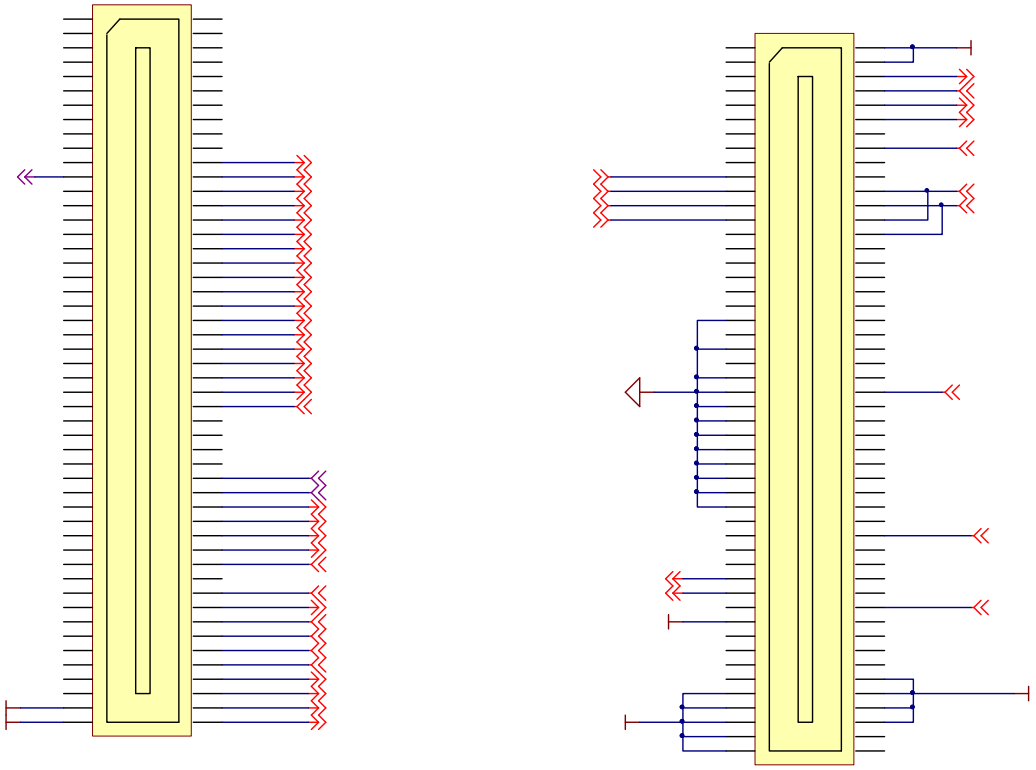
Red Ethernet: Topología de red de Bus donde todos los computadores están conectados por un cable de alta velocidad (de hasta 100Mbps).

Sistema Embebido: Sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas frecuentemente en un sistema de

computación en tiempo real. En un sistema embebido la mayoría de los componentes se encuentran incluidos en una placa base.

