



Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ingeniería en Ciencias y Sistemas

## **SISTEMAS DE SERVICIO MÓVIL BASADOS EN MENSAJERÍA DE TEXTO SMS**

**Gabriel Alejandro Díaz López**

Asesorado por la Inga. Alejandra María Escobar

Guatemala, abril de 2011



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**SISTEMAS DE SERVICIO MÓVIL BASADOS EN MENSAJERÍA DE TEXTO  
SMS**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**GABRIEL ALEJANDRO DÍAZ LÓPEZ**  
ASESORADO POR LA INGA. ALEJANDRA MARÍA ESCOBAR

AL CONFERÍRSELE EL TÍTULO DE  
**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, ABRIL DE 2011



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Luis Pedro Ortiz de León
VOCAL V	P. A. José Alfredo Ortiz Herincx
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Pedro Pablo Hernández
EXAMINADOR	Ing. Juan Álvaro Díaz Ardavin
EXAMINADOR	Ing. Cesar Batz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez



## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **SISTEMAS DE SERVICIO MÓVIL BASADOS EN MENSAJERÍA DE TEXTO SMS**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha enero de 2010.

Gabriel Alejandro Díaz López





## **ACTO QUE DEDICO A:**

- Dios** Con tu amor infinito me has hecho sentir parte de tu armonía perfecta, gracias porque siempre estás a mi lado colmándome de bendiciones, gracias padre por permitirme compartir esta alegría con las personas que amo.
- Mis padres**  
**Julio Cesar López**  
**y Dilia López** Mi vida es una secuencia de recuerdos hermosos gracias a ustedes, siempre han sido mi guía, mis mentores, mis amigos y mis pilares, en ustedes he encontrado siempre un amor incondicional y un ejemplo de perseverancia y entrega. Este triunfo es tanto mío como de ustedes, los amo.
- Mis hermanos**  
**Bryan Antonio y**  
**Luis Fernando** Gracias por siempre demostrarme su amor fraternal y apoyo, siempre me han hecho sentir orgulloso de tenerlos como hermanos, ruego al creador que siempre nos mantenga unidos y en paz.
- Mis abuelos**  
**Felipa Marroquín y**  
**Francisco López** Gracias por siempre apoyarme y demostrarme ese calor especial que sólo la familia puede hacer sentir.



# ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
LISTA DE SÍMBOLOS .....	XI
GLOSARIO .....	XIII
RESUMEN.....	XXXI
OBJETIVOS.....	XXXIII
INTRODUCCIÓN.....	XXXV
1. INTRODUCCIÓN A LAS APLICACIONES MÓVILES Y SERVICIOS SMS.....	1
1.1. Evolución del <i>software</i> móvil.....	1
1.2. Segmentos de mercado.....	2
1.2.1. Aplicaciones orientadas a servicios de localización .....	3
1.2.2. Industria del entretenimiento.....	5
1.2.3. Codificación / Decodificación de imágenes.....	6
1.2.4. Servicios de consulta y transacciones .....	8
1.3. Proveedor de servicios.....	10
1.4. Conexiones disponibles Móvil-Servidor .....	11
1.4.1. <i>Internet</i> .....	12
1.4.2. <i>Bluetooth</i> .....	12
1.4.3. Infrarrojo ( <i>Socket</i> ) .....	13
1.4.4. Mensajería de Texto .....	13
2. APLICACIONES MÓVILES BASADAS EN MENSAJERÍA DE TEXTO ....	15
2.1. Mensajería de texto, <i>Short Message Service</i> SMS.....	15
2.1.1. Qué es SMS.....	15

2.1.2.	Características de SMS .....	16
2.1.2.1.	Los mensajes SMS se pueden enviar y leer en cualquier momento .....	17
2.1.2.2.	Los mensajes SMS pueden ser enviados a un teléfono móvil fuera de línea .....	17
2.1.2.3.	Los Mensajes SMS son menos molestos y permiten mantenerlo en contacto .....	17
2.1.2.4.	Los SMS son soportados por el 100% de los teléfonos GSM y se pueden intercambiar independientemente del operador .....	18
2.1.2.5.	SMS es una tecnología apropiada para aplicaciones Inalámbricas .....	18
2.2.	Ejemplo de aplicaciones de mensajería SMS .....	19
2.2.1.	Mensajes de texto de persona a persona .....	19
2.2.2.	Suministro de información .....	20
2.2.3.	Descarga de contenido .....	21
2.2.4.	Alertas y notificaciones de correo electrónico, fax y mensajes de voz .....	21
2.2.5.	Alertas de transacciones de tarjeta de crédito y <i>e-Commerce</i> .....	22
2.2.6.	Sistema de monitoreo remoto .....	23
2.2.7.	<i>Marketing SMS</i> .....	23
2.3.	<i>Short Message Service Central SMSC</i> .....	24
2.3.1.	El centro de mensajes SMS .....	24
2.4.	Opciones SMS / SMSC para el tratamiento de los mensajes .....	25
2.4.1.	Período de validez de un mensaje texto .....	25
2.4.2.	Informe de envío .....	26
2.5.	Proceso de transmisión de un SMS entre terminales .....	28

2.5.1.	Mensajes SMS Intra Operador.....	28
2.5.2.	Proceso de transmisión de mensajes SMS intra operador.....	29
2.5.3.	Mensajes SMS entre operadores.....	30
2.5.4.	Proceso de transmisión de mensajes SMS entre operador.....	30
2.5.5.	Mensajes de texto SMS internacionales .....	32
2.6.	Enviar y recibir mensajes de texto haciendo uso de una computadora .....	33
2.6.1.	Envío de mensajes SMS desde un ordenador .....	33
2.6.2.	Envío de mensajes SMS desde un ordenador mediante un teléfono móvil o <i>módem</i> GSM .....	33
2.6.3.	Envío de mensajes de texto desde una computadora a través de una conexión directa con el SMSC o SMS <i>Gateway</i> de una operadora de telefonía móvil .....	37
2.6.3.1.	Dificultades para conseguir una conexión directa a un SMSC o al SMS <i>Gateway</i> de una operadora de telefonía móvil .....	38
2.6.3.2.	Proveedores de servicios de SMS (Proveedores de <i>Gateway</i> SMS, Distribuidores SMS, SMS <i>Brokers</i> ) .....	39
2.6.4.	Recepción de mensajes SMS usando un ordenador .....	42
2.6.4.1.	Utilización de una PC para recibir mensajes SMS conectando un teléfono móvil o <i>módem</i> GSM/GPRS .....	42
2.6.4.2.	Usar un ordenador para recibir mensajes SMS a través de una conexión directa con el SMSC o SMS <i>Gateway</i> de una operadora de telefonía móvil. ....	43

2.6.4.3.	Usar un ordenador para recibir mensajes SMS a través de una conexión al <i>Gateway</i> SMS de un proveedor de servicios de SMS .....	44
3.	SERVIDORES SMS .....	47
3.1.	<i>Gateway</i> de mensajería SMS .....	47
3.2.	<i>Gateway</i> de mensajería SMS a IP .....	48
3.3.	Aplicaciones del <i>Gateway</i> SMS .....	49
3.4.	Como crear la conexión .....	53
3.5.	El <i>Gateway</i> y su relación con los SMSC y las aplicaciones de mensajería SMS .....	57
3.6.	Algunos proyectos SMS <i>Gateway</i> de fuente abierta.....	61
3.6.1.	<i>Kannel</i> .....	61
3.6.2.	<i>Alamín</i> .....	63
3.6.3.	<i>SMSTLink</i> .....	65
3.6.4.	<i>SMSTools</i> .....	66
4.	PROGRAMACIÓN EN DISPOSITIVOS MÓVIL .....	67
4.1.	Principales Sistemas operativos móvil.....	67
4.1.1.	<i>Symbian</i> .....	70
4.1.1.1.	Componentes de la arquitectura <i>Symbian</i> .....	72
4.1.2.	<i>Android</i> .....	74
4.1.2.1.	Componentes de la arquitectura <i>Android</i> .....	75
4.1.3.	<i>BlackBerry</i> .....	78
4.1.4.	<i>Limo</i> .....	79
4.1.5.	<i>Moblin</i> .....	80
4.1.6.	OS X / <i>iPhone</i> .....	82
4.1.7.	<i>Windows Mobile</i> .....	85
4.1.8.	Tabla de Resumen de SO Móvil .....	89

4.2	La plataforma <i>Java Micro Edition</i> J2ME / <i>Java ME</i> .....	90
4.2.1.	<i>Java Micro Edition</i> .....	92
4.2.2.	Componentes de JME.....	95
4.2.3.	Máquinas Virtuales JME .....	96
4.2.3.1.	<i>KVM Kilobyte Virtual Machine</i> .....	97
4.2.3.2.	<i>CVM Compact Virtual Machine</i> .....	101
4.2.4.	Configuraciones .....	102
4.2.4.1.	Configuración de dispositivos con conexión CDC ( <i>Connected Limited Configuration</i> ) .....	102
4.2.4.2.	Configuración de dispositivos limitados con conexión CLDC ( <i>Connected Limited Device Configuration</i> ) .....	103
5.	IMPLEMENTACIÓN DEL SISTEMA MÓVIL Y SERVIDOR SMS .....	105
5.1.	Desarrollo en el Cliente.....	105
5.1.1.	Realizar la idea en J2ME .....	105
5.1.2.	Prerrequisitos para la implementación en <i>Java ME</i> .....	109
5.1.3.	Implementar la aplicación.....	111
5.1.4.	Probar .....	114
5.1.5.	Compatibilidad, Cubrir todos los equipos.....	115
5.1.6.	Certificar la aplicación .....	117
5.1.7.	Distribuir la aplicación <i>Java ME</i> .....	118
5.2.	Diagrama de aplicación "Confirmación Pago de Boletas de Inscripción" .....	120
5.2.1.	Caso de uso extendido "Confirmación Pago de Boletas de Inscripción" .....	121
5.3.	Servicios del proveedor de contenido .....	124
5.3.1.	Instalación del <i>Gateway SMS Kannel</i> .....	126
5.3.2.	Configuración de <i>Kannel</i> como un <i>Gateway SMS</i> .....	129

5.3.3.	Ejecución de <i>Kannel</i> como <i>Gateway</i> SMS.....	141
5.3.4.	Enviar mensajes usando la interfaz HTTP.....	142
5.4.	Crear los servicios del proveedor de contenido.....	142
5.4.1.	Elegir el servidor <i>web</i> .....	145
5.4.1.1.	Base de Datos .....	147
5.5.	Ciclo del HTTP <i>Server</i> .....	147
CONCLUSIONES.....		151
RECOMENDACIONES .....		153
BIBLIOGRAFÍA.....		155
APÉNDICE .....		157



## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Teléfonos celulares de Gama Alta .....	2
2.	Localización por método de triangulación .....	4
3.	Publicidad de <i>Nokia</i> para la industria del entretenimiento.....	6
4.	Publicidad japonesa por medio de códigos de barra en 3D .....	7
5.	Servicios de consulta de información.....	9
6.	Flujo de envío y recepción de mensajes de texto SMS .....	25
7.	Proceso de transmisión de mensajes Intra Operador .....	30
8.	Proceso de transmisión de mensajes Entre Operador.....	32
9.	Flujo de transmisión de SMS de la red Móvil hacia la red IP .....	48
10.	Esquema general de una <i>Gateway</i> SMS .....	55
11.	Diagrama de arquitectura uniendo elementos conocidos .....	57
12.	Participación del <i>Gateway</i> .....	58
13.	Aplicación de mensajería de texto sin usar un <i>Gateway</i> .....	59
14.	Aplicación de mensajería de que utiliza un <i>Gateway</i> .....	60
15.	Arquitectura del <i>Gateway</i> Kannel.....	62
16.	Arquitectura del <i>Gateway</i> Alamin.....	64
17.	Miembros del consorcio <i>Symbian</i> .....	71
18.	Arquitectura del sistema <i>Symbian</i> .....	73
19.	Arquitectura del sistema <i>Android</i> .....	77
20.	Arquitectura del sistema <i>Limo</i> .....	80
21.	Arquitectura del sistema <i>Moblin</i> .....	82
22.	Arquitectura de la plataforma <i>Java ME</i> .....	93
23.	Entorno de ejecución JME .....	97

24.	Proceso de desarrollo y despliegue de aplicaciones JME .....	100
25.	Componentes de descarga <i>NetBeans</i> .....	110
26.	Logo J2ME <i>Polish</i> .....	110
27.	Crear proyecto J2ME en <i>Netbeans</i> .....	112
28.	<i>Editor Model Driven Architecture</i> de <i>Netbeans</i> .....	113
29.	Código de importación de paquetes .....	113
30.	Ambiente de diseño de componentes LCDUI en <i>Netbeans</i> .....	114
31.	MDA de <i>Netbeans</i> con los componentes de la aplicación móvil.....	120
32.	Menú principal desde el editor de componentes LCDUI.....	121
33.	Solicitud de datos de boleta desde el editor de componentes LCDUI ...	122
34.	Código para capturar el ingreso de usuario.....	122
35.	Ingreso de datos para confirmar el pago de boleta desde el editor de componentes LCDUI .....	123
36.	Configuración de variables del grupo Core.....	131
37.	Configuración de variables del grupo SMSC .....	133
38.	Configuración de variables del grupo SMSBOX .....	136
39.	Configuración de variables del grupo SENDSMS-USER.....	137
40.	Configuración de variables del grupo SMS-SERVICE.....	140
41.	Código fuente de captura del texto a enviar por la aplicación .....	140
42.	Variable Post-URL .....	140
43.	Configuración de servicio <i>Default</i> .....	141
44.	Variable Post-URL con los parámetros sustituidos por valores .....	143
45.	Código fuente de la conexión HTTP por medio de la librería HTTP_CLIENT .....	143
46.	Código fuente de la salida Estándar .....	144
47.	Configuración de la variable <i>Max-Messages</i> .....	145
48.	Código fuente de recepción de parámetros.....	148
49.	Código fuente del proceso de conexión.....	148
50.	Código fuente de la consulta SQL .....	148

51.	Código fuente de la lectura de registros de SQL .....	149
52.	Código fuente del despliegue de la respuesta vía salida Estándar .....	149

## **TABLAS**

I.	Comandos AT relacionados a SMS .....	35
II.	Sistemas operativos para teléfonos Móviles .....	89
III.	Comparación de costos sistema propuesto contra sistema actual .....	107



## LISTA DE SÍMBOLOS

Símbolo	Descripción
#	Almohadilla, indica que el texto que le precede no es tomado en cuenta en la lectura del compilador del lenguaje C y C++.
\$	Apertura de la instrucción de línea de comando del sistema operativo <i>Linux</i> .
%	Cualquier carácter antes de o después de una palabra, en algunos lenguajes es utilizado para hacer búsquedas.
?	Delimita el nombre de una página <i>web</i> dinámica del inicio de los parámetros de ejecución en una URL.
&	Delimita el listado de parámetros recibidos por una página <i>web</i> en una URL.



## GLOSARIO

- 3GPP** *3rd Generation Partnership Project (3GPP)* es un acuerdo de colaboración en tecnología de telefonía móvil, que fue establecido en diciembre de 1998. Esta cooperación es entre ETSI (Europa), ARIB/TTC (Japón), CCSA (China), ATIS (América del Norte) y TTA (Corea del Sur).
- Ancho de Banda** Ancho de banda puede referirse a la capacidad de ancho de banda o ancho de banda disponible en *bit/s*, lo cual típicamente significa el rango neto de *bits* o la máxima salida de una huella de comunicación lógico o físico en un sistema de comunicación digital. La razón de este uso es que de acuerdo a la Ley de *Hartley*, el rango máximo de transferencia de datos de un enlace físico de comunicación es proporcional a su ancho de banda (procesamiento de señal) en *hertz*, la cual es a veces llamada "ancho de banda análogo" en la literatura de la especialidad.
- Aplicación WEB** En la ingeniería de *Software* se denomina aplicación *web* a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor *web* a través de *Internet* o de una *intranet*, mediante un navegador. En otras palabras, es una aplicación *software* que se codifica en un lenguaje soportado por los navegadores *web* (HTML, *Java Script*, *Java*, *Asp.net*, etc.) en la que se confía la ejecución al navegador.

- API** Una interfaz de programación de aplicaciones o API (del inglés *Application programming interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción. Usados generalmente en las bibliotecas.
- Base de Datos** Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
- Binario** El sistema binario, en matemáticas e informática, es un sistema de numeración en el que los números se representan utilizando solamente las cifras cero y uno (0 y 1). Es el que se utiliza en las computadoras, pues trabajan internamente con dos niveles de voltaje, por lo que su sistema de numeración natural es el sistema binario (encendido 1, apagado 0).
- Broker** Corredor o agente es el término en idioma castellano para definir al *broker* (en idioma anglosajón), es decir, el individuo o firma que actúa como intermediario entre un comprador y un vendedor, usualmente cobrando una comisión. El oficio de agente normalmente requiere de una licencia para ejercer. Puede ser de seguros, energía, bienes raíces, aduanas, financiero, de cambio o de bolsas. Además se encarga de asesorar y aconsejar sobre temas financieros, aduanales o relacionados con el negocio.



- Bluetooth** *Bluetooth* es una especificación industrial para Redes Inalámbricas de Área Personal (WPANs) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.
- Comando AT** Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal *modem*. Los comandos AT se denominan así por la abreviatura de *attention*.
- CDMA** La multiplexación por división de código, acceso múltiple por división de código o CDMA (del inglés *Code Division Multiple Access*) es un término genérico para varios métodos de multiplexación o control de acceso al medio basados en la tecnología de espectro expandido.
- CSS** Las hojas de estilo en cascada (en inglés *Cascading Style Sheets*), CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (*World Wide Web Consortium*) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

- Directorio** Un directorio es un conjunto de objetos con atributos organizados en una manera lógica y jerárquica. El ejemplo más común es el directorio telefónico, que consiste en una serie de nombres (personas u organizaciones) que están ordenados alfabéticamente, con cada nombre teniendo una dirección y un número de teléfono adjuntos.
- Domótica** El hogar digital es una vivienda que a través de equipos y sistemas, y la integración tecnológica entre ellos, gracias a la domótica, ofrece a sus habitantes funciones y servicios que facilitan la gestión y el mantenimiento del hogar, aumentan la seguridad; incrementan el *confort*; mejoran las telecomunicaciones; ahorran energía, costes y tiempo, y ofrecen nuevas formas de entretenimiento, ocio y otros servicios dentro de la misma y su entorno sin afectar a las casas normales.
- Enlace** Es un navegador *web* de código abierto en modo texto y gráfico a partir de su versión 2 en modo terminal.
- ETSI** *European Telecommunications Standards Institute* (ETSI) o Instituto Europeo de Normas de Telecomunicaciones es una organización de estandarización de la industria de las telecomunicaciones (fabricantes de equipos y operadores de redes) de Europa, con proyección mundial. El ETSI ha tenido gran éxito al estandarizar el sistema de telefonía móvil GSM.

- Firewall** Un muro de fuego (*firewall* en inglés) es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones.
- Framework** La palabra inglesa *framework* define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.
- FTP** FTP (sigla en inglés de *File Transfer Protocol* - Protocolo de Transferencia de Archivos) en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (*Transmission Control Protocol*), basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.
- Giga bytes** Un *gigabyte* es una unidad de medida informática cuyo símbolo es el GB, y puede equivalerse a 230 *bytes* o a 109 *bytes*, según el uso.

**GNU GPL**

La Licencia Pública General de GNU o más conocida por su nombre en inglés “*GNU General Public License*” o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la *Free Software Foundation* a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de *software*. Su propósito es declarar que el *software* cubierto por esta licencia es *software* libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**GPRS**

*General Packet Radio Service* (GPRS) o servicio general de paquetes vía radio es una extensión del Sistema Global para Comunicaciones Móviles (*Global System for Mobile Communications* o GSM) para la transmisión de datos no conmutada (o por paquetes).

**GPS**

El GPS (*Global Positioning System*: sistema de posicionamiento global) o NAVSTAR-GPS1 es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. Aunque su invención se atribuye al gobierno francés y belga, el sistema fue desarrollado, instalado y actualmente operado por el Departamento de Defensa de los Estados Unidos.

- GSM** El Sistema Global para las Comunicaciones Móviles (GSM, proviene de "*Groupe Special Mobile*") es un sistema estándar, completamente definido, para la comunicación mediante teléfonos móviles que incorporan tecnología digital. Por ser digital cualquier cliente de GSM puede conectarse a través de su teléfono con su computador y puede hacer, enviar y recibir mensajes por *e-mail*, faxes, navegar por *Internet*, acceso seguro a la red informática de una compañía (*LAN/Intranet*), así como utilizar otras funciones digitales de transmisión de datos, incluyendo el Servicio de Mensajes Cortos (SMS) o mensajes de texto.
- HTML** Siglas de *HyperText Markup Language* (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas *web*. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.
- HTTP** Define la sintaxis y la semántica que utilizan los elementos *software* de la arquitectura *web* (clientes, servidores, *proxies*) para comunicarse, las siglas significan "*Hyper Text Transfer Protocol*" que es el protocolo de transferencia de datos.
- IDE** Un entorno de desarrollo integrado o IDE (acrónimo en inglés de *integrated development environment*), es un programa informático compuesto por un conjunto de herramientas de programación.

- Infrarojo** Las redes por infrarrojos permiten la comunicación entre dos nodos, usando una serie de *leds* infrarrojos para ello. Se trata de emisores/receptores de las ondas infrarrojas entre ambos dispositivos, cada dispositivo necesita "ver" al otro para realizar la comunicación por ello es escasa su utilización a gran escala.
- Internet** Es un conjunto descentralizado de redes de comunicación interconectadas, que utiliza la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen, funcionen como una red lógica única, de alcance mundial.
- Java** *Java* es un lenguaje de programación orientado a objetos desarrollado por *Sun Microsystems* a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.
- Kbit** Un *kilobit* es una unidad de medida de información (abreviado kb o *kbit*). En la práctica la unidad *kilobit* se usa para medir el tráfico de la información por un canal digital, y se representa como *kilobits/segundo* esta unidad representa la cantidad de *bits* que se transfieren de un punto a otro en un segundo.

<b>Kernel</b>	En informática, el núcleo (también conocido en español con el anglicismo <i>kernel</i> , de la raíz germánica <i>Kern</i> ) es la parte fundamental de un sistema operativo. Es el <i>software</i> responsable de facilitar a los distintos programas acceso seguro al <i>hardware</i> de la computadora o en forma más básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema.
<b>Linux</b>	GNU/ <i>Linux</i> es uno de los términos empleados para referirse al sistema operativo libre similar a <i>Unix</i> que utiliza el núcleo <i>Linux</i> y herramientas de sistema GNU.
<b>M2M</b>	M2M ( <i>Machine to Machine</i> o Máquina a Máquina) es un concepto genérico que indica el intercambio de información en formato de datos entre dos máquinas remotas.
<b>MMS</b>	<i>Multimedia Messaging System</i> (MMS) o sistema de mensajería multimedia es un estándar de mensajería que le permite a los teléfonos móviles enviar y recibir contenidos multimedia, incorporando sonido, video, fotos o cualquier otro contenido disponible en el futuro.
<b>Módem</b>	Un módem es un dispositivo que sirve para enviar una señal llamada moduladora mediante otra señal llamada portadora.
<b>Modelo pull</b>	Modelo de acceso a la información contenida en la red según el cual es el usuario quien se encarga de obtenerla, buscando por sus propios medios y sin ayuda de aplicaciones especializadas tipo agente.

**MP3** Calidad/tamaño. MPEG-1 Audio Layer 3, más conocido como MP3, es un formato de audio digital comprimido con pérdida desarrollado por el *Moving Picture Experts Group* (MPEG) para formar parte de la versión 1 (y posteriormente ampliado en la versión 2) del formato de vídeo MPEG. El mp3 estándar es de 44 kHz y un *bitrate* de 128 Kbps (calidad CD) por la relación de su nombre es el acrónimo de MPEG-1 *Audio Layer* 3 y el término no se debe confundir con el de reproductor MP3.

**MySql** Es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

**PayPal** *PayPal* es una empresa estadounidense perteneciente al sector del comercio electrónico por *Internet* que permite la transferencia de dinero entre usuarios que tengan correo electrónico, una alternativa al tradicional método en papel como los cheques o giros postales. *PayPal* también procesa peticiones de pago en comercio electrónico y otros servicios *webs*, por los que cobra un porcentaje. La mayor parte de su clientela proviene del sitio de subastas en línea *eBay*.

**PDA** PDA, del inglés *Personal digital assistant* (asistente digital personal), es un computador de mano originalmente diseñado como agenda electrónica (calendario, lista de contactos, bloc de notas y recordatorios).



- PHP** Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas *web* dinámicas.
- Ping*** La utilidad *ping* comprueba el estado de la conexión con uno o varios equipos remotos por medio de los paquetes de solicitud de eco y de respuesta de eco (ambos definidos en el protocolo de red ICMP) para determinar si un sistema IP específico es accesible en una red. Es útil para diagnosticar los errores en redes o enrutadores IP.
- Protocolo** En informática, un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red. Un protocolo es una convención o estándar que controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales. En su forma más simple, un protocolo puede ser definido como las reglas que dominan la sintaxis, semántica y sincronización de la comunicación. Los protocolos pueden ser implementados por *hardware*, *software*, o una combinación de ambos. A su más bajo nivel, un protocolo define el comportamiento de una conexión de *hardware*.
- SavaJe*** *SavaJe* fue el desarrollador del sistema operativo *SavaJe*, un *Java* del sistema operativo para teléfonos móviles avanzados. El sistema operativo *SavaJe* fue un monolítico sistema operativo y *Java* de la plataforma, una implementación de *Sun Microsystems' Java Standard Edition*, en contraposición a la más limitada *Micro Edition* ofrece habitualmente en los teléfonos móviles.

**SDK** Un *kit* de desarrollo de *software* o SDK (siglas en inglés de *software development kit*) es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de *software*, *frameworks*, plataformas de *hardware*, computadoras, videoconsolas, sistemas operativos, etc.

**SMS** El servicio de mensajes cortos o SMS (*Short message service*) es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto, o más coloquialmente, textos o mensajitos) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano. SMS fue diseñado originariamente como parte del estándar de telefonía móvil digital GSM, pero en la actualidad está disponible en una amplia variedad de redes, incluyendo las redes 3G.

**Server  
apache** Es un servidor *web* HTTP de código abierto para plataformas *Unix*. *Windows*, *Macintosh* y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

- SIM** Una tarjeta SIM (acrónimo en inglés de *Subscriber identity module*, en español módulo de identificación del suscriptor) es una tarjeta inteligente desmontable usada en teléfonos móviles y *módems* USB. Las tarjetas SIM almacenan de forma segura la clave de servicio del suscriptor usada para identificarse ante la red, de forma que sea posible cambiar la línea de un terminal a otro simplemente cambiando la tarjeta.
- Sitio WEB** Es un conjunto de páginas *web*, típicamente comunes a un dominio o subdominio en la *World Wide Web*.
- Smartphone** Un *Smartphone* (teléfono inteligente en español) es un dispositivo electrónico que funciona como un teléfono móvil con características similares a las de un ordenador personal. Casi todos los teléfonos inteligentes son móviles que soportan completamente un cliente de correo electrónico con la funcionalidad completa de un organizador personal. Una característica importante de casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero. El término "Inteligente" hace referencia a cualquier interfaz, como un teclado QWERTY en miniatura, una pantalla táctil, o simplemente el acceso a *internet* y al correo electrónico de una compañía, pagando o personal, gratuito.

**Symbol**

*Symbol* es una empresa adquirida por *Motorola* que comercializa sistemas de radiofrecuencia computarizada, entre otros productos. El uso habitual de este tipo de sistemas es para capturar por medio de su láser, códigos de barras de tiendas de autoservicio, tiendas o negocios que se comunican con el sistema correspondiente, el cual permite administrar listados de precios, así como gestiones de inventarios y almacenes, son prácticas y fáciles de manejar por su lenguaje amistoso con el usuario terminal, además de ser portátiles y muy ligeras.

**Tag HTML**

Una etiqueta o baliza (términos a veces reemplazados por el anglicismo *tag*) es una marca con tipo que delimita una región en los lenguajes basados en XML. También puede referirse a un conjunto de juegos informáticos interactivos que se añade a un elemento de los datos para identificarlo (*Oxford English Dictionary*). Esto ocurre, por ejemplo, en los archivos MP3 que guardan información sobre una canción así como sobre el artista que la ha cantado o compuesto.

## **TCP/IP**

La familia de protocolos de *Internet* es un conjunto de protocolos de red en los que se basa *Internet* y que permiten la transmisión de datos entre redes de computadoras. En ocasiones se le denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de *Internet* (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia.

Existen tantos protocolos en este conjunto que llegan a ser más de 100 diferentes, entre ellos se encuentra el popular HTTP (*HyperText Transfer Protocol*), que es el que se utiliza para acceder a las páginas *web*, además de otros como el ARP (*Address Resolution Protocol*) para la resolución de direcciones, el FTP (*File Transfer Protocol*) para transferencia de archivos, y el SMTP (*Simple Mail Transfer Protocol*) y el POP (*Post Office Protocol*) para correo electrónico, TELNET para acceder a equipos remotos, entre otros.

## **Tecnología *push***

Tecnología *Push*, o servidor *push*, describe un estilo de comunicaciones sobre *Internet* donde la petición de una transacción se origina en el servidor. Por el contrario a la Tecnología *Pull*, donde la petición es originada en el cliente.

## **TDMA**

La multiplexación por división de tiempo (TDM) es una técnica que permite la transmisión de señales digitales y cuya idea consiste en ocupar un canal (normalmente de gran capacidad) de transmisión a partir de distintas fuentes, de esta manera se logra un mejor aprovechamiento del medio de transmisión.

- Telnet**                    Telnet (*TELEcommunication NETwork*) es el nombre de un protocolo de red que sirve para acceder mediante una red a otra máquina para manejarla remotamente como si estuviéramos sentados delante de ella. También es el nombre del programa informático que implementa el cliente. Para que la conexión funcione, como en todos los servicios de *Internet*, la máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones. El puerto que se utiliza generalmente es el número 23.
- Tomcat**                    *Tomcat* (también llamado *Jakarta Tomcat* o *Apache Tomcat*) funciona como un contenedor de *servlets* desarrollado bajo el proyecto *Jakarta* en la *Apache Software Foundation*. *Tomcat* implementa las especificaciones de los *servlets* y de *JavaServer Pages* (JSP) de *Sun Microsystems*.
- UIQ**                         UIQ (*User Interface Quartz*, Interfaz de Usuario Quartz) es una plataforma para terminales móviles desarrollada por UIQ *Technology* basada en el sistema operativo *Symbian*. Se caracteriza por agregar soporte para pantallas táctiles.
- UMTS**                    Sistema Universal de Telecomunicaciones Móviles (*Universal Mobile Telecommunications System* - UMTS) es una de las tecnologías usadas por los móviles de tercera generación (3G, también llamado W-CDMA), sucesora de GSM, debido a que la tecnología GSM propiamente dicha no podía seguir un camino evolutivo para llegar a brindar servicios considerados de Tercera Generación.

- Unicode** El *Standard Unicode* es un *estándar* de codificación de caracteres diseñado para facilitar el tratamiento informático, transmisión y visualización de textos de múltiples lenguajes y disciplinas técnicas además de textos clásicos de lenguas muertas. El término *Unicode* proviene de los tres objetivos perseguidos: universalidad, uniformidad y unicidad.
- vCard** *vCard* es un formato estándar para el intercambio de información personal, específicamente tarjetas personales electrónicas (*electronic business cards*). Las *vCards* son usualmente adjuntadas a mensajes de *e-mail*, pero pueden ser intercambiadas en muchas otras formas, como en la *World Wide Web* o a través de códigos QR. Pueden contener nombre, dirección, números telefónicos, URLs, logos, fotografías, e incluso clips de audio.
- WAP** *Wireless Application Protocol* o WAP (protocolo de aplicaciones inalámbricas) es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas. Se trata de la especificación de un entorno de aplicación y de un conjunto de protocolos de comunicaciones para normalizar el modo en que los dispositivos inalámbricos, se pueden utilizar para acceder a servicios de red.
- Web 2.0** Término acuñado por *Tim O'Reilly* en 2004 para referirse a una segunda generación de *Web* basada en comunidades de usuarios y una gama especial de servicios *web*, como las redes sociales, los *blogs*, los *wikis*, etc.

**Wifi**

Wi-Fi es una marca de la *Wi-Fi Alliance* (anteriormente la WECA: *Wireless Ethernet Compatibility Alliance*), la organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares 802.11 relacionados a redes inalámbricas de área local.

**X.25**

X.25 es un estándar UIT-T para redes de área amplia de conmutación de paquetes. Su protocolo de enlace, LAPB, está basado en el protocolo HDLC (publicado por ISO, y el cual a su vez es una evolución del protocolo SDLC de IBM). Establece mecanismos de direccionamiento entre usuarios, negociación de características de comunicación, técnicas de recuperación de errores. Los servicios públicos de conmutación de paquetes admiten numerosos tipos de estaciones de distintos fabricantes. Por lo tanto, es de la mayor importancia definir la interfaz entre el equipo del usuario final y la red.



## RESUMEN

La constante evolución de las tecnologías de comunicación contribuye a la búsqueda de nuevos métodos de satisfacer las necesidades informáticas de los usuarios, hoy en día no es posible esperar a estar sentado frente a una computadora para realizar nuestro trabajo, para ejecutar una venta, para solicitar información de precios de un producto, realizar un pago, etc.

La sociedad requiere métodos más rápidos y menos costosos para consumir los servicios prestados por las empresas, métodos que requieran menos desplazamiento, gasto de combustible, gasto de recurso humano, reducción en el uso de papel, de impresiones, de tiempo, etc. Como innumerables son los servicios, innumerables son los usuarios y a medida que nuestra economía crece son más los servicios que se adhieren a esta cadena.

En nuestro país, Guatemala y el resto de países centroamericanos se esconde un problema detrás del consumo de los servicios y es el pago de la facturación. Las empresas proveedoras de servicios ofrecen diversas formas de pago, algunas tradicionales y otras modernas. Los métodos tradicionales demandan costos elevados y usualmente estos costos son absorbidos por los clientes, mientras que los métodos modernos, encuentran barreras como la resistencia al cambio, la falta de seguridad, falta de centralización y otros.

Para aminorar estos problemas se propone la utilización de la tecnología celular móvil como método de consulta y método de operación para el pago de servicios, ejemplificándolo con la implementación del proyecto "**Confirmación Pago de Boletas de Inscripción**" el cual pretende ser el equivalente del proceso de pago de la matrícula estudiantil anual actual, con la diferencia que todas las gestiones se harán desde el teléfono móvil del estudiante cumpliendo como objetivo principal la reducción de costos.

Antes de iniciar con la implementación, es necesario conocer las tecnologías tanto de *hardware* como de *software* disponibles para la creación de proyectos de tecnología celular. Más importante aún conocer las bases de SMS y su funcionamiento así como la forma de llevar SMS a la computadora y de vuelta al dispositivo móvil. Dichas arquitecturas se dividen en dos etapas, el desarrollo del lado del cliente (Dispositivo Móvil) y el desarrollo del lado del servidor (unión de la red IP y la red GSM).

## OBJETIVOS

### GENERAL

Describir la tecnología utilizada para el desarrollo de infraestructuras para aplicaciones de negocio y de servicios brindados por terceros, accesibles a través de dispositivos telefónicos móviles utilizando la tecnología SMS.

### ESPECÍFICOS

1. Identificar los elementos de infraestructura básicos (*hardware*) para montar un centro de servicios accesibles a través de dispositivos telefónicos móviles.
2. Descubrir los elementos de *software* involucrados en la arquitectura necesaria para montar un centro de servicios accesibles a través de dispositivos telefónicos móviles
3. Conocer la arquitectura de la red GSM y los métodos disponibles para ampliar dicha red uniéndola a redes IP.
4. Describir los distintos sistemas operativos móviles y lenguajes de programación disponibles para el desarrollo de aplicaciones para teléfonos celulares.

5. Crear y exponer un servicio consumible desde una aplicación móvil.
6. Crear una aplicación móvil capaz de consumir un servicio vía SMS.

## INTRODUCCIÓN

Cada vez es más común ver que el teléfono móvil se use para muchas otras funciones distintas a aquella para la que verdaderamente fue diseñado. El avance de la tecnología ha hecho que estos aparatos incorporen funciones que no hace mucho parecían futuristas, como juegos, reproducción de música MP3 y otros formatos, correo electrónico, SMS, agenda electrónica PDA, fotografía y video digital, video llamada, navegación por *internet* y hasta televisión digital.

Esto ha dado como resultado un mercado emergente de aplicaciones en la telefonía móvil.

Y es que ¿Quién no tiene uno, o incluso varios móviles hoy en día? Con una penetración superior al 100% en Europa, el teléfono móvil se ha convertido en un producto de consumo masivo y además, en el dispositivo más personal que existe, llegando a ser una señal de identidad más del usuario.

Esto, unido a la evolución de las redes de telefonía móvil y a la proliferación de teléfonos con capacidades cada vez más avanzadas, ha hecho que las empresas, en su búsqueda de nuevas formas de aumentar la satisfacción del usuario final, dirijan sus miradas hacia el teléfono móvil.

Esta evolución de la telefonía móvil ha hecho que las nuevas aplicaciones móviles puedan ser una realidad. Hasta ahora, las principales tecnologías utilizadas en el canal móvil (SMS, MMS, WAP) ofrecían una interfaz lenta, gráficamente pobre y con pocas opciones de control e interactividad.

Sin embargo, ya existen soluciones multimedia avanzadas que permiten explotar las posibilidades que brinda la tecnología actual para acceder a un nuevo paradigma de servicios y aplicaciones.

Esta situación está permitiendo acercar al canal móvil, muchos de los servicios y aplicaciones que los usuarios usaban habitualmente en otros canales, mientras demandaban soluciones de mayor movilidad y con un entorno personalizado, intuitivo en su uso y de elevada calidad gráfica.

Sectores como el de la banca o las telecomunicaciones demandan cada vez más este tipo de soluciones multimedia avanzadas, ya que en ellos existe una fuerte competitividad por captar nuevos clientes y retener a los ya existentes. La característica común a todos ellos es la necesidad de ofrecer un servicio diferenciador, ágil y de alta calidad a un gran número de clientes.

# 1. INTRODUCCIÓN A LAS APLICACIONES MÓVILES Y SERVICIOS SMSC

## 1.1. Evolución del *software* móvil

Actualmente el *hardware* de teléfonos móviles está llegando a una etapa de madurez en su desarrollo. El aumento de ventas ha permitido que los precios disminuyan y actualmente estos dispositivos se pueden obtener a precios muy competitivos con prestaciones muy altas. Grandes empresas como *Microsoft, Palm, ASUS, HP, Toshiba, Acer, Nokia, Sony, Ericsson, Motorola*, etc. están apostando por el desarrollo y comercialización de teléfonos móviles cada vez más funcionales y especializados.

Hoy día toca el turno de las aplicaciones móviles, las cuales llegan en un mundo donde realizar la más mínima tarea implica estar conectado o en términos informáticos "en línea". Para ello con tan solo el uso de un celular se pueden realizar proezas que años atrás no era posible concebir.

Por otra parte, la evolución de las redes públicas de datos (GSM (WAP/SMS), GPRS y UMTS), así como los estándares *Bluetooth* y *Wifi* (*Wireless LAN*), permiten que estos dispositivos estén realmente conectados con los sistemas de información de las organizaciones. Los antiguos sistemas de aplicaciones móviles basados en *hardware* industrial (*Symbol / Symbian*) de alto coste, están empezando a ser desplazados por sistemas desarrollados sobre estándares abiertos de propósito general.

Figura 1. **Teléfonos celulares de Gama Alta**



Fuente: Asociación Hispanoamericana de Centros de Investigación, Las telecomunicaciones y la movilidad en la sociedad de la información Pág. 16

## 1.2. Segmentos de mercado

Los dispositivos móviles componen la red física en mayor crecimiento alrededor del mundo. De acuerdo a un análisis llevado a cabo por la compañía *The Mobile World*, el número de usuarios en telefonía móvil a nivel mundial pasará la barrera de los 3 mil millones de usuarios (dato obtenido en 2007), gran parte por la inclusión a estas redes de personas de África y China.



Según los resultados obtenidos, por cada minuto que pasa, se registran unos mil abonados nuevos a los servicios de telefonía móvil alrededor del mundo, lo cual es una cifra impactante.

Se tardó aproximadamente unos 20 años en lograr la cifra de mil millones de usuarios, mientras que en menos de 40 meses se logró llegar a los dos mil millones. Dos años después, en julio de 2007, se lograra la cifra de los tres mil millones revelada por este estudio.

En el último cuatrimestre se han vendido unos 240 millones de teléfonos móviles, y se han dado de alta unas 135 millones de líneas, lo cual nos indica que no todos los móviles vendidos apuntan a nuevas líneas, sino al reemplazo (actualización) que realizan clientes, lo que también marca un gran negocio para empresas que desarrollan estos productos.

Es en este entorno emergente es en donde surgen nuevas áreas de oportunidad:

- Aplicaciones orientadas a servicios de localización
- Industria del entretenimiento
- Codificación / Decodificación de imágenes
- Servicios de consulta y transacciones

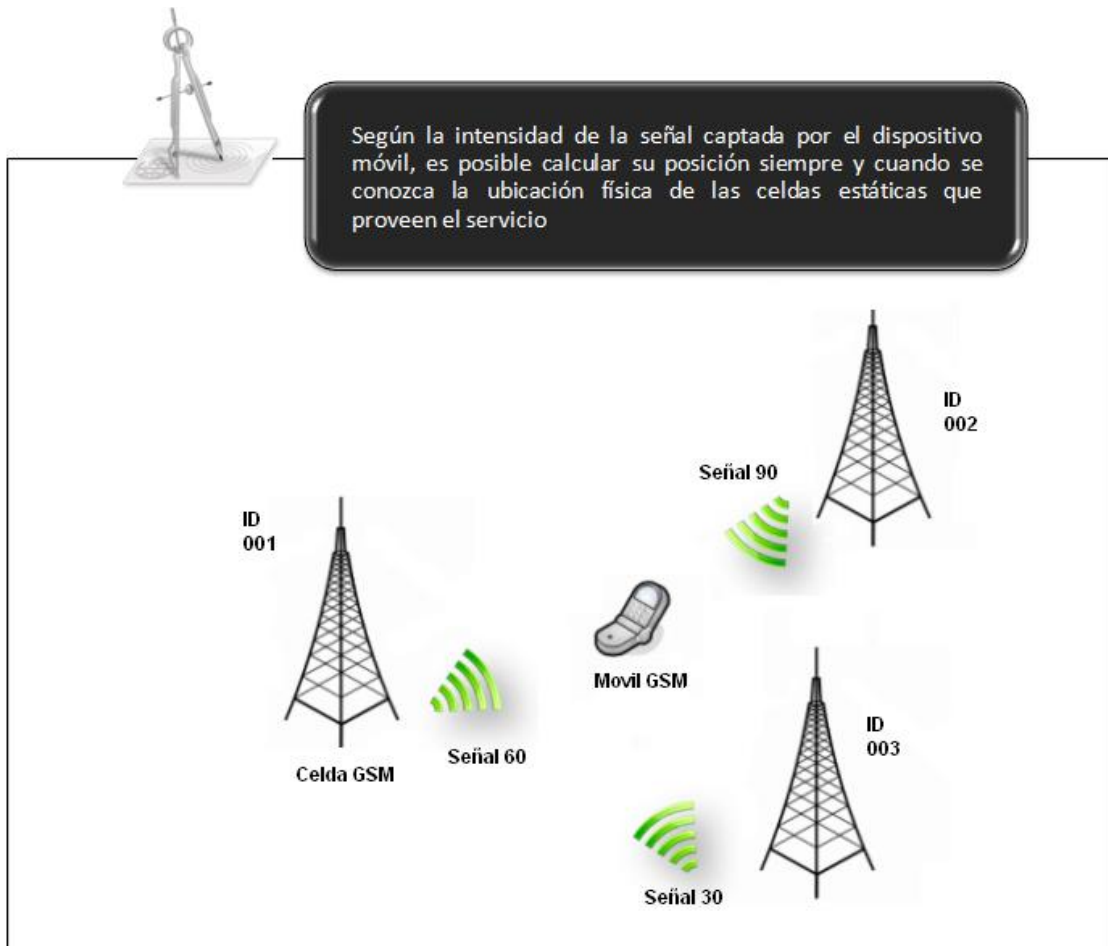
### **1.2.1. Aplicaciones orientadas a servicios de localización**

La localización GSM es un servicio ofrecido por las empresas operadoras de telefonía móvil o bien por terceros que permite determinar, con un cierto rango de precisión, la ubicación física de un terminal móvil determinado.

Las posibilidades de aplicación de esta tecnología permiten a familiares y amigos estar en constante conocimiento de la ubicación de los mismos.

A diferencia del GPS que es una tecnología de alto coste y con dificultades técnicas, la localización por la red GSM puede ser explotable bajo ciertos niveles de consentimiento tanto de los usuarios localizadores como de la privacidad de los usuarios a localizar.

Figura 2. **Localización por método de triangulación**



Fuente: elaboración propia

### **1.2.2. Industria del entretenimiento**

Los juegos para celular son los principales propulsores del mercado de juegos en muchos países, mucho más promisorios que los juegos para PC debido al bajo costo. La facturación de las desarrolladoras de juegos para celular fue de aproximadamente US\$ 30 millones en 2007 y crece en la tasa de los 30% anuales.

Los juegos han representado hasta el 50% del total de las descargas para teléfonos móviles, pero a mediados de 2008 la cuota de juegos móviles alcanzaba solamente el 35% de todo el contenido descargado.

Actualmente, los celulares son tan potentes como cualquier otro aparato de juegos, pero no supera la barrera que poseen juegos diseñados para el control fácil con los botones, como por ejemplo, apretar dos teclas al mismo tiempo, distancia de los dedos etc. Además de eso, la tendencia de aparatos cada vez menores puede interferir en la calidad de la imagen del juego. Los dispositivos también son limitados para imágenes en 3D, reacciones rápidas y sonidos poderosos.

Figura 3. **Publicidad de Nokia para la industria del entretenimiento**



Fuente: *Series 40 Platform: Introductory White Paper v3.3*

[HTTP://www.forum.nokia.com/info/sw.nokia.com/id/78c142ab-99d9-4ce6-b810-3962c7cea3f1/Series\\_40\\_Platform\\_White\\_Paper\\_V3\\_3\\_en.pdf.html](http://www.forum.nokia.com/info/sw.nokia.com/id/78c142ab-99d9-4ce6-b810-3962c7cea3f1/Series_40_Platform_White_Paper_V3_3_en.pdf.html)

### 1.2.3. **Codificación / Decodificación de imágenes**

Este tipo de aplicación brinda la posibilidad de utilizar la cámara del celular como lector o decodificador de códigos de barras, simplemente pasando la cámara del móvil por encima del código que se desea inspeccionar y pulsando el botón para realizar la captura. Con lo que es posible con una simple captura tener acceso a la *web*, ver cualquier tipo de mensajes u obtener un número de teléfono directamente para su utilización.

Otro uso es la decodificación de códigos QR, un código QR es un sistema para almacenar información en una matriz de puntos o un código de barras bidimensional creado por la compañía japonesa *Denso-Wave* en 1994; se caracterizan por los tres cuadrados que se encuentran en las esquinas y que permiten detectar la posición del código al lector.

La sigla "QR" se derivó de la frase inglesa "Quick Response" pues el creador aspiraba a que el código permitiera que su contenido se leyera a alta velocidad. Los códigos QR son muy comunes en Japón y de hecho son el código bidimensional más popular en ese país.

Figura 4. **Publicidad japonesa por medio de códigos de barra en 3D**



Fuente: *Series 40 Platform: Introductory White Paper v3.3*

[HTTP://www.forum.nokia.com/info/sw.nokia.com/id/78c142ab-99d9-4ce6-b810-3962c7cea3f1/Series\\_40\\_Platform\\_White\\_Paper\\_V3\\_3\\_en.pdf.html](http://www.forum.nokia.com/info/sw.nokia.com/id/78c142ab-99d9-4ce6-b810-3962c7cea3f1/Series_40_Platform_White_Paper_V3_3_en.pdf.html)

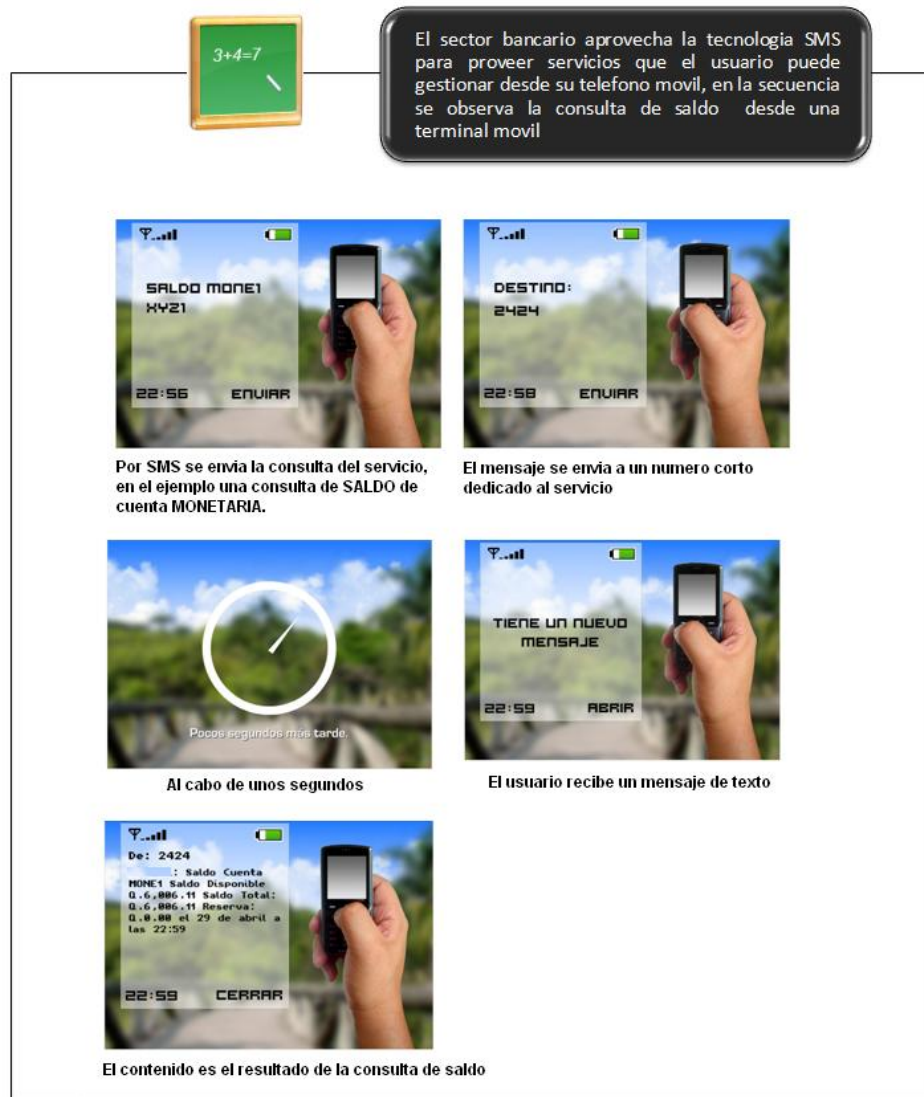
#### **1.2.4. Servicios de consulta y transacciones**

Son servicios enfocados a consultas y transacciones auto gestionadas por el usuario, un ejemplo típico es el sector bancario, el banco puede ofrecer a sus clientes la posibilidad de auto gestionar todas sus operaciones, como la consulta de cuentas corrientes y tarjetas de crédito y débito o la localización de sucursales y cajeros con mapas interactivos, entre otras.

El sector de las telecomunicaciones, asimismo, puede ofrecer a sus clientes la posibilidad de acceder a información sobre los planes de precios, canjeo de puntos, consulta de saldo, recarga de tarjeta, y muchos otros.