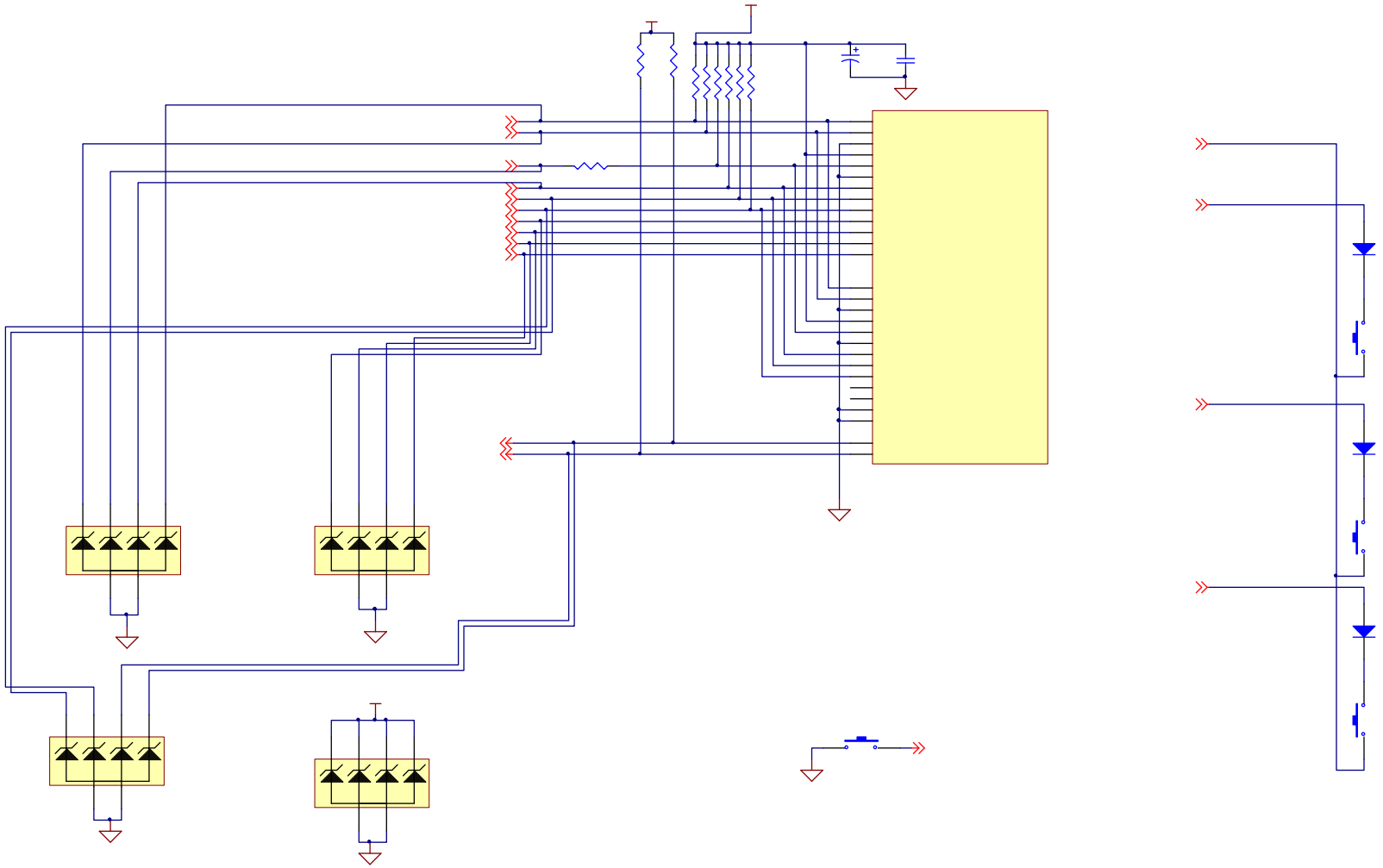
ANEXO 1

Circuito esquemático del módulo del procesador.