Asimismo, este tipo de soluciones consiguen en una optimización de los costes relacionados con la atención al cliente, puesto que se ofrece a los clientes una resolución satisfactoria dentro de la aplicación multimedia a la gran mayoría de consultas que hasta ahora requerían de un operador humano, y además pueden convertirse en un canal efectivo para realizar acciones de venta cruzada de otros productos y servicios de las empresas de telecomunicaciones o entidades bancarias, lo que aumenta el interés de estos sectores por ofrecer soluciones de este tipo.

Figura 5. Servicios de consulta de información



Fuente: elaboración propia

### **1.3. Proveedor de servicios**

Un proveedor de servicios es una empresa que proporciona servicios de mensajería de SMS, MMS o *Internet*, pero no es un operador de red móvil. Actúa como intermediario entre los operadores de redes móviles y usuarios de los servicios de telefonía. Un proveedor de servicio firma diferentes acuerdos con muchos operadores de redes de telefonía móvil para obtener permiso de intercambiar conexiones con sus redes móviles, y de este modo ofrecer una interfaz para que los usuarios puedan enviar y recibir mensajes con los cuales se puedan consumir los servicios. Los proveedores de servicios SMS son también conocidos como *Gateway providers, resellers o broker*.

#### **Proveedor de Gateway**

Un proveedor de servicios *Gateway* proporciona una interfaz donde los usuarios pueden realizar una conexión donde la petición es transformada a una instrucción que permita el consumo de un servicio o bien al enrutamiento del mensaje a una red diferente.

#### ***Resellers o Brokers***

Son entidades que compran cierta cantidad de mensajes SMS a varias compañías de telefonía móvil o bien cierta cantidad de *Megabytes* de descarga que luego venden los mensajes SMS a un precio superior al costo.



Tanto los proveedores de *Gateway* y los *Brokers*, se encargan de lidiar con la labor de enrutamiento y tratamiento de las conexiones realizadas por los usuarios y de devolver el resultado del servicio solicitado, ejemplo de este tipo de transacciones son los números cortos a los cuales se puede enviar un mensaje de texto y recibir a cambio una noticia, un código, participar en una promoción, etc. Usualmente la empresa que renta el servicio únicamente debe de proveer el contenido y los usuarios de consumir el servicio.

En el capítulo 3 se explica con mayor detalle el proveedor de servicios *Gateway* y en el capítulo 5 se explicará cómo configurar y exponer un servicio a través del *Gateway*.

#### **1.4. Conexiones disponibles Móvil-Servidor**

Es difícil imaginar que estas aplicaciones puedan realizarse de forma aislada, al contrario, estas aplicaciones exigen que las terminales móviles posean un método de acceso fuera del dispositivo con el fin de establecer comunicación con un servidor, otros dispositivos o bien cualquier otro aparato con el que sea necesario interactuar con el fin de establecer comunicación.

La telefonía móvil, también llamada telefonía celular, básicamente está formada por dos grandes partes: una red de comunicaciones (o red de telefonía móvil) y los terminales (o teléfonos móviles) que permiten el acceso a dicha red.

Los métodos de transmisión que dispone una terminal móvil son diversos y se mencionan a continuación:

### **1.4.1. *Internet***

Con la aparición de la telefonía móvil digital, fue posible acceder a páginas de *Internet* especialmente diseñadas para móviles (conocido como tecnología WAP).

Posteriormente, nació el GPRS, que permitió acceder a *Internet* a través del protocolo TCP/IP. Mediante el software adecuado es posible acceder, desde un terminal móvil, a servicios como FTP, *Telnet*, mensajería instantánea, correo electrónico, utilizando los mismos protocolos que un ordenador convencional. La velocidad del GPRS es de 54 *kbit/s* en condiciones óptimas, y se tarifa en función de la cantidad de información transmitida y recibida.

El *Internet* puede ser la solución ideal a utilizar para cualquiera de las aplicaciones mencionadas anteriormente para implementar la entrada y salida de información entre el teléfono móvil y cualquier otro aparato conectado a la red en cualquier parte del mundo, sin embargo no hay que olvidar que la red de telefonía móvil es hasta cierto punto independiente de la red *Internet* y más aun hay que tener presente que la interconexión es hecha por medio de un *carrier* el cual cobra honorarios de servicio. Por lo que este sistema aún es caro tomando en cuenta que el sistema de tarificación no es una verdadera tarifa plana sino que establece limitaciones en cuanto a datos o velocidad.

### **1.4.2. *Bluetooth***

*Bluetooth* se denomina al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, con una cobertura baja y basada en transceptores de bajo costo.

Gracias a este protocolo, los dispositivos que lo implementan pueden comunicarse entre ellos cuando se encuentran dentro de su alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen por qué estar alineados, pueden incluso estar en habitaciones separadas si la potencia de transmisión lo permite.

*Bluetooth* es un protocolo que viene implementado en la mayoría de dispositivos, puede ser útil en aplicaciones que no requieran largo alcance por ejemplo algunos juegos.

#### **1.4.3. Infrarrojo (Socket)**

Es un medio de comunicación parecido a *Bluetooth*, permite implementar la conexión tipo *socket* y posee las mismas ventajas y limitantes que *Bluetooth* excepto porque los dispositivos deben estar alineados, esto es una limitante física pero una ventaja en privacidad.

#### **1.4.4. Mensajería de Texto**

La mensajería de texto es formalmente utilizada para la activación de servicios de forma remota. Actualmente es utilizada como un método de reemplazo a la red internet ya que representa un costo mucho menor además de que puede implementarse en cualquier modelo de terminal, sin embargo requiere de la implementación de interfaces *middleware* que permitan la interacción entre la mensajería y los servidores de aplicaciones.

SMS es un medio de comunicación actual y con grandes ventajas:

- Alrededor del mundo se estiman 3.000 millones de terminales móviles que implican una penetración de 44,64% de la población y que continúa creciendo;
- No tiene limitantes geográficas, ya que la cobertura de la red GSM, que es la que soporta el SMS es prácticamente total;
- Gran expansión del SMS en la población principalmente entre los jóvenes; sin embargo en la actualidad el SMS está penetrando cada vez más en los hábitos de las personas y la edad media de los usuario está aumentando progresivamente;
- Los operadores cuentan con sistemas de gestión de colas por lo que nunca se pierden los mensajes;
- Los mensajes son entregados siempre aunque el teléfono esté apagado o fuera de cobertura;
- Se dispone con confirmación de entrega de los mensajes con datos de fecha y hora;
- La comunicación es robusta y confiable puesto que se soporta en redes públicas de telecomunicaciones.

## 2. APLICACIONES MÓVILES BASADAS EN MENSAJERÍA DE TEXTO

### 2.1. Mensajería de texto, *Short Message Service* SMS

#### 2.1.1. Qué es SMS

SMS significa servicio de mensajes cortos. Es una tecnología que permite el envío y recepción de mensajes entre teléfonos móviles. SMS apareció por primera vez en Europa en 1992. Se incluyó desde sus comienzos en el GSM (*Global System for Mobile Communications*). Más tarde fue trasladado a las tecnologías inalámbricas como CDMA y TDMA. Los estándares GSM y SMS fueron desarrollados originalmente por el ETSI. ETSI es la abreviatura de Instituto Europeo de Normas de Telecomunicación. Ahora el 3GPP (*Third Generation Partnership Project*) es responsable del desarrollo y mantenimiento de las normas GSM y SMS.

Tal y como sugiere el nombre de "servicio de mensajes cortos", los datos que puedan ser consideradas por un mensaje SMS es muy limitada. Un mensaje de texto puede contener un máximo de 140 octetos (1120 *bits*) de datos, por lo que un mensaje de texto puede contener un máximo de:

- 160 caracteres, si usa codificación de caracteres de 7 *bits*. (7 *bits* de codificación de caracteres es adecuado para codificar los caracteres latinos como el alfabeto inglés o español).

- 70 caracteres si usa codificación de caracteres de 16-bit *Unicode* (Mensajes de texto que contienen caracteres no latinos como el chino debe utilizar caracteres de 16 *bits* de codificación de caracteres).

Los mensajes de texto SMS son utilizados internacionalmente. Funcionan bien con todos los idiomas soportados por *Unicode*, incluyendo árabe, chino, japonés y coreano.

Además de texto, SMS también puede transportar datos binarios. Es posible enviar tonos de llamada, imágenes, logotipos de operadores, fondos de pantalla, animaciones, tarjetas de visita (por ejemplo, *vCards*) y las configuraciones de WAP a un teléfono móvil con mensajes SMS.

Una de las principales ventajas del SMS es que está incluido en el 100% de teléfonos móviles GSM. Casi todos los planes de suscripción inalámbrica proporcionada por los transportistas de bajo costo incluyen servicio de mensajería SMS. A diferencia de los SMS, las tecnologías móviles como WAP y *Java* móviles no son compatibles en muchos modelos de teléfonos móviles antiguos.

### **2.1.2. Características de SMS**

SMS es un éxito en todo el mundo. El número de mensajes SMS intercambiados cada día es enorme. La mensajería SMS es ahora una de las más importantes fuentes de ingresos de los operadores inalámbricos. ¿Qué es tan especial acerca de SMS que lo hace tan popular en todo el mundo? Algunas de las razones se describen a continuación.

### **2.1.2.1. Los mensajes SMS se pueden enviar y leer en cualquier momento**

Hoy en día, casi todas las personas tienen un teléfono móvil y se lo llevan consigo la mayor parte del tiempo. Con un teléfono móvil, puede enviar y leer mensajes SMS en cualquier momento, sin importar que esté en su oficina, en un autobús o en el hogar.

### **2.1.2.2 . Los mensajes SMS pueden ser enviados a un teléfono móvil fuera de línea**

A diferencia de una llamada telefónica, es posible enviar mensajes SMS, incluso cuando el teléfono móvil destino no está encendido o cuando está en un lugar donde la señal inalámbrica no está disponible temporalmente. El sistema de SMS de la red móvil del operador almacenará el mensaje de texto y luego será enviado cuando el destinatario este en línea.

### **2.1.2.3. Los Mensajes SMS son menos molestos y permiten mantenerlo en contacto**

A diferencia de una llamada telefónica, no es necesario leer o contestar un mensaje SMS de inmediato. Además, la escritura y lectura de mensajes SMS no hacen ningún ruido. Si bien usted tiene que correr de un teatro o la biblioteca para responder a una llamada de teléfono, no es necesario hacerlo si se utiliza la mensajería SMS.

**2.1.2.4. Los SMS son soportados por el 100% de los teléfonos GSM y se pueden intercambiar independientemente del operador**

SMS es una tecnología madura. Es soportado por todos los teléfonos móviles GSM. No sólo se puede intercambiar mensajes SMS con los usuarios móviles del mismo operador de red, sino también se puede intercambiar mensajes SMS con los usuarios móviles de muchos otros operadores móviles en todo el mundo.

**2.1.2.5. SMS es una tecnología apropiada para aplicaciones Inalámbricas**

Estas son algunas de las razones que hacen de SMS de una tecnología apropiada para las aplicaciones inalámbricas:

- La mensajería SMS es soportado por el 100% de los teléfonos móviles GSM. La construcción de aplicaciones inalámbricas sobre la tecnología SMS puede maximizar el potencial base sobre los usuarios.
- Los mensajes SMS son capaces de transportar datos binarios, además de texto. Se pueden utilizar para transferir tonos de llamada, imágenes, logotipos de operadores, fondos de pantalla, animaciones, *vCards*, (entradas de la agenda), etc.



- SMS soporta la facturación inversa, que permite una forma de pago conveniente para el usuario. Por ejemplo, suponiendo que desea desarrollar una aplicación de descarga de tonos comerciales que cobra un porcentaje extra al usuario para cada tono de llamada descargado. Para comprar un tono de llamada, el usuario escribe un mensaje de texto SMS ordinario que contiene el ID de la melodía que quiere comprar y enviar SMS a su aplicación de facturación. Su aplicación SMS entonces envía uno o más mensajes SMS que llevan la melodía. Al usuario se le cargará una tarifa por cada mensaje SMS recibido. El cargo se incluirá en la facturación del usuario del teléfono móvil o bien se deducirá de su tarjeta prepago.

## **2.2. Ejemplo de aplicaciones de mensajería SMS**

Hay muchos tipos diferentes de aplicaciones de SMS en el mercado y muchos otros más que se están desarrollando. Aplicaciones en las que la mensajería SMS puede ser utilizada son virtualmente ilimitadas. Se describen algunos ejemplos comunes de aplicaciones SMS para dar a continuación a algunas ideas de lo que se puede hacer con la mensajería SMS.

### **2.2.1. Mensajes de texto de persona a persona**

La mensajería de texto Persona a persona es la aplicación más comúnmente usada y es para lo que la tecnología SMS fue diseñado originalmente. En este tipo de aplicaciones de mensajería de texto, un usuario móvil escribe un mensaje de texto SMS con el teclado de su teléfono móvil, ingresa el número del teléfono móvil del destinatario y hace clic en una opción determinada en la pantalla, tales como " Enviar "o" Aceptar ", para enviar el mensaje de texto.

Cuando el teléfono móvil del destinatario recibe el mensaje de texto SMS, éste le notificará al usuario mediante un sonido o vibración. El usuario puede leer el mensaje de texto SMS algún tiempo después o inmediatamente y puede enviar un mensaje de texto de respuesta nuevo si así lo desea.

Una aplicación de *chat* es otro tipo de mensajería de texto Persona a persona que permite que un grupo de personas puedan intercambiar mensajes de texto SMS en forma interactiva. En un *chat*, todos los mensajes de texto SMS enviados y recibidos se muestran en la pantalla del teléfono móvil por orden de fecha y hora.

Los mensajes de texto SMS escritos por diferentes usuarios móviles pueden ser mostradas en diferentes colores para una mejor legibilidad.

### **2.2.2. Suministro de información**

Muchos proveedores de contenido hacen uso de mensajes de texto SMS para enviar información, como noticias, reporte del clima o los datos financieros a sus abonados. Muchos de estos servicios de información no son gratuitos. La facturación Inversa es una forma común utilizada por los proveedores de contenidos para facturar a sus usuarios. Al usuario se le cobra una determinada tasa por cada mensaje SMS recibido. El costo es incluido en la factura mensual del teléfono móvil o se deduce de los créditos de tarjetas de prepago.

### **2.2.3. Descarga de contenido**

Los mensajes SMS pueden transportar datos binarios y de este modo pueden ser utilizados como medio de transporte para realizar descargas inalámbricas. Objetos tales como tonos de llamada, fondos de escritorio, imágenes y logotipos del operador puede ser codificado en uno o más mensajes SMS según el tamaño del objeto. Al igual que los servicios de información, los servicios de descarga inalámbrica por lo general no son gratuitos y pueden ser cargados a la facturación del cliente.

### **2.2.4. Alertas y notificaciones de correo electrónico, fax y mensajes de voz**

SMS es una tecnología muy adecuada para la entrega de alertas y notificaciones de eventos importantes. Esto se debe a dos razones:

- a. Un teléfono móvil es un dispositivo que el propietario lleva consigo la mayor parte del tiempo. Siempre que un mensaje de texto SMS se recibe, el teléfono móvil le avisará mediante un sonido o vibración. Puede comprobar lo que el mensaje de texto SMS contiene inmediatamente o un tiempo después.
- b. SMS permite la tecnología "*push*" de la información. Esto es diferente del modelo "*pull*" en el que un dispositivo tiene que comprobar periódicamente el servidor para ver si hay alguna información nueva. El modelo "*push*" es menos adecuado para aplicaciones de alerta y notificación, ya que desperdicia recursos y aumenta el ancho de banda cargando el servidor.

En un sistema de notificación por correo electrónico, un servidor envía un mensaje de texto al teléfono móvil del usuario cuando llega un mensaje de correo electrónico a la bandeja de entrada. El mensaje de texto puede incluir la dirección de correo electrónico del remitente, el objeto y las primeras líneas del cuerpo de correo electrónico. Un sistema de notificación por correo electrónico permite al usuario personalizar varios filtros para que un SMS de alerta se envíe sólo si el mensaje de correo electrónico contiene ciertas palabras clave o si el remitente del *email* es una persona importante.

#### **2.2.5. Alertas de transacciones de tarjeta de crédito y e-Commerce**

Cuando una transacción de tarjeta de crédito o de *e-commerce* se hace, el servidor envía un mensaje de texto al teléfono móvil del usuario. El usuario puede saber inmediatamente si se ha hecho una transacción no autorizada.

#### **Alertas Bolsa de valores**

En una aplicación de alertas para el mercado de valores, un programa está en constante vigilancia y análisis del mercado de valores. Si una determinada condición se cumple, el programa envía un mensaje de texto al teléfono móvil del usuario para notificar de la situación. Por ejemplo, puede configurar el sistema de alerta de tal manera que si el precio de las acciones de una empresa es inferior a un determinado valor o cae por un determinado porcentaje, se enviará un SMS de alerta.

### **2.2.6. Sistema de monitoreo remoto**

En un sistema remoto de control un programa (a veces con la ayuda de un grupo de sensores) está en constante supervisión del estado de un sistema. Si una determinada condición se cumple, el programa envía un mensaje de texto al administrador del sistema para notificar a sobre la situación. Por ejemplo, un programa puede hacer "*ping*" a un servidor de forma periódica. Si no se recibe una respuesta desde el servidor, el programa puede enviar un SMS de alerta al administrador del sistema que le comunicará que el servidor puede estar fuera de línea.

### **2.2.7. Marketing SMS**

SMS se puede utilizar como una herramienta de *marketing*. Un ejemplo es un sistema de SMS tipo boletín. Después de inscribirse, el usuario recibirá mensajes de texto SMS acerca de los últimos descuentos y productos de la empresa. Si el usuario tiene alguna pregunta o comentario, puede enviar un mensaje de texto con la pregunta o comentario en el mismo. La compañía puede incluir su número de teléfono en el boletín de SMS para que el usuario pueda hablar con el personal de servicio al cliente directamente si es requerido.

## **2.3. Short Message Service Central SMSC**

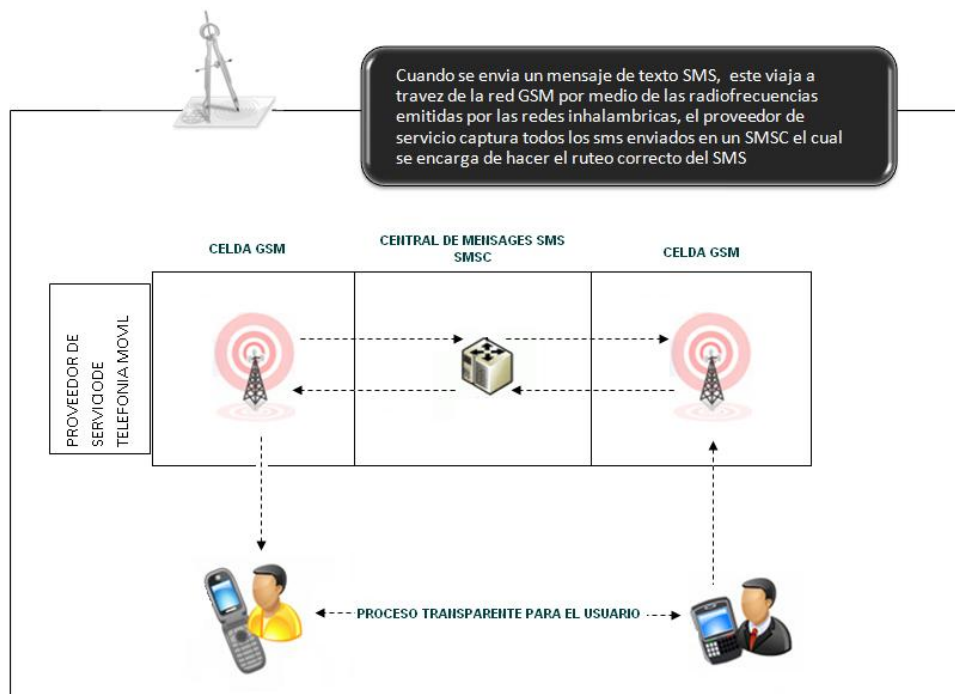
### **2.3.1. El centro de mensajes SMS**

Un centro de SMS (SMSC) es el responsable de manejar las operaciones SMS de una red inalámbrica. Cuando un mensaje SMS es enviado de un teléfono móvil, el primer lugar que alcanzará es una Central de mensajes SMS o SMSC. El SMSC entonces envía el mensaje entrante hacia su destino. Un mensaje SMS puede tener que pasar por más que una entidad de red antes de alcanzar se destino. La tarea principal de un SMSC es el enrutamiento de mensajes SMS y regular el proceso de transmisión. Si el recipiente no está disponible (por ejemplo, cuando el teléfono móvil es apagado), el SMSC almacenará el mensaje SMS. Esto expedirá el mensaje SMS cuando el recipiente está disponible.

Muy a menudo un SMSC es dedicado para manejar el tráfico SMS de una red inalámbrica. Un operador de red por lo general maneja su propio SMSC y los localiza dentro de su sistema de red inalámbrico. Sin embargo, es posible para un operador de red usar un tercer SMSC que es localizado fuera del sistema de red inalámbrico.

Debe de conocer la dirección de SMSC del operador de red inalámbrico para usar la mensajería SMS en su teléfono móvil. Típicamente una dirección de SMSC es un número de teléfono ordinario en el formato internacional. Un teléfono móvil debería tener una opción de menú que puede ser usada configurar la dirección de SMSC. Normalmente, la dirección de SMSC es predeterminada en la tarjeta de SIM por el operador de red inalámbrico, lo que quiere decir que el usuario del dispositivo no tiene que hacerle ningún cambio.

Figura 6. Flujo de envío y recepción de mensajes de texto SMS



Fuente: elaboración propia

## 2.4. Opciones SMS / SMSC para el tratamiento de los mensajes

### 2.4.1. Período de validez de un mensaje texto

Un mensaje SMS se almacena temporalmente en el centro de mensajes, si el receptor del teléfono móvil de destino está desconectado de la red (Apagado o fuera del área de cobertura). Es posible especificar el período tras el cual el mensaje SMS se suprimirá del centro de mensajes, transcurrido este tiempo, el mensaje de texto no será remitido al teléfono móvil del destinatario, hasta cuando esté nuevamente en línea. Este período se denomina el período de validez.

Aquí hay un ejemplo que ilustra la forma en que el período de validez se puede utilizar.

Supongamos que está viendo un interesante programa de televisión, el cual se está transmitiendo en este momento. Usted piensa que un amigo podría sentir interés por dicho programa. Por lo tanto, envía un mensaje de texto diciéndole a su amigo sobre el programa de TV. Digamos que el programa de TV dura sólo una hora. Su mensaje de texto no será útil si el teléfono móvil de su amigo no está disponible antes de que el programa termine. En esta situación, puede que quiera enviar un mensaje de texto con un período de validez de una hora. El operador de red no entregará el mensaje de texto a su amigo, si su teléfono móvil no está disponible antes de que finalice el período de validez.

Los teléfonos móviles deberían tener una opción de menú que pueda ser usada para fijar el período de validez. Tras configurar el período de validez, el teléfono móvil incluirá esta configuración para todos los mensajes de salida de forma automática.

#### **2.4.2. Informe de envío**

Es un mensaje que envía el SMSC indicando si el mensaje de texto enviado ha llegado al teléfono móvil del destinatario.

A veces es útil saber si un mensaje SMS ha llegado exitosamente al teléfono móvil del destinatario. Para obtener esta información, es necesario configurar una bandera en el mensaje de texto para notificar al centro de mensajes que se desea un informe sobre el estado del mensaje SMS. El informe es enviado en forma de SMS.



Los teléfonos móviles deberían tener una opción de menú donde se pueda configurar si el informe de estado está activado o desactivado. Tras configurar esta característica, el informe de estado será aplicado para todos los mensajes de salida de forma automática.

Esta opción de configuración está desactivada por defecto en la mayoría de los teléfonos móviles y GSM *módems*.

Después de dejar el teléfono móvil, un mensaje viaja al centro de mensajería SMSC. Cuando alcanza el centro de mensajes, el SMSC envía un informe de envío del mensaje al teléfono móvil del remitente para informar si existen errores o fallas (por ejemplo, formato de SMS incorrecto, SMSC ocupado, etc.) Si no hay ningún error o fallo, el centro de SMS envía un informe positivo al teléfono móvil. De lo contrario devuelve un informe negativo al teléfono móvil. El teléfono móvil alertara al usuario sobre el error y su causa.

Si el teléfono móvil no recibe el informe de envío de mensaje después de un período de tiempo, entonces se llega a la conclusión de que el informe se ha perdido. El teléfono móvil podría entonces volver a enviar el mensaje SMS al centro de mensajes SMSC. Una bandera se establecerá en el nuevo mensaje de texto SMS para informar al centro que este mensaje SMS había sido enviado con anterioridad. Si el envío de mensaje anterior se ha realizado correctamente, el SMSC ignorará el nuevo mensaje SMS, pero enviará un informe de envío de mensaje de vuelta al teléfono móvil. Este mecanismo evita que el mismo mensaje de texto sea enviado muchas veces al destinatario.

La mayoría de las veces este mecanismo no es utilizado en la interfaz de usuario del teléfono móvil, pero siempre es validado en un nivel inferior.

Ejemplo de este mecanismo es el envío de un mensaje de texto cuando el saldo para efectuar dicho envío es insuficiente. Es un mensaje que envía el teléfono móvil del destinatario al SMSC para indicar si el mensaje ha sido recibido correctamente.

Después de recibir un mensaje SMS, el teléfono móvil receptor enviará de vuelta un mensaje de entrega a la central de mensajes SMS para informar si existen errores o fallos (por ejemplo, el formato de los mensajes SMS no es válido, no hay suficiente espacio de almacenamiento, etc.) Este proceso es transparente para el usuario móvil. Si no hay ningún error o fallo, el teléfono móvil receptor envía un informe de entrega positivo al centro de SMS. De lo contrario devuelve un informe de entrega negativo.

Si el remitente solicitó un informe de estado del mensaje anteriormente, el centro de SMS enviará el informe de estado del mensaje indicando si el mensaje fue o no recibido por el teléfono móvil destinatario.

Si el centro de mensajes SMS no recibe el mensaje de informe de entrega después de un período de tiempo, se llega a la conclusión de que el informe de entrega del mensaje se ha perdido. El centro de SMS a continuación, vuelve a enviar el mensaje SMS al destinatario.

## **2.5. Proceso de transmisión de un SMS entre terminales**

### **2.5.1. Mensajes SMS Intra Operador**

Los mensajes Intra operador son aquellos que involucran a un solo operador de red inalámbrica, y se da entre dos usuarios de la misma compañía telefónica.

Normalmente, el costo de envío de mensajes de texto en una red de Intra operador es menor que el costo del envío de otros tipos de mensajes SMS como por ejemplo los mensajes entre operadores.

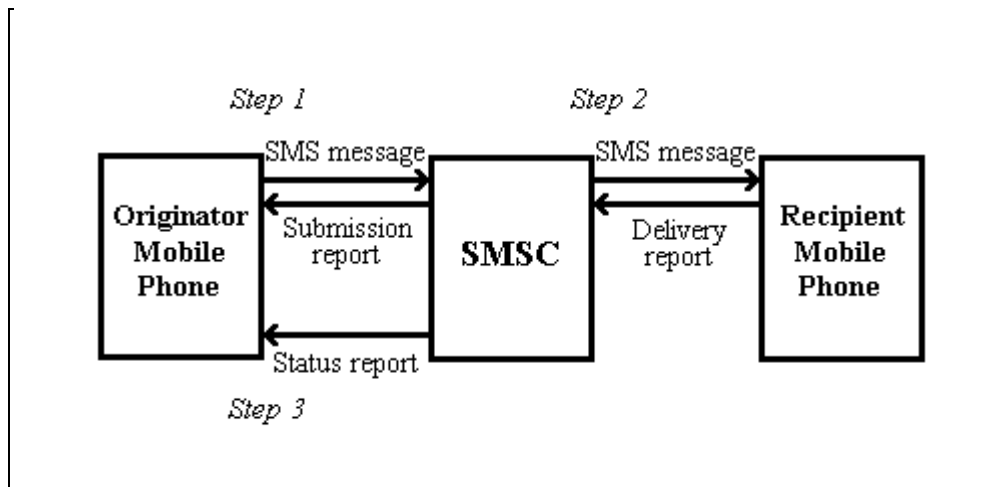
### **2.5.2. Proceso de transmisión de mensajes SMS intra operador**

La transmisión de mensajes Intra operador involucra únicamente un centro de SMS. Después de dejar el móvil del remitente, el mensaje intra operador llega al centro de mensajes SMS. Entonces el SMSC envía los mensajes SMS al teléfono móvil del destinatario.

Si el teléfono móvil del destinatario está fuera del área de cobertura o apagado (Fuera de línea), el centro de mensajes SMS almacenará el mensaje SMS. Se entregará el mensaje SMS al teléfono móvil del destinatario cuando esté nuevamente en línea. Si el período de validez del mensaje de texto expira y el teléfono móvil del destinatario aun se encuentra fuera de línea, el mensaje de texto SMS será eliminado del SMSC.

Cuando el centro de mensajes SMS recibe el informe de entrega del mensaje (enviado por el teléfono móvil del destinatario) o elimina el mensaje de texto (por ejemplo, cuando expira el período de validez), el SMSC envía un informe de estado del mensaje al remitente (cuando es solicitado con anterioridad).

Figura 7. **Proceso de transmisión de mensajes Intra Operador**



Fuente: *Short Message Service / SMS Tutorial*

[HTTP://www.developershome.com/SMS/intraInterInternationalSMS.asp](http://www.developershome.com/SMS/intraInterInternationalSMS.asp)

### 2.5.3. Mensajes SMS entre operadores

Los mensajes entre operadores involucran dos redes inalámbricas cada una propiedad de diferentes operadores de red. Este tipo de mensajes ocurren cuando los usuarios están suscritos en compañías diferentes.

Normalmente, el costo para el envío de mensajes SMS entre operadores es superior a la del envío de un mensaje intra operador.

### 2.5.4. Proceso de transmisión de mensajes SMS entre operador

La transmisión de un mensaje SMS entre operadores involucra uno o más centros de mensajes SMSC. En general, hay dos maneras diferentes para la transmisión de mensajes entre operadores.

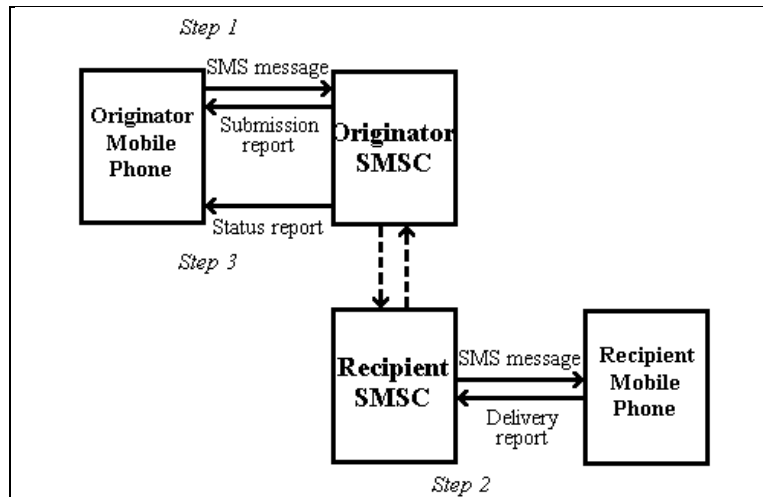
En la primera forma, la señal de transmisión se establece por interconexiones entre dos redes inalámbricas. Cuando el SMSC iniciador recibe un mensaje SMS entre operador, al mismo tiempo obtiene la información de ruteo de la red del teléfono móvil de destino y envía el mensaje de texto al destinatario de forma directa.

La primera forma se puede utilizar si las dos redes inalámbricas que participan en la transmisión de los SMS entre operadores se basan en tecnologías similares. Sin embargo, si esto no es cierto, la segunda forma tiene que ser utilizada. Por ejemplo, cuando un mensaje SMS es enviado desde una red GSM a una red CDMA.

En el segundo modo, el centro de SMS iniciador y el centro de mensajes del destinatario están interconectados a través de un *Gateway SMS* o con un protocolo de comunicación que cuenta con el apoyo de dos centros de SMS. El mensaje SMS llega al primer centro de mensajes SMSC (SMSC del emisor), éste enviará el mensaje SMS hacia el centro de mensajes SMSC del receptor.

El SMSC del destinatario se encargará de enviar el mensaje SMS al teléfono móvil del destinatario y el almacenamiento de los mensajes SMS si el receptor del teléfono móvil está desconectado.

Figura 8. **Proceso de transmisión de mensajes Entre Operador**



Fuente: *Short Message Service / SMS Tutorial*

[HTTP://www.developershome.com/SMS/intraInterInternationalSMS.asp](http://www.developershome.com/SMS/intraInterInternationalSMS.asp)

### 2.5.5. Mensajes de texto SMS internacionales

Los mensajes de texto entre operadores puede dividirse en dos categorías entre operadores locales y entre operadores internacionales. Un local es un mensaje SMS que se envía desde un operador de red inalámbrica a otro operador de red inalámbrica en el mismo país, mientras que un internacional es aquel que se envía desde la red inalámbrica en el operador de un país a un operador de red inalámbrica en otro país.

Por lo general, el costo para el envío de un mensaje SMS Entre internacionales es mayor que el local. Por lo tanto, el costo para el envío de un mensaje de texto Intra Operador  $\leq$  al costo del envío de un mensaje Entre operadores  $\leq$  al costo del envío de mensajes internacionales.

La interoperabilidad de la mensajería SMS entre dos redes inalámbricas a nivel local e incluso internacional, es, sin duda, un factor principal que contribuye al éxito de SMS en todo el mundo.

## **2.6. Enviar y recibir mensajes de texto haciendo uso de una computadora**

### **2.6.1. Envío de mensajes SMS desde un ordenador**

En general, hay dos métodos para enviar mensajes SMS desde una PC a un teléfono móvil:

- A. Conectando un teléfono móvil o un *módem* GSM/GPRS a la PC. Entonces la PC es cargada con un programa que envía instrucciones al teléfono móvil o *módem* GSM/GPRS para que éste envíe mensajes SMS.
- B. Conectando la PC a un centro de SMS (SMSC) o al *Gateway* SMS de un proveedor de telefonía móvil o bien Conectar la PC al *Gateway* de un proveedor de servicios SMS. Entonces se puede enviar mensajes SMS que usan un protocolo /interfaz, por ejemplo HTTP.

### **2.6.2. Envío de mensajes SMS desde un ordenador mediante un teléfono móvil o *Modem* GSM**

La especificación de SMS ha definido una forma de que un ordenador pueda enviar mensajes SMS a través de un teléfono móvil o *módem* GSM.

Un *módem* GSM/GPRS es un *módem* inalámbrico que trabaja con redes GSM/GPRS inalámbricas. Un *módem* inalámbrico es similar a un *módem* de marcado de disco *Dial-Up*. La diferencia principal es que un *módem* inalámbrico transmite datos por una red inalámbrica mientras que un *módem* de marcado de disco *Dial-Up* transmite datos por una línea telefónica de cobre.

Para enviar mensajes SMS, primero se debe colocar una tarjeta de SIM válida de un proveedor de red inalámbrica en un teléfono móvil o en el *módem* GSM/GPRS, que entonces es conectado a la computadora.

Hay varios modos de conectar un teléfono móvil o el *módem* GSM/GPRS a una computadora. Por ejemplo, un cable serial, un cable de USB, *Bluetooth* o infrarrojo. El modo de uso depende de la capacidad del teléfono móvil o *módem* GSM/GPRS. Después de la conexión del teléfono móvil o del *módem* a la computadora, se puede controlar el teléfono móvil o *módem* para el envío de instrucciones mediante el envío de instrucciones haciendo uso de una terminal. Las instrucciones utilizadas para controlar el teléfono móvil o *módem* son llamados comandos AT. (Comandos AT también se utilizan para el control de *módem Dial-Up*).



La siguiente tabla muestra los comandos AT que se relacionan con la escritura y el envío de mensajes SMS:

Tabla I. **Comandos AT relacionados a SMS**

<b>Comandos AT</b>	<b>Significado</b>
+ CMGS	Enviar mensaje
+ CMSS	Enviar mensaje de almacenamiento
+ CMGW	Escribir mensaje en la memoria
+ CMGD	Eliminar mensaje
+ CMGC	Comando Enviar
+ CMMS	Para enviar más mensajes

Fuente: elaboración propia

Una forma de enviar comandos AT a un teléfono móvil o *módem* es usar un programa terminal. Un programa de terminal funciona de la siguiente forma:

- Se envía los caracteres que componen el comando AT directamente al teléfono móvil o *módem*
- A continuación, muestra la respuesta que recibe desde el teléfono móvil *módem* en la pantalla

Para enviar mensajes SMS desde una aplicación, es necesario escribir el código fuente para la conexión y el envío de comandos AT al teléfono móvil o *módem*, al igual que lo hace un programa de terminal.

El código fuente puede ser escrito en lenguajes de programación como C, C++, *Java*, *Visual Basic*, *Delphi* u otros, sin embargo, escribir este código tiene algunas desventajas:

- Es necesario aprender cómo utilizar los comandos AT
- Es necesario saber cómo componer los *bits* y *bytes* de un mensaje SMS. Por ejemplo, para especificar la codificación de caracteres (por ejemplo, la codificación de 7 *bits* y 16 *bits* de la codificación *Unicode*) de un mensaje SMS, lo que se necesita es conocer los *bits* de la cabecera con el objetivo de asignar correctamente los *bits* del mensaje

Enviar mensajes SMS con un teléfono móvil o *módem* tiene una desventaja la velocidad de transmisión de SMS es bajo. A medida que la aplicación de mensajería SMS crece en usuarios, tiene que manejar un mayor tráfico de mensajes y, por último, el teléfono móvil o *módem* puede no ser capaz de soportar la carga. Para obtener una alta velocidad de transmisión de SMS, es necesario poseer una conexión directa a un SMSC o *Gateway* SMS de un operador de red o proveedor de servicios de SMS. Sin embargo, los comandos AT no se utilizan para comunicarse con un centro SMSC o *Gateway* SMS. Esto significa que al momento de querer pasar de una aplicación que usa un *módem* inalámbrico, a una aplicación basada en una solución SMSC se necesita realizar cambio significativo en la aplicación de mensajería SMS.

En la mayoría de los casos, en lugar de escribir su propio código para interactuar con el teléfono móvil o *módem* GSM a través de comandos AT, es mejor es usar una API (*Application Programming Interface*) o las librerías de mensajería de un SDK (*Software Development Kit*) / de alto nivel. El API / SDK contienen librerías que encapsulan los detalles de bajo nivel.

Así, un desarrollador de aplicaciones de SMS no necesita saber los comandos AT y la composición de mensajes SMS en el nivel de *bits*.

Algunos API / SDK de mensajería SMS soportan los mismos protocolos que un SMSC así que al momento de pasar de una solución basada en un dispositivo móvil o *módem* GSM a una solución SMSC por lo general sólo será necesario modificar algunas propiedades de un archivo de configuración y posiblemente realizar unos pequeños cambios en la configuración de la aplicación SMS.

Otra forma de ocultar el bajo nivel de la capa de comandos AT es colocando un *Gateway* SMS entre la aplicación de mensajería SMS y el teléfono móvil o *módem* GSM.

### **2.6.3. Envío de mensajes de texto desde una computadora a través de una conexión directa con el SMSC o SMS Gateway de una operadora de telefonía móvil**

Enviar mensajes SMS desde un ordenador a través de un teléfono móvil o *módem* GSM / GPRS tiene una limitación muy importante, la tasa de envío de SMS es muy baja. Si en nuestro caso es necesario el envío de SMS a alta velocidad, la obtención de una conexión directa al centro de SMS (SMSC) o *Gateway* SMS de un proveedor de servicios inalámbricos es necesaria. La conexión puede hacerse a través de *Internet*, X.25 o conexión telefónica. Si no se puede conseguir una conexión directa con el SMSC o *Gateway* SMS de un proveedor de servicios inalámbricos, otra opción es conseguir una conexión con el *Gateway* SMS de un proveedor de servicios de SMS, que enviará los mensajes SMS a un SMSC.

### **2.6.3.1. Dificultades para conseguir una conexión directa a un SMSC o al SMS Gateway de una operadora de telefonía móvil**

Puede ser difícil para las pequeñas empresas individuales o los desarrolladores de aplicaciones para obtener una conexión directa con el SMSC o *Gateway SMS* de un proveedor de servicios inalámbricos porque el proveedor de servicios inalámbricos podría proporcionar este servicio solamente a quienes tengan una transferencia alta de mensajes SMS. La adquisición de mensajes SMS al por mayor implica que el costo total será muy alto (aunque el precio a pagar por cada mensaje SMS será bajo).

Además, la información sobre el servicio (por ejemplo, el costo del servicio, protocolos soportados, la cobertura de red) por lo general no se especifica claramente en el sitio *web* del proveedor. Esto se debe a que el proveedor de servicios inalámbricos quiere saber más acerca de nuestra aplicación de mensajería SMS antes de ofrecer un precio y proveer más información. Para decidir cuál es proveedor de servicios inalámbricos más conveniente, es necesario establecer reuniones con el personal de ventas de cada uno de los operadores inalámbricos. Esto es problemático en caso que lo deseado sea enviar un pequeño número de mensajes SMS.

Una manera más conveniente para enviar mensajes SMS es utilizar el servicio de conectividad SMS de un proveedor de servicios de SMS, que se describe en la siguiente sección.

### **2.6.3.2. Proveedores de servicios de SMS (Proveedores de *Gateway* SMS, Distribuidores SMS, SMS *Brokers*)**

Existe gran demanda para aquellas aplicaciones SMS que no requieren el envío o recepción de gran cantidad de mensajes SMS. Un ejemplo es un sistema de control remoto. Si el sistema de monitoreo remoto considera que un determinado servidor no responde, se enviará un SMS de alerta al administrador del sistema. Este sistema de monitoreo remoto tendrá una cantidad muy pequeña del tráfico de SMS por mes ya que los servidores monitoreados deben estar funcionando bien la mayor parte del tiempo.

Puesto que los proveedores de servicios inalámbricos, por lo general no proporcionan SMSC o *Gateway* de acceso SMS a usuarios que no demandan gran cantidad de tráfico de SMS, algunas empresas vienen a llenar el vacío incluyendo que no hay requisito de compra mínima o requisitos mínimos de uso mensual.

Los proveedores de servicios SMS son también conocidos como los proveedores de *Gateway* SMS, Revendedores de SMS y *Brokers* de SMS.

- Proveedores de *Gateway* SMS - Un proveedor de servicios SMS proporciona un *Gateway* SMS para que sus usuarios envíen mensajes SMS. El *Gateway* enruta los SMS a otros destinos que finalmente resultan en un proveedor de SMSC.

- Revendedores y *Brokers* de SMS - los proveedores de servicios SMS compran una cantidad grande de mensajes SMS de uno o varios proveedores de servicios inalámbricos a bajo precio por mensaje. Luego venden los mensajes SMS a un precio más alto que el costo.

A diferencia de los operadores inalámbricos, muchos proveedores de servicios SMS proporcionar información detallada acerca de su servicio de conectividad SMS en su sitio *web*. Por ejemplo, se puede encontrar el costo del servicio, la cobertura de red, protocolos soportados, guías para los desarrolladores, etc. Por lo tanto, el servicio de los diferentes proveedores se puede comparar fácilmente. Si no se está satisfecho con, por ejemplo el precio o de cobertura de red de un proveedor de servicio SMS, puede simplemente salir de su sitio *web* y buscar otro proveedor de servicio SMS.

Otra ventaja de usar los servicios de conectividad SMS de los proveedores de servicio SMS es que su cobertura de red es muy buena. Ellos trabajan duro para abarcar el mayor número de redes inalámbricas como sea posible a fin de que sus servicios sean atractivos. Algunos proveedores de servicios de SMS permiten el envío a otros tipos de redes inalámbricas aparte de la GSM, como CDMA y TDMA.

Es fácil enviar mensajes SMS con un proveedor de servicios de SMS, siendo estos los pasos típicos:

- A. Crear una cuenta en el proveedor de servicio *web* del sitio de SMS (Un proveedor de servicios de SMS puede permitir que de nueva matriculación a los usuarios, enviar mensajes SMS gratis una pocas para probar su calidad de servicio);

- B. Entrar en la cuenta;
- C. Comprar un número de créditos en línea o mensajes de texto SMS. El pago puede hacerse por medio de tarjeta de crédito o *PayPal*;
- D. Enviar mensajes SMS utilizando un protocolo de interfaz (por ejemplo, HTTP, correo electrónico, FTP), soportado por el proveedor de servicios de SMS.

Los proveedores de servicios de SMS se pueden dividir en dos categorías dependiendo de cómo se haga el pago por el uso del servicio de mensajería SMS:

- Basado en el crédito – Consiste en comprar una cantidad de créditos. El envío de un mensaje SMS le costará uno o más créditos, en función del país al que desee enviar el mensaje SMS. Por ejemplo, el envío de un mensaje de texto SMS a la India podría costarle un crédito, mientras que el envío de un mensaje de texto SMS a los EEUU podría costar dos créditos.
- Basados en SMS – Consiste en comprar un número de mensajes SMS. El costo por mensaje SMS es el mismo para todos los destinos. Por ejemplo, si se compran diez mensajes SMS, puede enviar diez mensajes SMS no importa si el destino es la India o los EEUU.

El costo por mensaje SMS enviado depende de qué proveedor de servicios de SMS se elija y la cantidad de mensajes SMS que se compra. Se inicia en torno a \$0.06 dólares a \$0.07 por mensaje. Mientras más mensajes SMS se hayan adquirido, menor será el costo de enviar un mensaje SMS.

#### **2.6.4. Recepción de mensajes SMS usando un ordenador**

En general, hay tres modos de recibir mensajes SMS en la PC:

- A. Conectando un teléfono móvil o el *módem* GSM/GPRS a la PC. Entonces se configura la PC para obtener los mensajes SMS recibidos por el teléfono móvil o el *módem* GSM/GPRS.
  
- B. Consiguiendo el acceso al centro de SMS (SMSC) o la entrada SMS de un portador inalámbrico. Cualquier mensaje SMS recibido será expedido a la PC.
  
- C. Consiguiendo el acceso a la entrada SMS de un proveedor de servicio SMS. Cualquier mensaje SMS recibido será expedido a la PC.

De las tres opciones, la que no representa gastos fijos o de suscripción es el inciso A mientras que las otras dos requieren de un gasto fijo o de suscripción.

##### **2.6.4.1. Utilización de una PC para recibir mensajes SMS conectando un teléfono móvil o *módem* GSM/GPRS**

Recibir mensajes a través de un teléfono móvil o el *módem* GSM/GPRS tiene una ventaja principal sobre los otros dos métodos: Los portadores inalámbricos por lo general no cobran ningún honorario para recibir mensajes entrantes SMS con sus tarjetas de SIM. La desventaja de recibir mensajes SMS usando este método es que el teléfono móvil o el *módem* GSM/GPRS no pueden manejar una cantidad grande de tráfico SMS.



Un camino para resolver esto es el de balancear el equilibrio de tráfico SMS usando varios teléfonos móviles o *módems* GSM/GPRS.

Cada teléfono móvil o el *módem* GSM/GPRS tendrán su propia tarjeta de SIM y el número de teléfono móvil. En términos de programa, enviar y recibir mensajes SMS por un teléfono móvil o el *módem* GSM/GPRS son similares. Lo que usted tiene que hacer debe enviar instrucciones (en forma de órdenes AT) al teléfono móvil o el *módem* GSM/GPRS.

En términos de programación, el envío y recepción de mensajes SMS a través de un teléfono móvil o *módem* GSM / GPRS son similares. Lo que hay que hacer es enviar las instrucciones (en forma de comandos AT) en el teléfono móvil o *módem* GSM / GPS.

#### **2.6.4.2. Usar un ordenador para recibir mensajes SMS a través de una conexión directa con el SMSC o SMS Gateway de una operadora de telefonía móvil**

Al igual que el envío de mensajes SMS, recibir mensajes SMS a través de un teléfono móvil o *módem* GSM / GPRS tiene una limitación importante, que es la velocidad de transmisión. Para recibir mensajes SMS a través de una conexión directa con el SMSC o *Gateway* SMS de un proveedor de servicios inalámbricos, por lo general es necesario ponerse en contacto con el personal proveedor de servicios inalámbricos para discutir los detalles. Un proveedor de servicios inalámbricos por lo general sólo puede proporcionar este servicio a aquellos que tienen un tráfico enorme de SMS.

Sin embargo, si se divide el costo total por el número de mensajes SMS que se le permite recibir a través del SMSC o *Gateway* SMS, es posible que el precio por cada mensaje SMS entrante sea muy bajo. El servicio puede incluir un gasto único fuerte, una cuota de suscripción mensual y / o un precio por cada mensaje.

Para conocer el coste exacto del servicio, protocolos soportados, la cobertura de red y otra información, por lo general se tiene que contactar con el personal proveedor de servicios inalámbricos.

#### **2.6.4.3. Usar un ordenador para recibir mensajes SMS a través de una conexión al *Gateway* SMS de un proveedor de servicios de SMS**

Otra forma de recibir mensajes SMS utilizando un ordenador es a través de una conexión con el *Gateway* SMS de un proveedor de servicios de SMS. Los proveedores de servicios SMS por lo general dan soporte por medio de dos tipos de solución:

- A. *SIM hosting* - El proveedor de servicios SMS ofrece una tarjeta SIM para cada usuario. Todos los mensajes SMS enviados al número de teléfono móvil de la tarjeta SIM serán reenviados a nuestro sistema utilizando un protocolo de interfaz como por ejemplo HTTP.
  
- B. Número de teléfono compartido - El proveedor de servicios de SMS que ofrece un número de teléfono que es compartido por todos sus clientes para recibir mensajes SMS. Para que el mensaje sea redireccionado a nuestro sistema, el mensaje debe de iniciar con una palabra clave.

Cuando se reconoce la palabra clave el mensaje es redireccionado a nuestro sistema por medio de un protocolo de interfaz como HTTP.

Podemos percibir que estos métodos de recepción convierten el contenido del mensaje de texto SMS a un protocolo diferente como por ejemplo el HTTP lo cual facilita enormemente el tratamiento del mensaje y la ejecución de servicios basados en el contenido del mensaje de texto. Los cuales serán a su vez la respuesta que recibirá el usuario de la terminal móvil.



### **3. SERVIDORES SMS**

#### **3.1. Gateway de mensajería SMS**

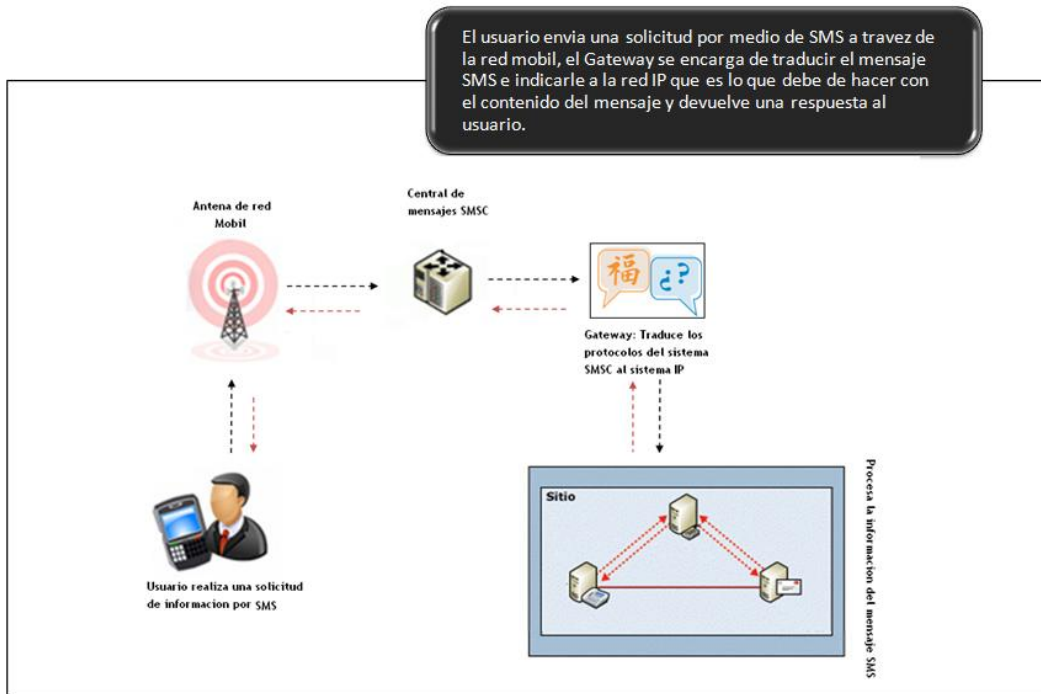
En general, un *Gateway* es un programa que permite conectar sistemas que tienen informaciones en diferentes formatos, realizando la conversión entre ellos de la forma más transparente posible.