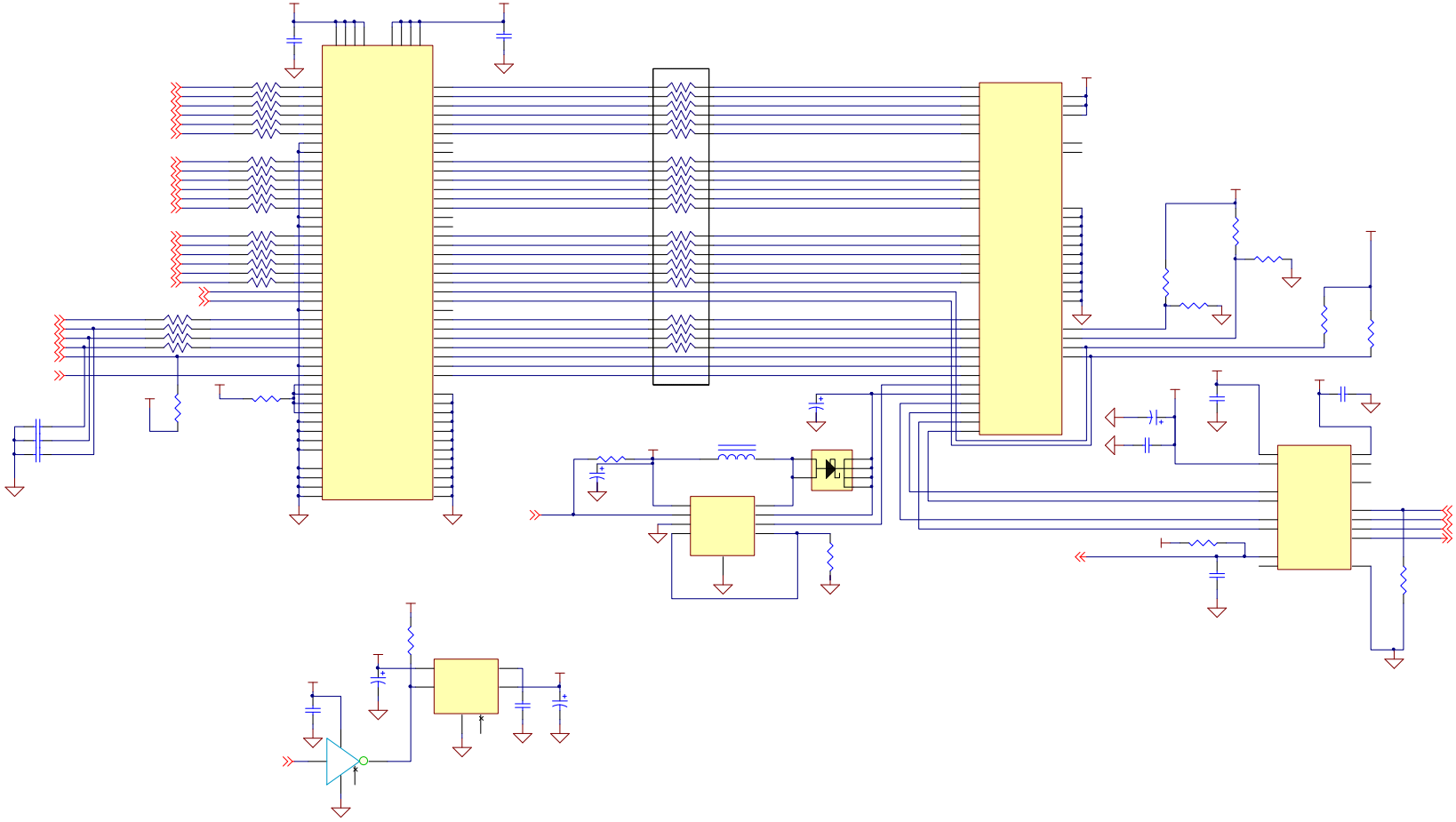
ANEXO 2

Circuito esquemático del módulo del conector SD/MMC.



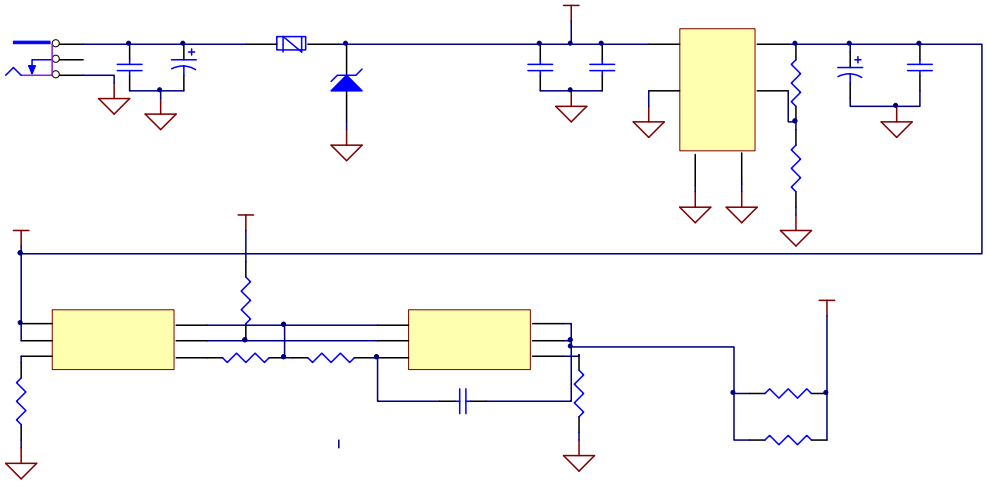
ANEXO 3

Circuito esquemático del módulo del LCD.