En un *Gateway* de mensajería SMS, una de las partes a las que está conectado el *Gateway* es la red de telefonía celular móvil, sobre la que se envían los mensajes SMS. La otra vertiente del *Gateway* puede estar conectada a diferentes sistemas o servicios, aunque los casos más comunes incluyen sistemas de correo y redes IP (sistemas cliente/servidor).

El objetivo final del *Gateway* es enviar mensajes SMS a petición de equipos cliente que no disponen de los medios necesarios y recibir mensajes SMS para que sean procesados o reenviados por otros medios dentro de la red IP.

En otras palabras, un *Gateway* SMS es un programa encargado de traducir los mensajes SMS provenientes de la red de telefonía móvil a un formato legible por otra red de equipos tipo cliente/servidor (o red IP) que por naturaleza no posee los mismos protocolos de comunicación.

Figura 9. Flujo de transmisión de SMS de la red Móvil hacia la red IP



Fuente: elaboración propia

Bajo este esquema es posible proveer servicios de red IP, que los usuarios pueden consumir o activar enviando un mensaje de texto SMS al servidor equipado con el *Gateway*.

### 3.2. Gateway de mensajería SMS a IP

Las aplicaciones de un *Gateway* de mensajería SMS a IP son múltiples, pero, básicamente, el objetivo es extender a la red GSM las funcionalidades de aplicaciones que actualmente corren sobre una red IP con arquitectura cliente/servidor y posibilitar el acceso a recursos de diferentes tipos desde teléfonos móviles con posibilidades de envío y recepción de mensajes cortos.

Algunas aplicaciones que se utilizan habitualmente con *Gateway SMS* incluyen los servicios bancarios vía GSM, los *chats* de SMS, concursos de preguntas y respuestas, servicios de información de consumo de telefonía, servicios de información de servicios varios (resultados de apuestas, estado de carreteras, meteorología, etc.), y servicios de información de estado de equipos informáticos.

### **3.3. Aplicaciones del *Gateway SMS***

Los posibles usos que se pueden dar a un *Gateway SMS* entre las redes GSM e IP son bastante extensos.

La siguiente es una lista de actividades que se realizan en la actualidad con este tipo de sistemas de mensajería:

- Correo electrónico. El *Gateway* convierte un mensaje de correo en SMS y un mensaje SMS en mensaje de correo, con las consiguientes generaciones/eliminaciones de cabeceras de mensaje para ahorrar espacio, recordemos que un SMS tiene un límite en el número de caracteres que se pueden enviar y recibir.
- Distribución de mensajes SMS. Al igual que funcionan las listas de correo electrónico, en las que un mensaje es reenviado a los suscriptores de la lista, en la lista de distribución de mensajes SMS el *Gateway* permite el mantenimiento (alta/baja/consulta) de suscriptores y envía al resto de suscriptores de la lista los mensajes que no son comandos de actuación sobre el propio *Gateway*. Este tipo de actividad es apropiada por ejemplo para avisarle a los clientes sobre ofertas o promociones, enviar el estado del tiempo, condiciones del tránsito en la ciudad, etc.

- Recepción de alarmas de los sistemas de monitorización de servicios. Aplicaciones como "*mon*", "*heartbeat*", agentes "SNMP", etc. generan avisos cuando se alcanzan ciertos eventos. Estos avisos pueden ser encaminados mediante mensaje SMS dependiendo de su importancia, para que ciertas personas sean avisadas inmediatamente. Por ejemplo, en la actualidad la mayoría de sistemas de impresión se conectan a la red de trabajo. Es posible activar un servicio que le indique al asistente de servicio técnico cuando ha ocurrido un error trágico o cuando una impresora se ha quedado sin papel.
- Transporte de contenidos *web*. El SMS es utilizado como paquete de transporte para hacer llegar desde el móvil al servidor la petición de una página *web* y desde el servidor al móvil el contenido de dicha página una vez "filtrada" para eliminar imágenes, *tags* HTML, cabeceras de página, etc.
- Concursos de preguntas y respuestas. Ante una solicitud desde el teléfono móvil para comenzar el concurso, el *Gateway* envía mensajes con preguntas, recibe respuestas y mantiene un contador de resultados para cada participante, de manera que se pueden generar calificaciones, quien obtenga el puntaje mayor será el ganador, digamos en una promoción.



- Sistemas de seguimiento de flotas de vehículos. Un teléfono celular unido a un módulo GPS permite enviar información acerca de la posición exacta del portador del teléfono, de manera que para flotas de vehículos se reciben sus notificaciones de posición en el *Gateway* y ésta actualiza una base de datos consultable por otras aplicaciones que pueden mostrar la situación de cada elemento de la flota. Este es un ejemplo profundo de la interactividad y ampliación de la red móvil a la red IP.
- Notificación de estado de dispositivos aislados. Máquinas de *vending*, dispositivos de control de temperatura y presencia, etc. que se encuentren aislados y sin comunicación con una red mediante enlace físico pueden hacer uso de los mensajes SMS para recibir órdenes y para notificar de su estado (queda poca bebida, la temperatura ha superado los 45 grados, etc.). Normalmente esta comunicación se realiza sin intervención manual, por lo que realmente se conectan equipos automáticos en ambos lados. Es lo que *Nokia* denomina M2M (*machine to machine*).

Además, de inmediato, a cada persona le surgen nuevas aplicaciones, orientadas a su área de conocimiento:

- Consulta de bases de datos
- Mantenimiento de sistemas. Consultas de estado de servicios, reinicio de equipos
- Notificación de informaciones académicas. Notas, convocatorias de examen, etc.

- Banca GSM. Servicios financieros y alarmas para el seguimiento de operaciones de valores
- Cualquier otro tipo de alarmas debido a caídas de tensión, detección de intrusos en *firewalls*, etc.
- Domótica. Control y consulta de dispositivos a distancia
- Avisos de intervención para equipos médicos, mantenimiento, rescate, etc.
- Dadas las características de la red GSM que permiten la movilidad de los terminales (teléfonos) en su zona de cobertura, se pueden imaginar aplicaciones que aprovechen la posibilidad de localización de un teléfono, en base a la estación base de la red que en ese momento tiene conexión con él. Sin embargo, esta información no es directamente accesible desde el exterior de la red del operador de telefonía, por lo que, salvo aplicaciones fuertemente integradas con la red del operador, no es posible su utilización.

Se puede imaginar una aplicación que permita emitir mensajes SMS a teléfonos móviles entre las 2 y las 3 de la tarde en la zona de cobertura de una estación base situada junto a un restaurante, el restaurante previamente ha contratado los servicios de publicidad ofrecidos por un operador. Así es que para todos aquellos que pasen cerca del restaurante entre las 2 y 3 de la tarde, se les enviará un mensaje de texto indicando la ubicación del restaurante y el menú del día.

Al margen de la aplicación, podrían surgir problemas con la utilización de la posición de los terminales para operaciones no solicitadas por el propietario del terminal, ya que al fin y al cabo, la situación de cada terminal es información privada del propietario.

De cualquier manera, el *Gateway* de mensajería sólo trata de servir de intermediario y facilitar la labor de desarrollo de las extensiones móviles para una aplicación dada. El *Gateway* de mensajería SMS/IP trata de ser como un servidor *web*, realiza sus tareas de cambio de formato y ajuste del mensaje, pero precisa de contenidos que realmente le den una utilidad, aunque en este caso los contenidos son pequeñas o grandes aplicaciones que permiten interactuar al usuario móvil con otro programa.

También tiene un comportamiento similar al de un Agente de Transferencia de Mensajes de correo (MTA) ya que, de alguna forma, ésa también es su tarea: el encolado, conmutado y entrega de mensajes.

### **3.4. Como crear la conexión**

El éxito de este servicio, el SMS, parece provenir de la sencillez y facilidad de manejo, por un lado, y de que "hay alguien al otro lado" con quien realizar el acto de la comunicación. Estos dos factores han provocado dicho éxito, aún teniendo en contra el precio del servicio en algunos casos y las limitadas características de esta comunicación.

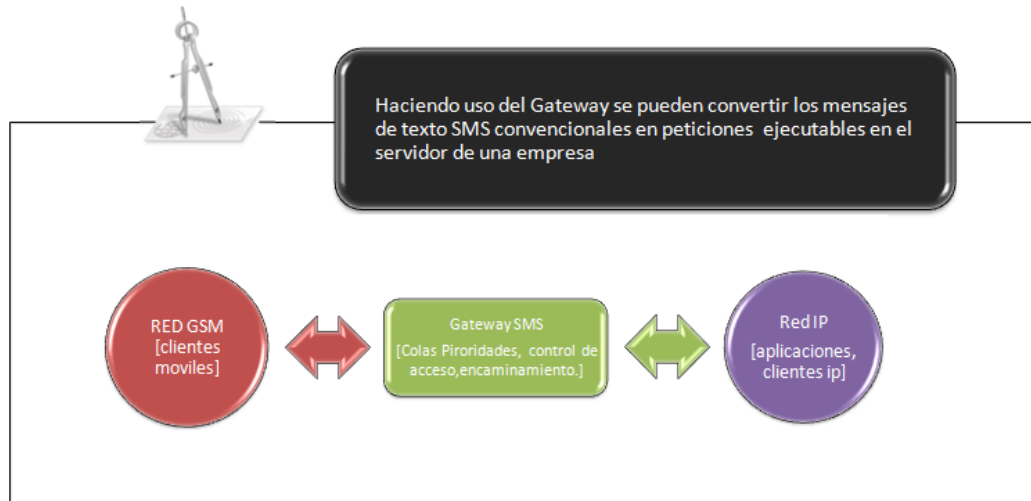
Las posibilidades de comunicación mediante "mensajes cortos GSM" (SMS) son muchas y muy variadas, pero siempre limitadas por las características de estos mensajes:

- Límite de un máximo de 160 caracteres
- Muy baja velocidad (en comparación con las líneas telefónicas convencionales)
- Duración limitada (24 ó 48 horas normalmente, si no se entregan antes son cancelados)
- No es un servicio garantizado (el mensaje suele llegar pero no hay garantía de ello, ni de que lleguen en el orden en que se han enviado)
- Posibilidad de comunicación sólo entre teléfonos celulares GSM entre los que haya "visibilidad" (que los operadores de los dos teléfonos, emisor y receptor, tengan convenio de intercambio de mensajes)

Existen muchas especificaciones de formato de mensaje para los servicios prestados a través de SMS que les dotan de gran potencia y complejidad. Pero es en el uso básico con un sistema de enlace sencillo donde se están obteniendo los mejores resultados, tanto de cantidad de mensajes enviados como de servicios que se están utilizando.

En todo caso, estas posibilidades resultan suficientes aprovechadas de forma adecuada, y una de esas formas es tener uno de los lados de la comunicación gobernado por un servicio automático que se encargue de responder a las peticiones recibidas desde múltiples teléfonos celulares. La automatización de la recepción de los mensajes SMS, su procesado y posterior respuesta es lo que conforma la funcionalidad de un *Gateway* SMS.

Figura 10. **Esquema general de una Gateway SMS**



Fuente: elaboración propia

El acceso a la red GSM se puede obtener de diferentes formas. El método más sencillo es utilizando directamente un terminal GSM conectado al ordenador que actúa de *Gateway*. En realidad este terminal puede ser un teléfono GSM normal con su *kit* de conexión a PC (cable y *software*) o un *módem* GSM (igual que los *módems* convencionales de red telefónica básica - RTB- pero su medio de transmisión es la red GSM, no el par de hilos telefónicos).

La comunicación entre el ordenador y el terminal se suele realizar por un puerto de comunicaciones serie. Casi todos los teléfonos móviles actuales incluyen un *módem* GSM en su interior, de manera que la forma de comunicar ordenador y teléfono/*módem* GSM es la misma que con un *módem* de RTB convencional, es decir, comandos AT.

Si el teléfono no incluye un *módem* GSM en su interior, es necesario comunicar con el teléfono utilizando las especificaciones del protocolo que el fabricante haya utilizado (normalmente se trata de protocolos propietarios, aunque cada vez menos). Este es el caso de algunos modelos de teléfonos *Nokia* como el 5110 y 6110.

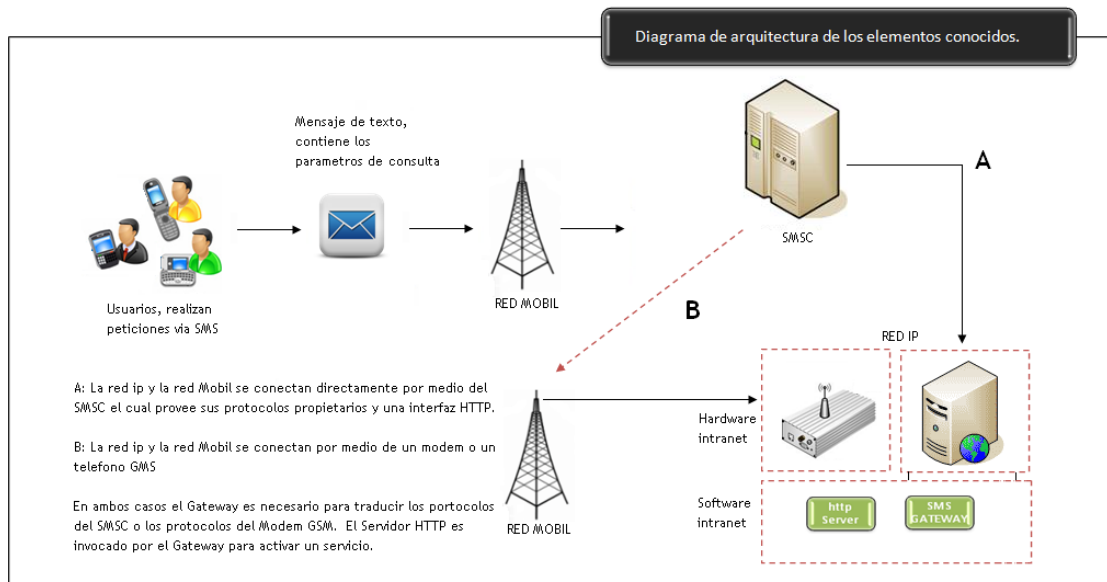
En *GNU/Linux* se puede utilizar el proyecto *Gnokii*, que trata de implementar el paquete de *software Nokia Data Suite* para la comunicación con estos teléfonos que implementan un protocolo propietario de *Nokia*.

El otro método más habitual para acceder a la red GSM es contactar directamente con un centro servidor de mensajes (SMSC) del operador de telefonía. Los SMSC de cada fabricante incorporan también protocolos propietarios, por lo que es necesario realizar en cada caso un diálogo diferente con cada tipo de SMSC, además de que el medio de conexión también puede variar de unos a otros (IP, X.25, RDSI,...). Una vez comprendido el funcionamiento básico de un *Gateway* de mensajería SMS, nos centraremos en la unión de las redes GSM, en un lado del *Gateway*, e IP, en el otro, dada su gran extensión y su uso común en todos los tipos de sistemas informáticos.

Existen, o pueden existir, otros *Gateway* SMS entre las redes de telefonía celular y otras redes de ordenadores, o incluso otras redes celulares de diferente tecnología, pero no serán objeto de estudio en este documento.

En el siguiente gráfico se muestra una arquitectura con los elementos vistos en este capítulo y en el capítulo 2, para aterrizar de forma gráfica los conceptos vistos.

Figura 11. Diagrama de arquitectura uniendo elementos conocidos



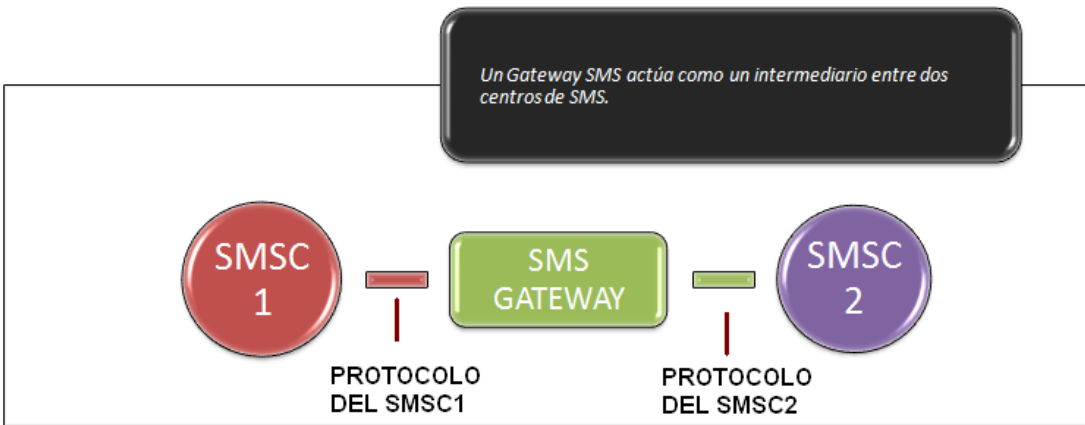
Fuente: elaboración propia

### 3.5. El Gateway y su relación con los SMSC y las aplicaciones de mensajería SMS

Uno de los problemas de la mensajería SMS es que los SMSC desarrollados por diferentes empresas utilizan su propio protocolo de comunicación y la mayoría de estos protocolos son propietarios. Por ejemplo, *Nokia* tiene un protocolo llamado CIMD mientras que otro proveedor SMSC, CMG, tiene un protocolo llamado EMI. No es posible interconectar dos SMSC si ambos no soportan el mismo protocolo de comunicación. Para hacer frente a este problema, un *Gateway* SMS se coloca entre dos SMSCs. Esto se ilustra en la siguiente figura.

El *Gateway* SMS funciona como un relé entre los dos SMSCs. Se encarga de traducir el protocolo de un SMSC al protocolo de otro SMSC para que ambos puedan comunicarse. Este método puede ser usado por dos diferentes operadores de red para poder interconectar sus SMSC con el fin de permitir el intercambio de mensajes en una red Entre operadores.

Figura 12. **Participación del Gateway**



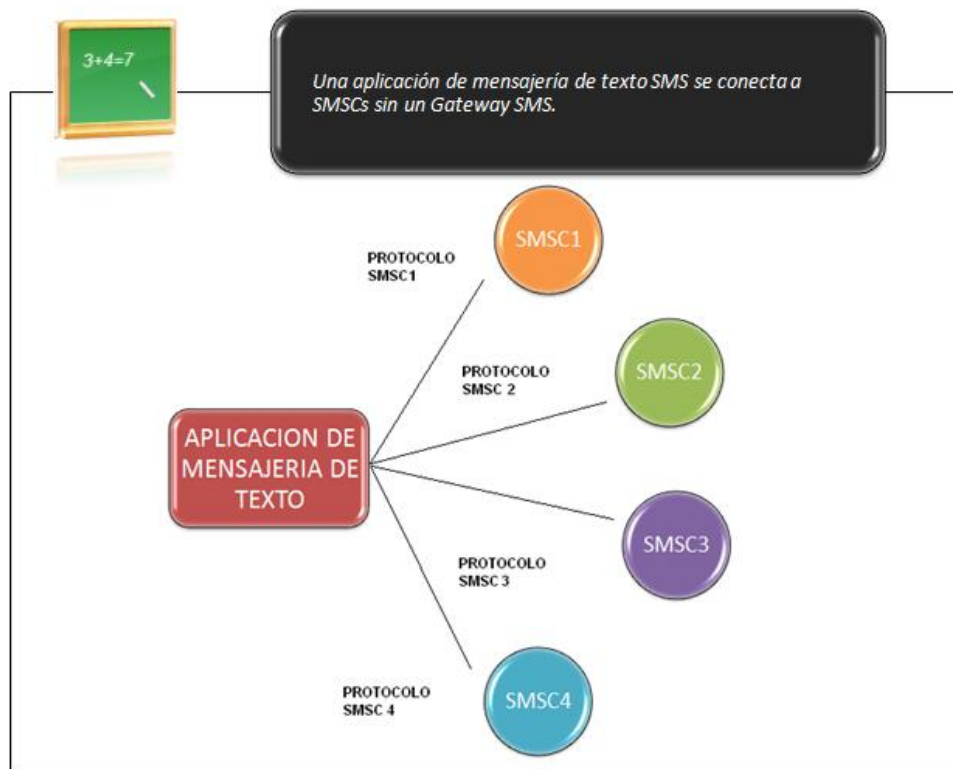
Fuente: elaboración propia

Además de los operadores inalámbricos, también los proveedores de contenido y desarrolladores de aplicaciones SMS pueden encontrar útil un *Gateway* SMS. Vamos a considerar la siguiente situación. Suponiendo que el cliente es el creador de una aplicación de mensajería de texto SMS. Para enviar y recibir mensajes de texto SMS en su servidor, puede optar por conectarse a través de un SMSC de algún operador de red móvil.



Diferentes operadores móviles pueden utilizar SMSCs. de diferentes fabricantes, lo que significa que su aplicación de mensajería de texto SMS puede necesitar ser soportada por múltiples protocolos SMSC específicos. (Esto se ilustra en la siguiente figura). Como resultado, la aplicación de mensajería de texto SMS agranda su complejidad y el tiempo de desarrollo se incrementa.

Figura 13. **Aplicación de mensajería de texto sin usar un Gateway**



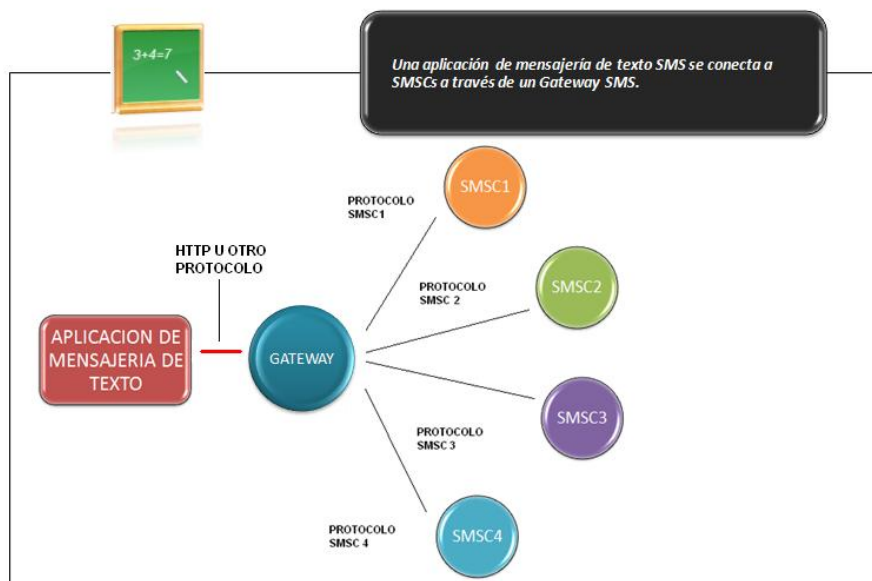
Fuente: elaboración propia

Para hacer frente al problema anterior, un *Gateway* SMS se puede configurar para manejar las conexiones con el SMSCs. Ahora, la aplicación de mensajería de texto SMS sólo necesita saber cómo conectar con el *Gateway* SMS. Para soportar las distintas tecnologías de SMSC, sólo tiene que modificar la configuración del *Gateway* SMS. Sin cambios en el código fuente de la aplicación de mensajería de texto SMS. La utilización de un *Gateway* SMS puede acortar el tiempo y tamaño de la aplicación de mensajería de texto.

Para conectarse a un *Gateway* SMS, puede utilizar un protocolo como SMSC SMPP y CIMD. Algunos *Gateway* permiten el uso de una interfaz HTTP.

HTTP es más fácil de usar que los protocolos del SMSC. El inconveniente es que puede haber un menor número de características para el uso de SMS. Por ejemplo, un *Gateway* SMS no admite el envío de mensajes de imagen a través de la interfaz HTTP.

Figura 14. **Aplicación de mensajería de que utiliza un *Gateway***



Fuente: elaboración propia

### 3.6. Algunos proyectos SMS Gateway de fuente abierta

#### 3.6.1. *Kannel*

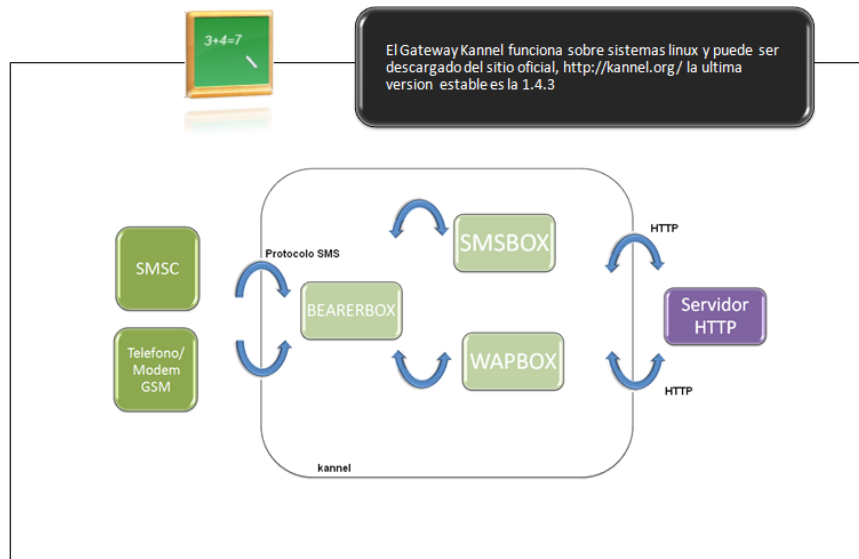
*Kannel* utiliza la licencia BSD. *Kannel* fue en primer lugar un *Gateway* WAP, pero amplió sus funcionalidades posteriormente para dar soporte a WAP usando SMS como transporte de los datos y para obtener páginas *web*. El código ahora es bastante estable, estando reconocido y certificado por el *WapForum* como un *Gateway* que se ajusta a los estándares WAP.

Detrás de *Kannel* hay un numeroso grupo de desarrolladores y colaboradores. La empresa que comenzó a desarrollar *Kannel* se llamaba *WapIt*. Recientemente tuvo problemas económicos y cerró.

Desde entonces, se está tratando de crear una fundación que dé apoyo al mantenimiento del desarrollo de *Kannel*, pero las últimas noticias que aparecen en el sitio *web* de *Kannel* son del mes de junio de este año.

*Kannel* es un proyecto grande, con una estructura detallada que le convierte en "demasiado complejo" para algunas instalaciones. Sin embargo, su objetivo principal es la escalabilidad y alta disponibilidad desde su planteamiento inicial. Básicamente, en una instalación mínima, aparece lo que en *Kannel* denominan "*Bearerbox*", el componente que comunica con el centro servidor de mensajes (SMSC) o un teléfono/ *módem* GSM que puede sustituirlo en parte. Con este *Bearerbox* comunican los otros componentes, "*SMSbox*" y "*WAPBox*", que se encargan de acceder a los servidores HTTP que tienen el contenido que se desea acceder, y que realizan las conversiones/compresiones que establece el protocolo WAP para el envío de los mismos a los teléfonos.

Figura 15. **Arquitectura del Gateway Kannel**



Fuente: elaboración propia

En una instalación orientada a SMS, desde el teléfono celular se envía un mensaje que es recibido por el SMSC u otro teléfono móvil, que recibe el mensaje para pasárselo al *Bearerbox*; el mensaje es encaminado al *SMSbox*, que accede a una página *web* (puede ser un CGI que genere la página dinámicamente, para ejecutar comandos o lanzar programas, por ejemplo) y el resultado es devuelto al *Bearerbox* de nuevo para ser devuelto al teléfono que originó la petición.

*Kannel* está orientado principalmente al trabajo directo con centros servidores de mensajes (SMSC) e implementa muchos protocolos de acceso a SMSC de diferentes fabricantes. Es en la conexión directa con un SMSC cuando se obtienen los mejores resultados de rendimiento; sin embargo, no siempre se tiene disponible una conexión de este tipo con la red GSM, por lo que también implementa conexión mediante *módem* GSM o teléfono móvil con *módem* incorporado.

### 3.6.2. *Alamín*

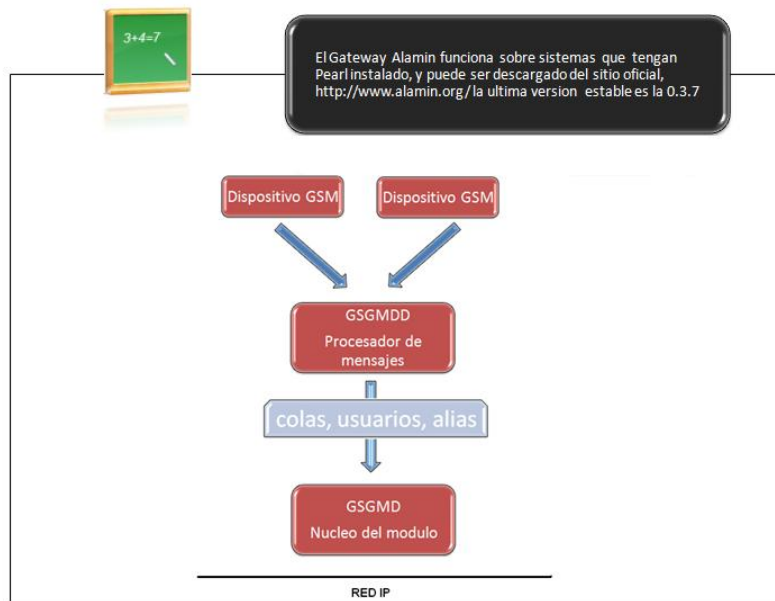
*Alamín* utiliza la licencia GPL. *Alamín* nació inicialmente con el objetivo de poder lanzar sencillas líneas de comando desde un teléfono móvil. En marzo de 1998 se implementó un servicio que monitorizaba un buzón de correo y recibía sus mensajes, interpretaba su contenido y realizaba alguna acción, devolviendo un resultado también por correo electrónico.

El acceso al teléfono móvil se realizaba mediante un *Gateway SMS/email*. Este pequeño *Gateway* estaba realizado en *Perl* y sus especificaciones se pueden encontrar en [HTTP://andressh.alamin.org/SSJP/](http://andressh.alamin.org/SSJP/). Este servicio no llegó a ser puesto en marcha en ninguna instalación real.

Posteriormente, a mediados de 1999, el proyecto *Gnokii* permitió acceso directo a la red GSM. Ya existían los *módems* GSM, pero estaban muy poco extendidos, eran caros y difíciles de conseguir. *Alamín* utilizó primero *Gnokii* como medio de acceso a la red GSM, y al evitar usar otro *Gateway* a más bajo nivel, se ganó en fiabilidad. En mayo de 2000 se pusieron en marcha las primeras instalaciones en explotación de *Alamín*.

*Alamín* se compone principalmente de dos elementos: el manejador de mensajes (*GSMDD*, *Message Dispatcher*) y el núcleo que gestiona las peticiones desde la red IP (*GSGCMD*, *Core Module*).

Figura 16. **Arquitectura del Gateway Alamin**



Fuente: elaboración propia

El *Core Module* es el que centra la gestión del servicio, el control de acceso de las peticiones desde la red IP, despliega los alias en los números que corresponda y prepara las solicitudes para ser enviadas. La comunicación entre los clientes de la red IP y el *Core Module* se realiza mediante el protocolo SMSQP, que utiliza por defecto el puerto 11201/TCP para su comunicación. Este protocolo no está terminado de especificar, ya que se está desarrollando con *Alamín*.

El *Message Distpatcher* se encarga de la entrega de los mensajes desde las colas de prioridades al dispositivo GSM, comprueba la existencia de mensajes entrantes y lanza programas externos para el tratamiento de los mensajes que han llegado. Inicialmente, el *Message Distpatcher* utilizaba *Gnokii* para realizar estas tareas, pero ahora ya usa comunicación directa con módems GSM mediante comandos AT.

En el momento actual de desarrollo, sólo se permite un dispositivo GSM por cada *Message Distpatcher*.

En sucesivas versiones se podrá utilizar múltiples dispositivos GSM, tablas de rutas para decidir por dónde se envía un mensaje, y su arquitectura será cambiada para facilitar que se puedan añadir de forma sencilla nuevos "controladores" para otros dispositivos GSM o para utilizar otros *Gateway* en cascada (HTTP, accesos directos a SMSC, protocolos de otros *Gateway*, etc.) .

Adicionalmente a estos dos componentes van a ser añadidos otros que implementan diferentes protocolos para permitir que otras aplicaciones, directamente, sin cambios, accedan a la red GSM. Uno de ellos, que ya está incluido en la distribución del *Gateway* y que implementa el protocolo SMTP, es GSGSMTPD, de esta forma, el *Gateway* traduce correo electrónico a SMS. Están pendientes de implementación otros protocolos, como POP, que permitirá que desde cualquier cliente de correo se puedan recibir mensajes SMS convertidos a correo electrónico.

También está pendiente de implementar el modo PDU en el envío de mensajes para *módems* GSM, el modo PDU en el envío de mensajes SMS mediante comandos AT permite enviar caracteres no imprimibles, es decir, no sólo letras y números. Esto permitirá el envío de mensajes en formatos avanzados o el envío de melodías y logotipos.

### **3.6.3. SMSLink**

*SMSLink* utiliza la licencia GPL. *SMSLink* es similar en muchos aspectos a *Alamín*, pero tiene ciertas diferencias de arquitectura.

En principio, sólo hay un proceso corriendo, y lanza nuevos hilos para cada petición de envío de mensaje recibido de la red IP. El acceso concurrente al dispositivo GSM se gestiona mediante un sistema de bloqueos y semáforos.

Permite el uso de múltiples dispositivos GSM desde su comienzo. Para la recepción de mensajes, existe la opción de almacenar los mensajes entrantes o pasárselos a un programa que los procese. Los dispositivos GSM soportados son todos los que soporten comandos AT extendidos, esto es, *módems* GSM y teléfonos con *módem* incorporado. Tanto el modo PDU como el modo texto están disponibles para el envío de mensajes.

En este momento no incorpora todavía gestión de colas, prioridades, usuarios y permisos por usuario. Estas son las deficiencias con respecto a *Alamín*, sin embargo, éste sólo puede manejar un dispositivo GSM, y en caso de ser un *módem*, no puede usar el modo PDU para envío de mensajes con melodías y logotipos.

#### **3.6.4. SMSTools**

*SMSTools* utiliza la licencia GPL. La filosofía general de la aplicación es la misma que *Alamín* y *SMSLink*. Tan sólo hay diferencias en detalles de implementación. Soporta hasta 32 dispositivos GSM simultáneos. Gestiona una cola por cada dispositivo y se pueden establecer rutas de manera que ciertos mensajes se envíen sólo por un dispositivo determinado.

También permite establecer qué programa debe ser lanzado cada vez que se recibe o se envía un mensaje. No resulta muy flexible en redes con problemas de transmisión o en momentos de mucha carga de la red, ya que la configuración de reintentos no resulta flexible.



## 4. PROGRAMACIÓN EN DISPOSITIVOS MÓVIL

### 4.1. Principales Sistemas operativos móvil

No hay necesidad de remontarse a la primera generación de telefonía celular analógica para observar que desde el arranque comercial del sistema GSM los terminales móviles han sufrido una profunda evolución. Desde el punto de vista de un sistema de comunicación celular esta evolución se percibe de modo que parece que los terminales “se han vuelto inteligentes”, en el sentido de que gran parte de la lógica de los servicios se ha desplazado de la red al terminal. En lo que respecta al punto de vista del usuario, el terminal ha dejado ser un dispositivo sólo “para hablar y comunicarse por mensajes cortos” y ha pasado a ser un “PC en miniatura” con prestaciones y servicios avanzados (fotos, vídeo, música, juegos, agenda personal, correo, navegación por *Internet*, etc.).

El incremento en las capacidades de proceso y la incorporación de nuevos dispositivos en los teléfonos móviles ha posibilitado la introducción de una serie de capacidades avanzadas en los terminales móviles, que los acercan cada vez más al concepto tradicional de ordenador personal frente al concepto más restringido de simple dispositivo de comunicaciones.

El desarrollo y gestión de las nuevas aplicaciones y servicios, que hacen uso de estas capacidades avanzadas de los terminales multimedia, ha sido posible gracias al empleo de nuevos sistemas operativos o plataformas de ejecución, más potentes y semejantes a los de los ordenadores personales.

En los últimos años han aparecido diversas alternativas en esta área, entre las que destacan:

- El sistema operativo *Symbian*, que es empleado por diversos fabricantes en una gran variedad de tipos de terminales. Su éxito está asociado a que no es propiedad de un único fabricante, sino de un consorcio de fabricantes.
- JME (*Java Micro Edition*), que no es un sistema operativo, sino una plataforma de ejecución que es soportada por múltiples sistemas operativos. Dicho de otra manera, un terminal que disponga del SO *Symbian*, del SO *Microsoft* o de otro cualquiera, incluirá habitualmente la correspondiente plataforma JME que le permita ejecutar aplicaciones escritas en *Java*.
- *JavaCard*, es una plataforma similar a la JME, pero se ejecuta en la tarjeta SIM de los terminales.
- El sistema operativo *Windows Mobile*, que es propiedad de *Microsoft*. Se utiliza en las PDAs (*Pocket PCs*) y los teléfonos (*Smartphone*).
- El sistema operativo *PalmOS*, que es propiedad de *Palm* y es empleado en sus PDAs y en algunos modelos de teléfonos.
- El sistema operativo *Linux*, está adquiriendo una considerable importancia en las PDAs y en los sistemas empotrados.

Una característica común de todos estos sistemas es que son abiertos, lo que significa que permiten el desarrollo, por parte de terceros, de nuevas aplicaciones que hagan uso de las capacidades ofrecidas por cada uno de ellos, y su posterior instalación en los terminales. Esta característica proporciona ventajas al usuario y al operador de telefonía móvil, pues:

- Por un lado, al usuario le permite “descargar” aplicaciones en el terminal de manera similar a como lo haría en un PC.
- Por otro, al operador le permite desarrollar sus propias aplicaciones, haciendo que se pueda diferenciar de los operadores de la competencia, esta diferenciación se produce por dos vías:
  - A través del desarrollo de aplicaciones exclusivas no disponibles con otros operadores.
  - Mediante la personalización (por ejemplo, con los colores y logotipos corporativos) y homogeneización de la apariencia (*look & feel*) de las aplicaciones de uso más corriente en los distintos tipos de terminales (reduciendo, por tanto, el período de aprendizaje al cambiar de terminal).
- Para el desarrollo de aplicaciones se dispone de avanzados entornos IDE (*Integrated Development environment*) que facilitan la tarea, ya que estos sistemas disponen, por ejemplo, de emulación de terminales y depuración de aplicaciones paso a paso.

- Finalmente, hay que indicar que actualmente entre los operadores de telefonía móvil se está planteando la posibilidad de realizar el desarrollo o adquisición de un sistema operativo, que además de ser abierto (sin comprometer la seguridad) no sea propiedad de un fabricante. Este sistema operativo sería controlado por los operadores y se convertiría en el estándar para todos los terminales.

La principal razón para esta estrategia es que las nuevas plataformas están cambiando la cadena de valor tradicional, y los operadores corren el riesgo de convertirse en meros “transportistas de datos” para servicios explotados por terceros. Un ejemplo de esta estrategia se puede ver en el SO *SavaJe*, que está despertando la atención de los operadores.

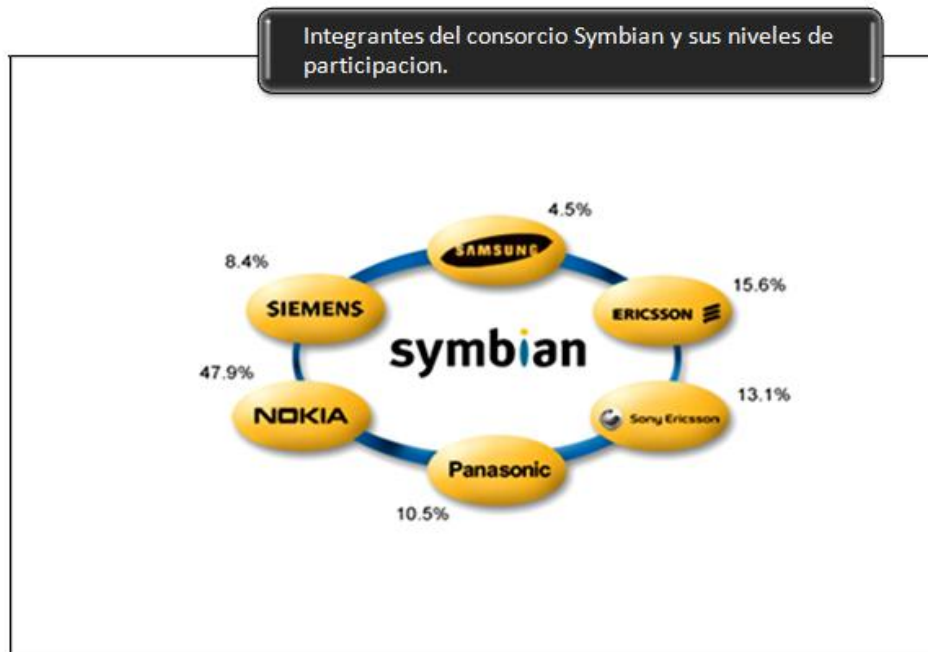
#### **4.1.1. *Symbian***

Actualmente *Symbian* es un consorcio propiedad de algunos de los principales fabricantes de terminales móviles a escala mundial: *Nokia*, *Panasonic*, *Samsung*, *Siemens*, *Ericsson* y *Sony-Ericsson*. El uso del sistema operativo (SO *Symbian*) realizado por este consorcio no está limitado exclusivamente a los propietarios, sino que cualquier otra compañía puede adquirir una licencia para incorporar dicho SO a sus terminales.

Un aspecto relevante actualmente, desde el punto de vista estratégico y comercial, en la evolución futura del consorcio *Symbian* es el papel dominante de *Nokia*, después de la salida de *Motorola* y *Pision* (ambos socios fundadores). Aunque *Motorola* ha abandonado completamente el consorcio sigue fabricando terminales con el SO *Symbian* como licenciataria de la tecnología.

Esta situación plantea dudas acerca del interés para otros fabricantes de permanecer como propietarios de *Symbian*. Tradicionalmente *Nokia* y *Sony-Ericsson* eran las compañías con un papel más relevante dentro del consorcio. Por ello ha habido distintas versiones del SO y las interfaces de usuario: los terminales de *Nokia* basados en las plataformas Series 60, 80 y 90 disponen de la versión 6.1 y la versión 7.0s; y los terminales de *Sony-Ericsson* disponen de la versión 7.0 con UIQ. Sin embargo, con la nueva situación es posible que *Nokia* adquiera un papel más predominante en la evolución futura del SO *Symbian*, lo cual es probable que provoque reacciones por parte de los restantes miembros del consorcio.

Figura 17. **Miembros del consorcio *Symbian***



Fuente: *Series 40 Platform: Introductory White Paper v3.3*

[HTTP://www.forum.nokia.com/info/sw.nokia.com/id/78c142ab-99d9-4ce6-b810-3962c7cea3f1/Series\\_40\\_Platform\\_White\\_Paper\\_V3\\_3\\_en.pdf.html](http://www.forum.nokia.com/info/sw.nokia.com/id/78c142ab-99d9-4ce6-b810-3962c7cea3f1/Series_40_Platform_White_Paper_V3_3_en.pdf.html)

#### 4.1.1.1. Componentes de la arquitectura *Symbian*

La versión más novedosa del SO *Symbian* es la 9. Sin embargo, existe un gran parque de terminales con la versión 7 y 8.

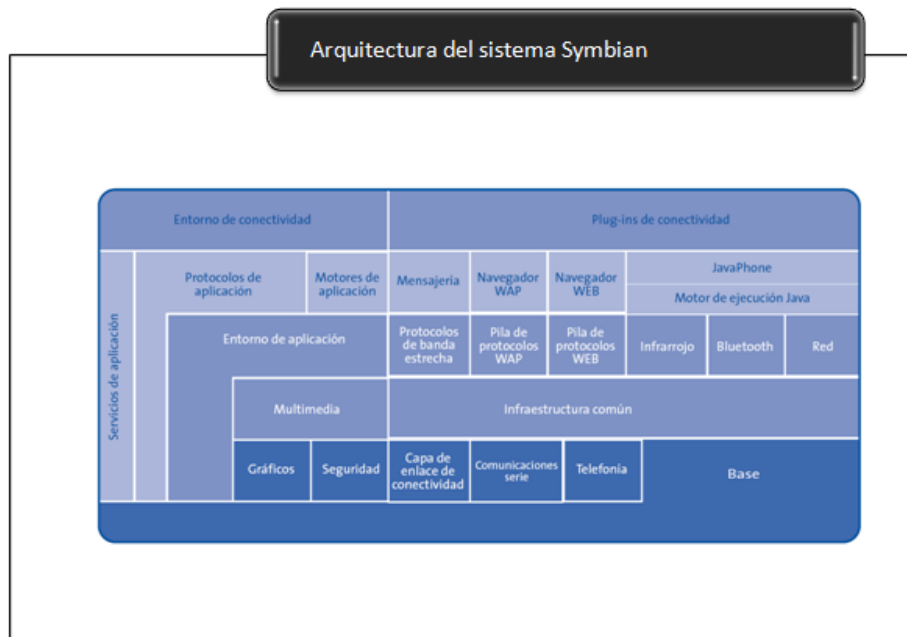
El primer terminal basado en el SO *Symbian*, el *Nokia Communicator 9200*, utilizaba la versión 6.0, que fue sustituida posteriormente por la versión 6.1. En la figura 18 se presenta la arquitectura del SO *Symbian* con sus funciones básicas. En la figura también se ilustra la dependencia entre los componentes del sistema (los componentes de los niveles más altos dependen de los componentes situados en los niveles inferiores).

Los componentes más importantes a destacar son:

- La base. Incluye los componentes básicos de todo sistema operativo: *kernel*, gestión de la memoria y de los procesos, sistema de ficheros, manejadores de dispositivos, seguridad de bajo nivel, librerías básicas de usuario, etc.
- El gestor de telefonía. Realiza la gestión de los sistemas de móviles celulares.
- Las pilas de infraestructura de comunicaciones y red. Incluyen la infraestructura de comunicaciones TCP/IP, GSM, GPRS, WAP, infrarrojos, *Bluetooth* y comunicaciones serie.
- El gestor multimedia. Se encarga de gestionar tanto la reproducción y grabación del audio como las diferentes funcionalidades de la imagen.

- El gestor de la seguridad. Se encarga de gestionar la seguridad relacionada con la descarga e instalación de las aplicaciones.
- La aplicación “*FrameWork*”. Contiene las librerías de *middleware* para la gestión de los datos, el texto, los portapapeles, los gráficos, la internacionalización y los componentes básicos de la interfaz gráfica de usuario.
- El gestor de mensajería. Realiza la gestión del correo electrónico, de los mensajes de texto y del *fax*.
- Los motores de la aplicación. Realizan la gestión de los contactos, la agenda, las tareas y otras aplicaciones.

Figura 18. **Arquitectura del sistema Symbian**



Fuente: Asociación Hispanoamericana de Centros de Investigación, Las telecomunicaciones y la movilidad en la sociedad de la información Pág. 87

#### 4.1.2. **Android**

*Android* es un Sistema Operativo además de una plataforma de *Software* basado en una versión modificada en el núcleo de *Linux*. Diseñada en un principio para dispositivos móviles, *Android* permite controlar dispositivos por medio de bibliotecas desarrolladas o adaptadas por *Google* mediante el lenguaje de programación *Java*. Inicialmente fue desarrollado por *Android Inc.*, compañía que fue comprada después por *Google*, y en la actualidad lo desarrollan los miembros de la *Open Handset Alliance* (liderada por *Google*).

*Android* es una plataforma de código abierto. Esto quiere decir, que cualquier desarrollador puede crear y desarrollar aplicaciones escritas con lenguaje C u otros lenguajes y compilarlas a código nativo de ARM (API de *Android*).

La presentación de la plataforma *Android* se realizó el 5 de noviembre de 2007 junto con la fundación *Open Handset Alliance*, un consorcio de 48 compañías de *hardware*, *software* y telecomunicaciones comprometidas con la promoción de estándares abiertos para dispositivos móviles.

Esta plataforma permite el desarrollo de aplicaciones por terceros a través del SDK, proporcionada por el mismo *Google*, y mediante el lenguaje de programación *Java*. Una alternativa es el uso del NDK (*Native Development Kit*) de *Google* para emplear el lenguaje de programación C.

El código fuente de *Android* está disponible bajo diversas licencias de *software* libre y código abierto destacando la versión 2 de la licencia *Apache*.



Tres son las principales versiones de *Android* hoy en día: 1.5, de nombre en clave *Cupcake*; 1.6 o *Donut* y 2.x, *Eclair*. Por supuesto también existieron versiones anteriores: la 1.0, con la que se lanzó el primer móvil *Android* del mercado, el HTC *Dream* o G1, de octubre de 2008; y la 1.1, de febrero de 2009, que solucionaba varios errores y añadía alguna que otra funcionalidad no muy importante.

*Android* podría ser una competencia directa a los sistemas operativos móviles como *Windows Mobile*, *Symbian*, *iPhone OS 3.0*, etc. aunque también podría aminorar o reducir la situación actual de *Microsoft* y sus Sistemas Operativos *Windows*.

En otras fuentes se ha dicho que HP Inc. uno de los gigantes en la fabricación y desarrollo de ordenadores más importantes del mundo, ha declarado que se están planteando la implantación del Sistema Operativo *Android* en Ultra portátiles o *Netbooks*.

#### **4.1.2.1. Componentes de la arquitectura *Android***

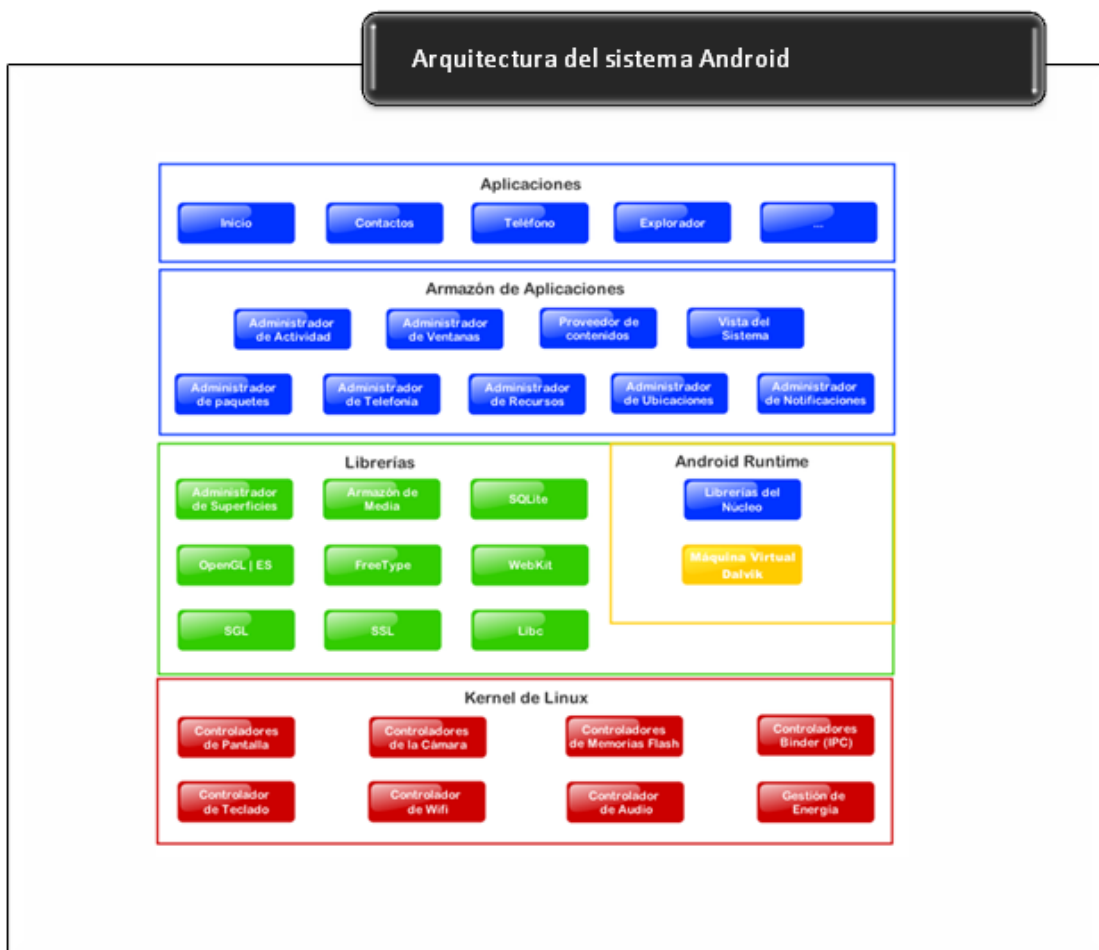
Los componentes principales del sistema operativo de *Android* (cada sección se describe en detalle):

- Aplicaciones: las aplicaciones base incluirán un cliente de *email*, programa de SMS, calendario, mapas, navegador, contactos, y otros. Todas las aplicaciones están escritas en lenguaje de programación *Java*.