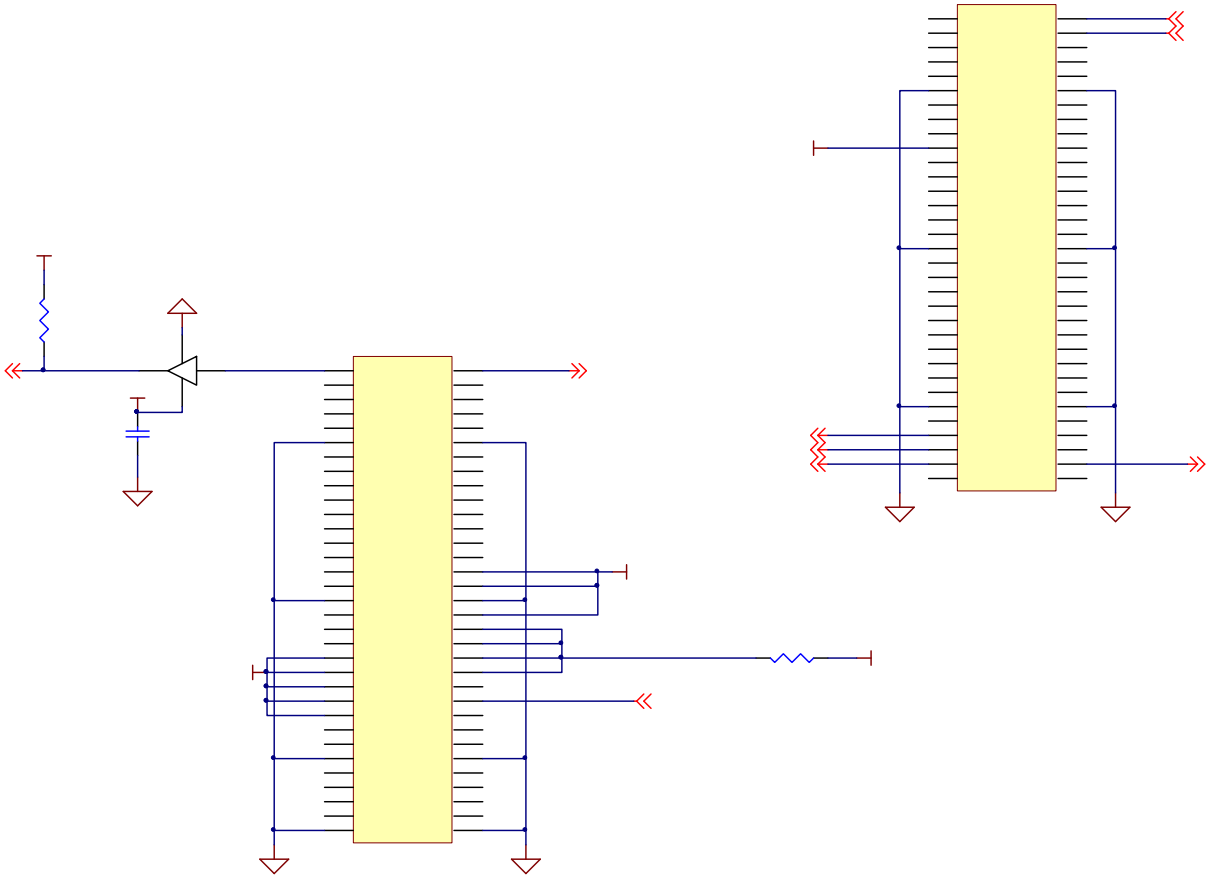
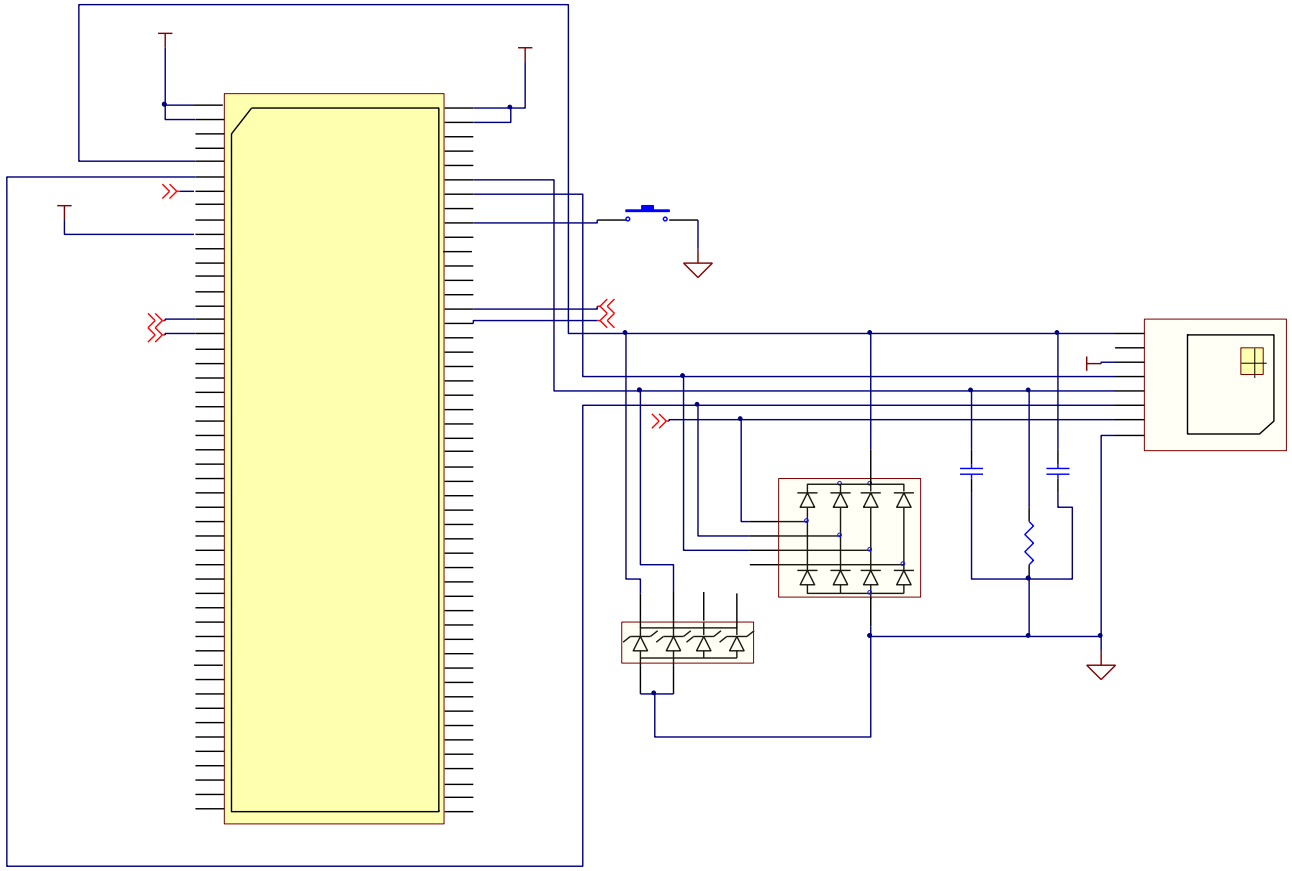
ANEXO 4

Circuito esquemático del módulo de distribución de potencia.



ANEXO 5

Circuito esquemático del módulo de los conectores TWL4030.



ANEXO 6
Circuito esquemático del módulo GSM/GPRS.