- *Framework* de aplicaciones: los desarrolladores tienen acceso completo a los mismos APIs del *FrameWork* usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del *FrameWork*). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- Bibliotecas: *Android* incluye un set de bibliotecas C/C++ usadas por varios componentes del sistema *Android*. Estas características se exponen a los desarrolladores a través del *FrameWork* de aplicaciones de *Android*; algunas son: *System C library* (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3d, *SQLite*, entre otras.
- *Runtime* de *Android*: *Android* incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje *Java*. Cada aplicación *Android* corre su propio proceso, con su propia instancia de la máquina virtual *Dalvik*. *Dalvik* ha sido escrito de forma que un dispositivo, puede correr múltiples máquinas virtuales de forma eficiente. *Dalvik* ejecuta archivos en el formato *Dalvik Executable* (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros, y corre clases compiladas por el compilador de *Java* que han sido transformadas al formato.dex por la herramienta incluida "dx".

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

Figura 19. **Arquitectura del sistema *Android***



Fuente: <http://es.wikipedia.org/wiki/Android>

### 4.1.3. *BlackBerry*

La solución *BlackBerry* es compatible con las principales plataformas de correo electrónico de la empresa y se pueden personalizar para satisfacer las necesidades de una persona o una empresa entera, a cualquier escala, en cualquier industria.

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

Este SO es desarrollado actualmente por RIM (*Research in Motion*) que también ofrece servicios de correo electrónico a otros dispositivos a través de *BlackBerry Connect*.

El SO *BlackBerry* está claramente orientado a su uso profesional como gestor de correo electrónico y agenda. Desde la versión actual, la cuarta, se puede sincronizar el dispositivo con el correo electrónico, el calendario, tareas, notas y contactos de *Microsoft Exchange Server* además es compatible también con *Lotus Notes* y *Novell GroupWise*.

*BlackBerry Enterprise Server* (BES) proporciona el acceso y organización del *email* a grandes compañías identificando a cada usuario con un único *BlackBerry PIN*. Los usuarios más pequeños cuentan con el *software BlackBerry Internet Service*, programa más sencillo que proporciona acceso a *Internet* y a correo POP3 / IMAP / *Outlook Web Access* sin tener que usar BES.

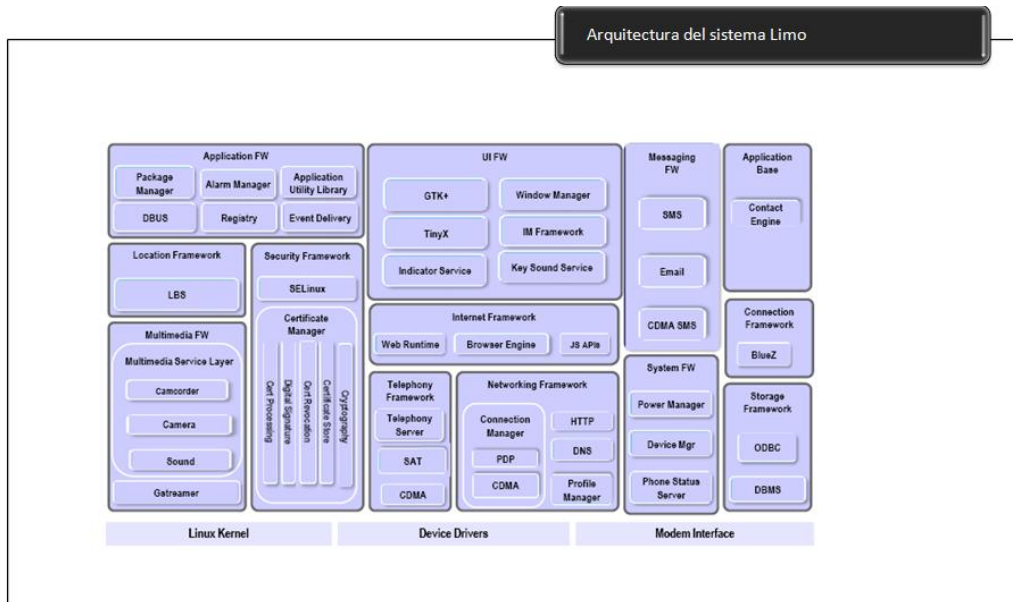
Al igual que en el SO *Symbian* desarrolladores independientes también pueden crear programas para *BlackBerry* pero en el caso de querer tener acceso a ciertas funcionalidades restringidas necesitan ser firmados digitalmente para poder ser asociados a una cuenta de desarrollador de RIM.

#### **4.1.4. LiMo**

*LiMo Foundation* es un consorcio de la industria dedicado a crear el primer sistema operativo basado en *Linux*, independiente de cualquier equipo físico y verdaderamente abierto, para dispositivos móviles. El respaldo de los principales líderes de la industria coloca a *LiMo* en el centro de la industria móvil y hace que *LiMo* sea la fuerza unificadora de *Linux* Móvil.

La misión de la *LiMo Foundation* es crear una plataforma abierta de *software* basado en *Linux* para uso de la industria en general a nivel mundial para la producción de dispositivos móviles por medio de un proceso de aportación balanceado y transparente que genere un rico ecosistema de productos, aplicaciones y servicios diferenciados de fabricantes, operadores, VSI e integradores de dispositivos.

Figura 20. **Arquitectura del sistema *LiMo***



Fuente: Asociación Hispanoamericana de Centros de Investigación, Las telecomunicaciones y la movilidad en la sociedad de la información Pág. 95

#### 4.1.5. ***Moblin***

*Moblin* es un sistema operativo tanto para dispositivos móviles, MIDs y *Netbooks Moblin* llegará para quedarse, ya que este nuevo proyecto es a código abierto y de distribución *Linux* con el soporte de la *Linux Foundation* enfocado al desarrollo de *software* orientado a dispositivos móviles para conectividad a *Internet* (MIDs) y nuevas clases de dispositivos como *Netbooks*.

*Moblin* surge de la idea de Intel donde se puso en marcha la *web* de *Moblin.org*. *Moblin* cuenta con una interfaz propia más estilizada y simplificada, buscando aprovechar al máximo el espacio del escritorio.

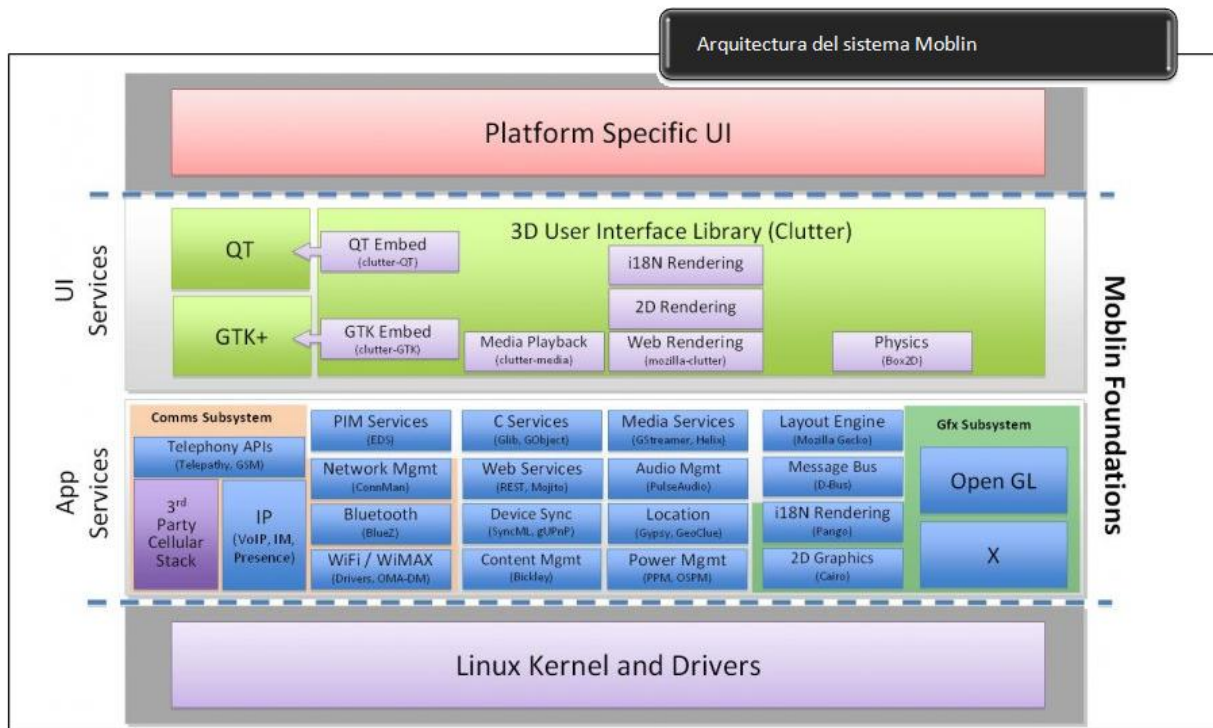
De esta manera *Moblin*, por ejemplo su menú se encuentra oculto hasta que colocas el ratón en la parte superior de la pantalla, desde donde podrás acceder a mensajería instantánea, navegador *web*, conexión a *Internet* y toda la multimedia.

*Moblin* era un proyecto de código abierto iniciado por Intel enfocado al desarrollo de *software* orientado a dispositivos móviles para conectividad a *Internet* (MIDs) y nuevas clases de dispositivos como *Netbooks* y *nettops*. Es decir, un sistema operativo diseñado para dispositivos móviles comprendidos entre los teléfonos y los ordenadores personales estándar.

Como el proyecto fue iniciado por *Intel*, el sistema operativo *Moblin 2* está especialmente diseñado para funcionar en procesadores Intel *Atom*, como los que podemos encontrar en un *Netbooks*. No obstante recientemente, abril del 2009, *Intel* traspasó *Moblin* a la *Linux Foundation*.

Hace poco ha salido la nueva versión de *Moblin*, siendo ésta la *Moblin 2.0* que integra funcionalidades sociales, ofreciendo acceso a sitios como *Facebook*, además de un navegador *web* basado en *Mozilla* y reproducción multimedia, todo ello desde un interfaz unificado.

Figura 21. **Arquitectura del sistema *Moblin***



Fuente: moblin.org

[HTTP://moblin.org/documentation/moblin-overview/architecture-overview](http://moblin.org/documentation/moblin-overview/architecture-overview)

#### 4.1.6. OS X / iPhone

El *iPhone OS* es el sistema operativo que utiliza el *iPod touch*, *iPhone* e *iPad*, diseñado por 175 ingenieros de *Apple*. Está basado en una variante del *Mach kernel* que se encuentra en *Mac OS X*. El *iPhone OS* incluye el componente de *software* “*Animation Core*” de *Mac OS X v10.5* que, junto con el *PowerVR MBX* el *hardware* de 3D, es responsable de las animaciones usadas en el interfaz de usuario. *iPhone OS* tiene 4 capas de abstracción: la capa del núcleo del sistema operativo, la capa de servicios principales, la capa de medios de comunicación y la capa de *Cocoa Touch*.



El sistema operativo, localizado en la partición *"/root"* del dispositivo, ocupa menos de 500 MB. Esto se realizó para poder soportar futuras aplicaciones de *Apple*. Este sistema operativo no tenía un nombre oficial hasta que salió la primera versión beta del *iPhone* SDK, el 6 de marzo de 2008. Antes de esto, *Apple* declaró, que “el *iPhone* controla un sistema operativo OS X”, una referencia al padre de los sistemas operativos de los *iPhone*, el *Mac OS X*. Como un *iPod*, el *iPhone* se maneja con la versión 7.3 de iTunes, el cual es compatible con *Mac OS X* versión 10.4.10, y con el *Windows XP* o el *Vista* de 32 bits. Al aparecer el *iTunes 7.6* se incluyeron las versiones de 64 bits del *Windows XP* y del *Vista*, y un *workaround* ha sido descubierto para el sistema operativo *Windows* anterior de 64 bits.

Desde la versión 2.0 estas son algunas características del sistema operativo:

- Mejora en la carga para los propietarios anteriores del *iPod touch*
- Soporta *Microsoft Exchange ActiveSync*
- Soporta el nuevo servicio de *Mobile Me de Apple*
- Botón para *e-mail*
- Botón para contactos
- Botón para calendario
- Lista de direcciones globales
- Certificados e identidades
- WPA2/802.1X
- Política de seguridad
- Lenguas para el teclado suplementarias incluyendo chino y coreano
- Soporta *Cisco IPsec VPN*
- La configuración del dispositivo
- Limpieza remota de archivos

- Capacidad de ver documentos *PowerPoint*
- Capacidad de ver documentos *iWork*, incluyendo páginas, números
- Eliminación masiva de *e-mails*
- Se pueden mover *e-mails* de forma masiva
- Protocolo de descubrimiento de servicio *Bonjour*
- Soporta SVG
- Controles de contenido explícito para seguridad de los niños
- Capacidad de buscar contactos
- Tienda de *Apple*
- Nuevo menú "Calendario" en "Ajustes"
- Calculadora puesta al día con características suplementarias en modo retrato, y una calculadora científica en modo apaisado con un icono puesto al día
- *iTunes* al día con un nuevo icono, e iconos ordenados por categoría
- Capacidad de ahorrar o abrir imágenes de sitios *Web* en *Safari*
- Soporta el chino tradicional y simplificado
- Etiquetado geográfico de las fotografías
- Noruego, sueco, danés, finlandés, polaco, coreano, y brasileño portugués son los idiomas que tiene la interfaz del *iPhone*, antes estas lenguas estaban sólo en el *iPod touch* desde 1.1.1
- *YouTube* tiene un *plugin* para *MobileSafari.app*
- Tiene la capacidad de ver vídeos musicales en retrato o en vista apaisada
- Tiene la capacidad de capturar la pantalla pulsando el botón "*home*" y pulsando el botón "duerme/desperta"
- Habilidad para apagar el Wi-Fi en modo avión
- El sonido desaparece cuando sales de un vídeo
- Icono de contactos en la pantalla «*Home*» (inicio)

- Las películas ahora se desplazan cuando el *iPhone* gira alrededor de algo (un paisaje o algo)
- Capacidad de seleccionar una cuenta de correo electrónico con *Mail*

#### **4.1.7. *Windows Mobile***

*Windows Phone*, anteriormente llamado *Windows Mobile* es un sistema operativo móvil compacto desarrollado por *Microsoft*, y diseñado para su uso en teléfonos inteligentes (*Smartphones*) y otros dispositivos móviles.

Todos los sistemas operativos o plataformas de *Microsoft* para dispositivos con capacidades más reducidas que los PCs derivan de *Windows CE*, que a su vez proviene de la especificación *Win32* empleada en PC, y se denominan genéricamente *Windows Mobile*.

La plataforma *Windows Mobile* para *Pocket PC* es un subconjunto bastante amplio de la API de *Windows CE*, y está orientada a dispositivos tipo PDA dotados de microprocesadores potentes y grandes pantallas.

Hay otra plataforma, conocida como *Windows Mobile* para *Smartphone*, que también es una adaptación de *Windows CE*, pero que, sin embargo, es más restrictiva, ya que está adaptada a teléfonos móviles. *Smartphone* especifica con alto grado de detalle las capacidades suministradas por el dispositivo, el modo de interacción con el usuario y las dimensiones de la pantalla gráfica (176 x 220 píxeles en la configuración preferente).

Por esta razón, una aplicación diseñada para un dispositivo *Smartphone* concreto funcionará de forma correcta en otro dispositivo que disponga de este sistema operativo, sin necesidad de realizar cambios (de forma similar a lo que ocurre con las plataformas *Symbian Series 60*, *70*, etc.).

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

Originalmente apareció bajo el nombre de *Pocket PC*, como una ramificación de desarrollo de *Windows CE* para equipos móviles con capacidades limitadas. En la actualidad, la mayoría de los teléfonos con *Windows Mobile* vienen con un *Stylus* digital, que se utiliza para introducir comandos pulsando en la pantalla. *Windows Mobile* ha evolucionado y cambiado de nombre varias veces durante su desarrollo, siendo la última versión la llamada *Windows Phone 7*, anunciada el 15 de febrero del 2010 y sujeta a disponibilidad a finales de 2010. En marzo de 2010, *Mozilla* anunció que no se continuaría con el desarrollo de su navegador *Firefox* para *Windows Mobile*.

Tanto *Windows Mobile* para *Pocket PC*, como *Windows Mobile* para *Smartphone*, poseen bastantes aspectos parecidos:

- En la pantalla "Hoy" nos mostrará la fecha actual, la información del dueño, las citas próximas, los mensajes *e-mail*, y las tareas. En la parte inferior aparecerá, generalmente, una barra con dos botones. También incluye una barra que incluye iconos para notificar el estado del *Bluetooth*, batería, cobertura, etc. Este tema predeterminado puede ser cambiado añadiendo o eliminando complementos, como por ejemplo, alarma, temperatura, estado de la batería.

- En la barra de tareas muestra: la hora actual, el volumen y el estado de la conectividad. Cuando un programa o un mensaje están abiertos el espacio en blanco, en el que estaba el reloj se convierte en una "ok" o un icono de cerrar (x). La característica principal de la barra de tareas es el botón de Inicio, que está diseñado para que sea parecido al botón de Inicio de las versiones de escritorio de *Windows*. El menú de Inicio ofrece programas abiertos recientemente, nueve entradas del menú personalizadas, y accesos directos a programas, ajustes, búsquedas, y ayuda.
- Las versiones *Pocket PC* incluyen en *Windows Mobile* aplicaciones de *Microsoft Office*, estos incluyen *Pocket Word* y *Pocket Excel*. En *Windows Mobile 5.0* se incluye *Pocket PowerPoint*. Éstas versiones incluyen muchas de las características que se utilizan en versiones de escritorio, pero algunas otras características como la inserción de las tablas e imágenes no se han incluido versiones anteriores a *Windows 5.0*. *ActiveSync* tiene la capacidad de convertir archivos de versiones de escritorio a archivos compatibles con *Pocket PC*.
- *Outlook Mobile* es también un programa que viene con *Windows Mobile*. Esto incluye tareas, calendario, contactos, y la bandeja de entrada. *Microsoft Outlook* para las versiones de escritorio se incluye a veces en los CD-ROM's del fabricante del *Pocket PC*.

- *Windows Media Player for Windows Mobile* se añade con el *software*. Actualmente, todas las *Pocket PC* incluyen la versión 9 del reproductor, pero la versión 10 se ha incluido con un *hardware* más nuevo y con las nuevas versiones de *Windows Mobile*. Para algunos dispositivos, la versión 10 está disponible para su descarga sólo para determinados dispositivos - estos incluyen los dispositivos de la gama de *Dell Axim*. *Windows Media Player* reproduce: WMA, WMV , MP3, y AVI. Los archivos MPEG actualmente no están soportados, y se debe descargar un programa de terceros para reproducirlos, y los archivos de WAV se reproducen en un reproductor por separado. Algunas versiones son también capaces de reproducir M4A.

#### 4.1.8. Tabla de Resumen de SO Móvil

Tabla II. Sistemas operativos para teléfonos Móviles

OS	LENGUAJES SOPORTADOS	DESCRIPCION
<i>Android</i>	<i>Java</i>	<i>Open Source OS</i> (basado en <i>Linux</i> ) <a href="http://developer.android.com">HTTP://developer.android.com</a>
<i>Blackberry</i>	<i>Java</i>	Compatible con J2ME, pero los API nativos proveen una mejor integración <a href="http://na.blackberry.com/eng/developers">HTTP://na.blackberry.com/eng/developers</a>
<i>LiMo</i>	C, C++	<i>Open Source OS</i> (basado en <i>Linux</i> ) <a href="http://limofoundation.org">HTTP://limofoundation.org</a>
<i>Maemo</i>	C, C++	<i>Open Source OS</i> (basado en <i>Linux</i> ) <a href="http://maemo.org/development">HTTP://maemo.org/development</a>
<i>Moblin</i>		<i>Open Source OS</i> (basado en <i>Linux</i> ) moblin.org
OS X / <i>iPhone</i>	<i>Objective-C, C</i>	Requiere una cuenta de <i>Apple Developer</i> <a href="http://developer.apple.com/iphone">HTTP://developer.apple.com/iphone</a>
<i>Symbian</i>	C, C++, <i>others</i>	<i>Open Source OS</i> Compatible con algunos teléfonos Series 60 Soporte para otros lenguajes como <i>Python</i> . La mayoría de estos soportan J2ME. <a href="http://www.symbian.org">www.symbian.org</a>
<i>Windows Mobile</i>	C#, C	Se puede programar directamente desde .NET Compact <i>Framework</i> y el SDK de .NET <a href="http://developer.windowsmobile.com">developer.windowsmobile.com</a>

Fuente: elaboración propia

## 4.2. La plataforma *Java Micro Edition* J2ME / *Java ME*

La compañía *Sun Microsystem* lanzó a mediados de los años 90 la tecnología *Java*, la cual, gracias al paradigma *Write Once, Run Anywhere*, se ha convertido en la tecnología dominante para el desarrollo de aplicaciones empresariales.

Una aplicación desarrollada en *Java* se puede ejecutar en cualquier plataforma que disponga de una máquina virtual *Java*, lo que no ocurre con otros lenguajes de programación, como es el caso de C++. El motivo es que la máquina virtual (que es diferente en cada plataforma específica) si “interpreta” los programas *Java*; sin embargo, aquellos programas que son “compilados” deben ser generados para cada una de las plataformas objetivo.

J2ME (*Java 2 Micro Edition*) Actualmente conocido como JME (*Java Micro Edition*) es la realización de la plataforma *Java* orientada a dispositivos con prestaciones inferiores a las de un PC, como es el caso de los teléfonos móviles. Forma parte de la versión 2 de *Java*, que incluye J2EE (*Java 2 Enterprise Edition*) para servidores y J2SE (*Java 2 Standard Edition*) para ordenadores de sobremesa.

La principal ventaja de JME es su independencia de la plataforma, lo que ha supuesto que prácticamente todos los fabricantes de terminales móviles, independientemente del SO empleado en el terminal, incorporen JME para la ejecución de aplicaciones desarrolladas por terceros (principalmente juegos) pero más adelante se explicará su potencial para aplicaciones empresariales.

Actualmente, excepto en los terminales de gama baja, la inmensa mayoría de los fabricantes de terminales incluyen esta tecnología.



Se abordara más a detalle la plataforma JME en las siguientes secciones por ser el centro de estudio de este capítulo.

*Java* en un principio fue diseñado para generar aplicaciones que controlaran electrodomésticos como lavadoras, frigoríficos, etc. Debido a su gran robustez e independencia de la plataforma donde se ejecutase el código, desde sus comienzos se utilizó para la creación de componentes interactivos integrados en páginas *Web* y programación de aplicaciones independientes. Estos componentes se denominaron *applets* y casi todo el trabajo de los programadores se dedicó al desarrollo de éstos. Con los años, *Java* ha progresado enormemente en varios ámbitos como servicios HTTP, servidores de aplicaciones, acceso a bases de datos (JDBC). Como vemos *Java* se ha ido adaptando a las necesidades tanto de los usuarios como de las empresas ofreciendo soluciones y servicios tanto a unos como a otros.

Debido a la explosión tecnológica de estos últimos años *Java* ha desarrollado soluciones personalizadas para cada ámbito tecnológico. Sun ha agrupado cada uno de esos ámbitos en una edición distinta de su lenguaje *Java*. Estas ediciones son *Java 2 Standard Edition*, orientada al desarrollo de aplicaciones independientes y de *applets*, *Java 2 Enterprise Edition*, enfocada al entorno empresarial y *Java Micro Edition*, orientada a la programación de aplicaciones para pequeños dispositivos.

La edición *Java Micro Edition* fue presentada en 1999 por *Sun Microsystems* con el propósito de habilitar aplicaciones *Java* para pequeños dispositivos. En esta presentación, lo que realmente se enseñó fue una primera versión de una nueva *Java Virtual Machine* (JVM) que podía ejecutarse en dispositivos *Palm*.

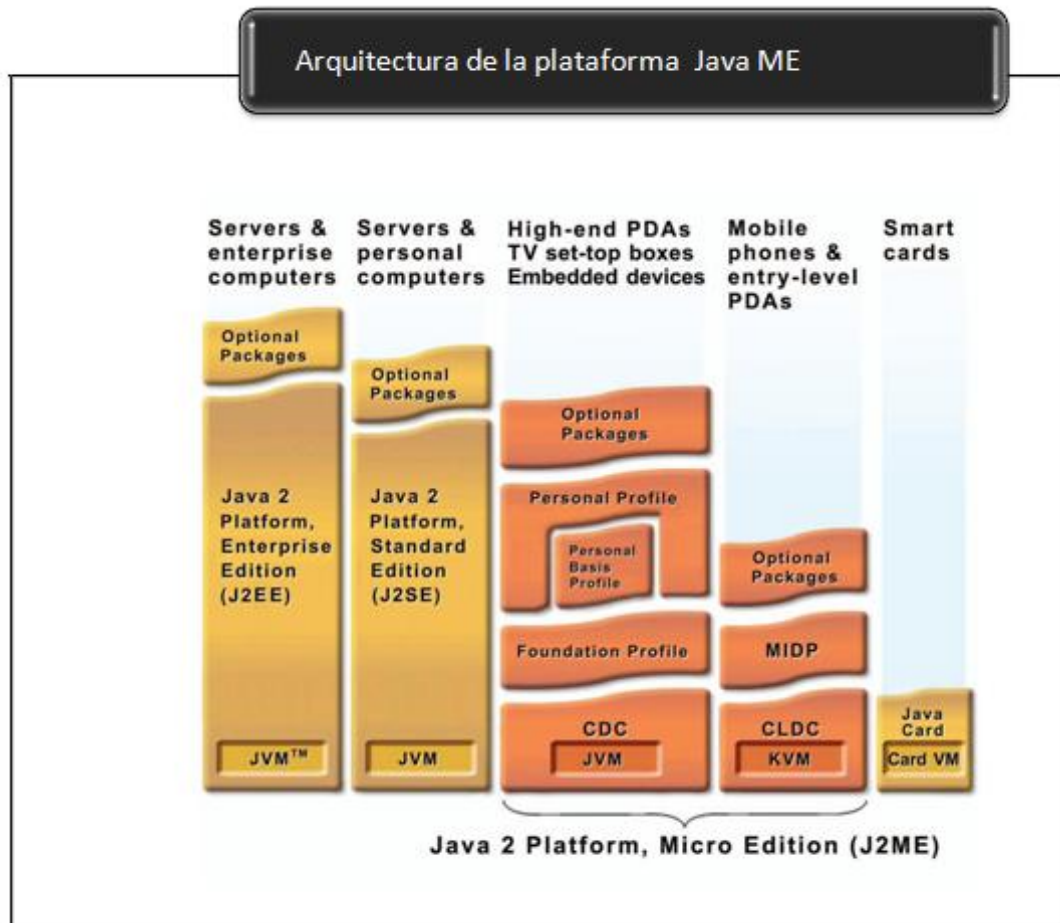
Para empezar puede decirse que *Java Micro Edition* es la versión del lenguaje *Java* que está orientada al desarrollo de aplicaciones para dispositivos pequeños con capacidades restringidas tanto en pantalla gráfica, como de procesamiento y memoria (teléfonos móviles, PDA, *Handheld*, *Pagers*, etc.).

La tardía aparición de esta tecnología, (se ha visto que la tecnología *Java* nació a mediados de los 90 y *Java Micro Edition* apareció a finales), puede ser debido a que las necesidades de los usuarios de telefonía móvil ha cambiado mucho en estos últimos años y cada vez demandan más servicios y prestaciones por parte tanto de los terminales como de las compañías. Además el uso de esta tecnología depende del asentamiento en el mercado de otras, como GPRS, íntimamente asociada a JME y que no ha estado al alcance del público hasta hace poco. JME es la tecnología del futuro para la industria de los dispositivos móviles. Actualmente las compañías telefónicas y los fabricantes de móviles están implantando los protocolos y dispositivos necesarios para soportarla.

#### **4.2.1. *Java Micro Edition***

Esta versión de *Java* está enfocada a la aplicación de la tecnología *Java* en dispositivos electrónicos con capacidades computacionales y gráficas muy reducidas, tales como teléfonos móviles, PDAs o electrodomésticos inteligentes. Esta edición tiene unos componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (*Kilo Virtual Machine*, debido a que requiere sólo unos pocos *Kilobytes* de memoria para funcionar) en vez del uso de la JVM clásica, inclusión de un pequeño y rápido recolector de basura y otras diferencias que se irán viendo más adelante.

Figura 22. **Arquitectura de la plataforma *Java ME***



Fuente: java a tope: j2me (java 2 micro edition). Edición electrónica

En la actualidad no es realista ver *Java* como un simple lenguaje de programación, sino como un conjunto de tecnologías que abarca a todos los ámbitos de la computación con dos elementos en común:

- El código fuente en lenguaje *Java* es compilado a código intermedio interpretado por una *Java Virtual Machine (JVM)*, por lo que el código ya compilado es independiente de la plataforma.

- Todas las tecnologías comparten un conjunto más o menos amplio de APIs básicas del lenguaje, agrupadas principalmente en los paquetes *java.lang* y *java.io*.

Un claro ejemplo de éste último punto es que JME contiene una mínima parte de las APIs de *Java*. Esto es debido a que la edición estándar de APIs de *Java* ocupa 20 Mb, y los dispositivos pequeños disponen de una cantidad de memoria mucho más reducida. En concreto, JME usa 37 clases de la plataforma J2SE provenientes de los paquetes *java.lang*, *java.io*, *java.util*. Esta parte de la API que se mantiene fija, forma parte de lo que se denomina “configuración” ya se hablará de ella más extensamente en el siguiente apartado.

Otras diferencias con la plataforma J2SE vienen dadas por el uso de una máquina virtual distinta de la clásica JVM denominada KVM. Esta KVM tiene unas restricciones que hacen que no posea todas las capacidades incluidas en la JVM. Estas diferencias se verán más detenidamente cuándo se analice las capacidades de la KVM en el siguiente apartado.

JME representa una versión simplificada de J2SE. Sun separó estas dos versiones ya que JME estaba pensada para dispositivos con limitaciones de proceso y capacidad gráfica. También separó J2SE de J2EE porque este último exigía unas características muy pesadas o especializadas de E/S, trabajo en red, etc. Por tanto, separó ambos productos por razones de eficiencia. Hoy, J2EE es un superconjunto de J2SE pues contiene toda la funcionalidad de éste y más características, así como JME es un subconjunto de J2SE (excepto por el paquete *javax.microedition*) ya que, como se ha mencionado, contiene varias limitaciones con respecto a J2SE.

Sólo de manera muy simplista se puede considerar a JME y J2EE como versiones reducidas y ampliadas de J2SE respectivamente en realidad cada una de las ediciones está enfocada a ámbitos de aplicación muy distintos. Las necesidades computacionales y APIs de programación requeridas para un juego ejecutándose en un móvil difieren bastante con las de un servidor distribuido de aplicaciones basado en EJB.

#### **4.2.2. Componentes de JME**

##### **Máquina virtual**

Se tiene una serie de máquinas virtuales *Java* con diferentes requisitos, cada una para diferentes tipos de pequeños dispositivos.

##### **Configuraciones**

Son un conjunto de clases básicas orientadas a conformar el corazón de las implementaciones para dispositivos de características específicas. Existen 2 configuraciones definidas en JME:

- *CLDC Connected Limited Device Configuration*: enfocada a dispositivos con restricciones de procesamiento y memoria.
- *CDC Connected Device Configuration* enfocada a dispositivos con más recursos.

## Perfiles

Son unas bibliotecas *Java* de clases específicas orientadas a implementar funcionalidades de más alto nivel para familias específicas de dispositivos.

En resumen, un entorno de ejecución determinado de JME se compone entonces de una selección de:

- A. Máquina virtual
- B. Configuración
- C. Perfil
- D. Paquetes opcionales

### 4.2.3. Máquinas Virtuales JME

Una máquina virtual de *Java* (JVM) es un programa encargado de interpretar código intermedio (*bytecode*) de los programas *Java* precompilados a código máquina ejecutable por la plataforma, efectuar las llamadas pertinentes al sistema operativo subyacente y observar las reglas de seguridad y corrección de código definidas para el lenguaje *Java*. De esta forma, la JVM proporciona al programa *Java* independencia de la plataforma con respecto al *hardware* y al sistema operativo subyacente. Las implementaciones tradicionales de JVM son, en general, muy pesadas en cuanto a memoria ocupada y requerimientos computacionales. JME define varias JVMs de referencia adecuadas al ámbito de los dispositivos electrónicos que, en algunos casos, suprimen algunas características con el fin de obtener una implementación menos exigente.

Figura 23. Entorno de ejecución JME



Fuente: elaboración propia

Existen 2 configuraciones CLDC y CDC, cada una con características propias, como consecuencia, cada una requiere su propia máquina virtual. La VM (*Virtual Machine*) de la configuración CLDC se denomina KVM y la de la configuración CDC se denomina CVM.

#### 4.2.3.1. KVM: *Kilobyte Virtual Machine*

Corresponde a la Máquina Virtual más pequeña desarrollada por Sun. Su nombre KVM proviene de *Kilobyte* (haciendo referencia a la baja ocupación de memoria, entre 40Kb y 80Kb). Se trata de una implementación de Máquina Virtual reducida y especialmente orientada a dispositivos con bajas capacidades computacionales y de memoria.

La KVM está escrita en lenguaje C, aproximadamente unas 24 000 líneas de código, y fue diseñada para ser:

- Pequeña, con una carga de memoria entre los 40 Kb y los 80 Kb, dependiendo de la plataforma y las opciones de compilación
- Alta portabilidad
- Modulable
- Lo más completa y rápida posible y sin sacrificar características para las que fue diseñada

Sin embargo, esta baja ocupación de memoria hace que posea algunas limitaciones con respecto a la clásica *Java Virtual Machine* (JVM):

- No hay soporte para tipos en coma flotante. No existen por tanto los tipos *double*, ni *float*. Esta limitación está presente porque los dispositivos carecen del *hardware* necesario para estas operaciones.
- No existe soporte para JNI (*Java Native Interface*) debido a los recursos limitados de memoria.
- No existen cargadores de clases (*class loaders*) definidos por el usuario, sólo existen los predefinidos.
- No se permiten los grupos de hilos o hilos *daemon*. Cuando requiere el uso de grupos de hilos se utiliza los objetos Colección para almacenar cada hilo en el ámbito de la aplicación.

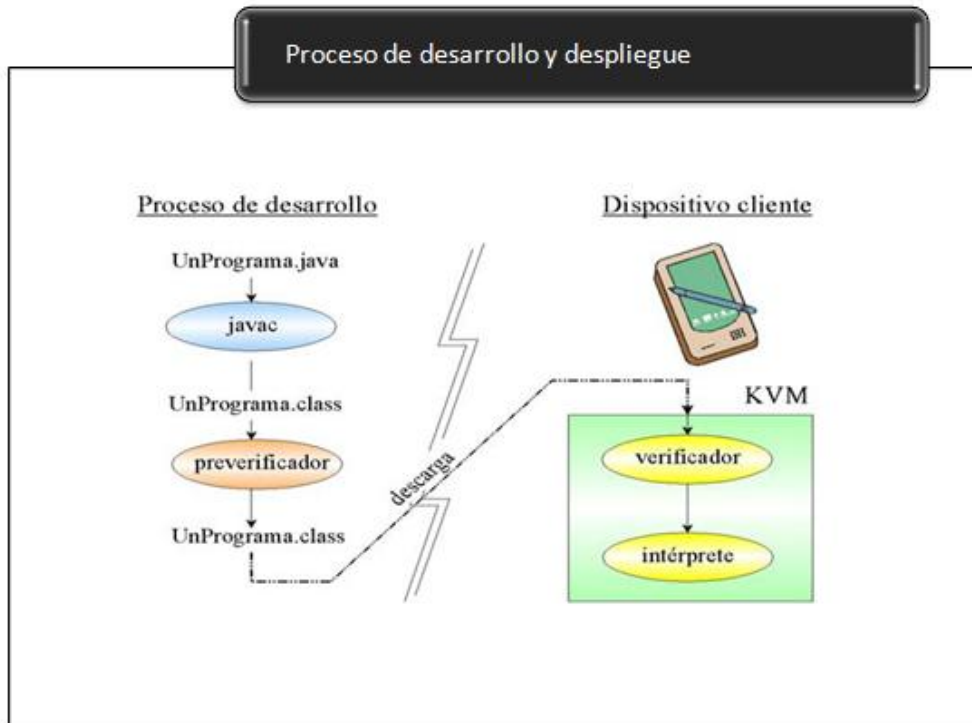


- No existe la finalización de instancias de clases. No existe el método *Object.finalize ( )*.
- No hay referencias débiles (reflexión).
- Limitada capacidad para el manejo de excepciones debido a que el manejo de éstas depende en gran parte de las APIs de cada dispositivo por lo que son éstos los que controlan la mayoría de las excepciones.

Aparte de la no inclusión de estas características, la verificación de clases merece un comentario aparte. El verificador de clases estándar de Java es demasiado grande para la KVM. De hecho es más grande que la propia KVM y el consumo de memoria es excesivo, más de 100Kb para las aplicaciones típicas. Este verificador de clases es el encargado de rechazar las clases no válidas en tiempo de ejecución. Este mecanismo verifica los *bytecodes* de las clases *Java* realizando las siguientes comprobaciones:

- Ver que el código no sobrepase los límites de la pila de la VM;
- Comprobar que no se utilizan las variables locales antes de ser inicializadas;
- Comprobar que se respetan los campos, métodos y los modificadores de control de acceso a clases.

Figura 24. **Proceso de desarrollo y despliegue de aplicaciones JME**



Fuente: java a tope: j2me (java 2 micro edition). Edición electrónica

Por esta razón los dispositivos que usen la configuración CLDC y KVM introducen un algoritmo de verificación de clases en dos pasos.

La KVM puede ser compilada y probada en 3 plataformas distintas:

- *Solaris Operating Environment*
- *Windows*
- *Palm OS*

#### 4.2.3.2. CVM *Compact Virtual Machine*

La CVM ha sido tomada como Máquina Virtual *Java* de referencia para la configuración CDC y soporta las mismas características que la Máquina Virtual de J2SE. Está orientada a dispositivos electrónicos con procesadores de 32 *bits* de gama alta y en torno a 2Mb o más de memoria RAM. Las características que presenta esta Máquina Virtual son:

- Sistema de memoria avanzado
- Tiempo de espera bajo para el recolector de basura
- Separación completa de la VM del sistema de memoria
- Recolector de basura modularizado
- Portabilidad
- Rápida sincronización
- Ejecución de las clases *Java* fuera de la memoria de sólo lectura (ROM)
- Soporte nativo de hilos
- Baja ocupación en memoria de las clases
- Proporciona soporte e interfaces para servicios en Sistemas Operativos de Tiempo Real
- Conversión de hilos *Java* a hilos nativos
- Soporte para todas las características de *Java2* v1.3 y librerías de seguridad, referencias débiles, Interfaz Nativa de *Java* (JNI), invocación remota de métodos (RMI), Interfaz de depuración de la Máquina Virtual (JVMDI)

#### **4.2.4. Configuraciones**

Una configuración es el conjunto mínimo de APIs *Java* que permiten desarrollar aplicaciones para un grupo de dispositivos. Estas APIs describen las características básicas, comunes a todos los dispositivos:

- Características soportadas del lenguaje de programación *Java*
- Características soportadas por la Máquina Virtual *Java*
- Bibliotecas básicas de *Java* y APIs soportadas

##### **4.2.4.1. Configuración de dispositivos con conexión CDC (*Connected Limited Configuration*)**

La CDC está orientada a dispositivos con cierta capacidad computacional y de memoria. Por ejemplo, decodificadores de televisión digital, televisores con *internet*, algunos electrodomésticos y sistemas de navegación en automóviles. CDC usa una Máquina Virtual *Java* similar en sus características a una de J2SE, pero con limitaciones en el apartado gráfico y de memoria del dispositivo.

Esta Máquina Virtual es la que se ha visto como CVM (*Compact Virtual Machine*). La CDC está enfocada a dispositivos con las siguientes capacidades:

- Procesador de 32 *bits*
- Disponer de 2 *Mb* o más de memoria total, incluyendo memoria RAM y ROM

- Poseer la funcionalidad completa de la Máquina Virtual *Java2*
- Conectividad a algún tipo de red

La CDC está basada en J2SE v1.3 e incluye varios paquetes *Java* de la edición estándar. Las peculiaridades de la CDC están contenidas principalmente en el paquete *javax.microedition.io*, que incluye soporte para comunicaciones HTTP y basadas en datagramas.

#### **4.2.4.2. Configuración de dispositivos limitados con conexión CLDC (*Connected Limited Device Configuration*)**

La CLDC está orientada a dispositivos dotados de conexión y con limitaciones en cuanto a capacidad gráfica, cómputo y memoria. Un ejemplo de estos dispositivos son: teléfonos móviles, buscapersonas (*Pagers*), PDAs, organizadores personales, etc. Ya se ha dicho que CLDC está orientado a dispositivos con ciertas restricciones. Algunas de estas restricciones vienen dadas por el uso de la KVM, necesaria al trabajar con la CLDC debido a su pequeño tamaño.

Los dispositivos que usan CLDC deben cumplir los siguientes requisitos:

- Disponer entre 160 Kb y 512 Kb de memoria total disponible. Como mínimo se debe disponer de 128 Kb de memoria no volátil para la Máquina Virtual *Java* y las bibliotecas CLDC, y 32 Kb de memoria volátil para la Máquina Virtual en tiempo de ejecución
- Procesador de 16 ó 32 *bits* con al menos 25 MHz de velocidad

- Ofrecer bajo consumo, debido a que estos dispositivos trabajan con suministro de energía limitado, normalmente baterías
- Tener conexión a algún tipo de red, normalmente sin cable, con conexión intermitente y ancho de banda limitado (unos 9 600 bps)

La CLDC aporta las siguientes funcionalidades a los dispositivos:

- Un subconjunto del lenguaje *Java* y todas las restricciones de su Máquina Virtual (KVM)
- Un subconjunto de las bibliotecas *Java* del núcleo
- Soporte para E/S básica
- Soporte para acceso a redes
- Seguridad

Un aspecto a tener muy en cuenta es la seguridad en CLDC. Esta configuración posee un modelo de seguridad *sandbox* al igual que ocurre con los *applets*. En cualquier caso, una determinada Configuración no se encarga del mantenimiento del ciclo de vida de la aplicación, interfaces de usuario o manejo de eventos, sino que estas responsabilidades caen en manos de los perfiles.

## **5. IMPLEMENTACIÓN DEL SISTEMA MÓVIL Y SERVIDOR SMS**

### **5.1. Desarrollo en el Cliente**

El punto de partida para toda aplicación móvil es el usuario final, el servicio que se va a proveer y las necesidades que se van a satisfacer mediante el servicio. Por lo tanto se destina una parte del tiempo al análisis de mercado, principalmente a la concepción de la idea y la identificación de nuestros clientes.

#### **5.1.1. Realizar la idea en J2ME**

El objetivo de este capítulo es mostrar el proceso de desarrollo de una aplicación móvil sin el uso de una metodología en especial, con lo cual se pretende guiar al lector mediante una secuencia de acciones en las cuales se pondrá en práctica los conceptos vistos en los capítulos anteriores.

En este capítulo se asume una postura emprendedora que permita al lector situarse en un entorno macro del problema tanto desde un punto de vista tecnológico, comercial, social, análisis de mercado y toma de decisiones financieras y legales. Al mezclar estos conceptos se obtendrá un producto final destinado para el mercado guatemalteco y la Universidad de San Carlos de Guatemala en sí.

La programación de aplicaciones en J2ME es una experiencia divertida y gratificante. Las capacidades de la plataforma *Java* están en constante evolución gracias a *Java Community Process* ([www.jcp.org](http://www.jcp.org)) que estandariza y crea nuevas APIs (como *Wireless Messaging API*) e incluso plataformas completas como la (*Mobile Service Architecture*), es por ello que la investigación constante nos permitirá realizar aplicaciones con contenidos modernos como la idea que se propone a continuación:

### **Diseño del Requerimiento**

La idea es muy importante pues representa los objetivos que se desean alcanzar mediante el desarrollo de la aplicación móvil.

Se necesita una aplicación para teléfonos móviles, que permita a los estudiantes de la Universidad de San Carlos de Guatemala realizar el proceso completo de inscripción de matrícula anual, el objetivo de la aplicación es la simplificación del proceso, optimización de los recursos y reducción del costo tanto para la Universidad como para el estudiante, por lo cual se compara el sistema actual con el sistema propuesto.



Tabla III. **Comparación de costos del sistema propuesto contra el sistema actual**

ACTUAL	J2ME
<p>El usuario ingresa a <i>Internet</i> para imprimir una boleta:</p> <p>Costo de <i>Internet</i> 30 minutos (promedio): Q.2.00</p> <p>Costo de impresión de la boleta (promedio): Q.0.50</p>	<p>El usuario envía un mensaje de texto mediante la aplicación y recibe el detalle de rubros, el precio total y los datos a incluir en la boleta de depósito costo Q.0.35 + Q.1.00 pago del servicio.</p>
<ol style="list-style-type: none"> <li>1. No es necesario imprimir una boleta exclusiva para el banco, el usuario puede entregar los datos de depósito al operador bancario o llevarlos en una hoja de papel que no sirva. Colabora con el medio ambiente, costos de impresión.</li> <li>2. No todos los estudiantes tienen acceso a <i>Internet</i> en su hogar por lo tanto el estudiante estaría ahorrando Q.0.65 por concepto de inscripción.</li> <li>3. La Universidad estaría cobrando Q.1.00 que cubrirá el costo de los mensajes de respuesta, mantenimiento y recuperación de inversión del equipo de mensajería.</li> </ol>	
<p>El usuario ingresa a nuevamente a <i>Internet</i> para confirmar su pago.</p> <p>Costo de <i>Internet</i> 30 minutos (promedio): Q.2.00</p>	<p>El usuario envía un mensaje de texto mediante la aplicación con la confirmación de pago y recibe la respuesta de confirmación de la Universidad, costo Q.0.35 + Q.1.00 pago del servicio.</p>

Fuente: elaboración propia

Un análisis más preciso puede considerar más información y a mayor detalle lo cual que permitiría evaluar la rentabilidad del proyecto.

Algunos elementos a considerar son:

- El porcentaje de alumnos real que no posee *Internet*
- Porcentaje real de estudiantes que poseen teléfono celular marca y modelo
- Costo promedio real por renta de *Internet* y el costo de impresión
- Tiempo mínimo promedio de renta, etc.

Para efecto de este documento y basados en el análisis de la tabla anterior, se nos indica que el costo se reduce para el estudiante, sin embargo la Universidad debe considerar el número de SMS a transmitir y el modelo de respuesta: Por medio de operador o por medio de Módem GPRS, en nuestro caso se asumirá que la respuesta se produce por medio de un *Módem* cuyo costo esta en USD \$.60.00 aproximadamente Q.480.00 inversión única.

Costo por transacción estudiante:

Costo del sistema actual:	Q.4.00
Costo del sistema propuesto:	Q.2.70

Beneficio para la Universidad:

Ingresos del sistema actual:	Q.0.00
Ingresos del sistema propuesto:	Q.2.00

## ROI

El retorno de inversión se alcanzaría con un total de 240 inscripciones, provenientes de los 120 mil estudiantes actualmente inscritos.

Por lo cual se puede intuir que el negocio es viable y con posibilidades de generar presupuesto a la Universidad el cual puede ser invertido en tecnología o infraestructura.

### 5.1.2. Prerrequisitos para la implementación en *Java ME*

Una aplicación móvil desarrollada en *Java (MIDlet)* necesita ser compilada, pre verificada y empaquetada. Para implementar una aplicación móvil en *Java* se necesita lo siguiente:

- *Java SDK* (no el *Java Runtime Environment*)
- Un IDE a elección del desarrollador, por ejemplo, *Eclipse* posiblemente con Herramientas de desarrollo móvil para *Java*, (MTJ), *NetBeans* con su paquete de movilidad, o *IntelliJ*
- Un emulador, por ejemplo, el *Wireless Toolkit*, el *Micro Emulator* o un Emulador específico del vendedor de la marca del móvil
- Dependiendo de la configuración, también puede ser necesario un ofuscador como *ProGuard*. Si se desea crear productos para el mercado empresarial o profesional puede ser útil una herramienta de construcción como *Maven* o *Ant*

- Es posible que desee utilizar J2ME *Polish* el cual es un *FrameWork* de fuente abierto el cual permite la creación de aplicaciones con mejor presentación y distribuir la misma a varios dispositivos

Una configuración que cumple con estos requisitos es una combinación de *Netbeans* 6.9 (versión más reciente)

Figura 25. Componentes de descarga *NetBeans*

NetBeans IDE incluye un emulador, Herramientas de desarrollo mediante MDA, lenguaje de programación Java, Dotfuscator y Proguard


Tecnologías *	Java SE	JavaFX	Java	Ruby	C/C++	PHP	All
NetBeans Platform SDK	•	•	•				•
Java SE	•	•	•				•
JavaFX		•	•				•
Java Web y EE			•				•
Java ME			•				•
Java Card™ 3 Connected			•				•
Ruby				•			•
C/C++					•		•
Groovy			•				•
PHP						•	•
Servidores incluidos							
GlassFish Server Open Source Edition 3.0.1			•	•			•
Apache Tomcat 6.0.26			•				•
	Download Libre, 54 MB	Download Libre, 122 MB	Download Libre, 234 MB	Download Libre, 86 MB	Download Libre, 36 MB	Download Libre, 31 MB	Download Libre, 310 MB

Fuente: netbeans.org <http://netbeans.org/downloads/index.html>

Y J2ME *Polish*:

Figura 26. Logo J2ME *Polish*

J2ME Polish es un framework utilizado para "Lustrar" aplicaciones móviles para darles un acabado profesional. Incluye ANT y MAVEN

**J2ME Polish** 

---

J2ME Polish is the answer to today's challenges in the mobile world. Extremely versatile and powerful, yet easy to use

Fuente: j2mepolish.org <http://www.j2mepolish.org/cms/topsection/home.html>

Una instalación completa y configuración de las herramientas está fuera del alcance de este documento, por lo cual se sugiere consultar la documentación respectiva de estas herramientas para tal caso.

Los principiantes a menudo seleccionan *NetBeans* como herramienta de desarrollo gracias que ya incluye las herramientas de desarrollo móvil en su paquete de instalación. También debe descargar y leer los *JavaDocs* de las tecnologías más importantes y las API disponibles. Puede descargar la mayoría de *JavaDocs* [www.jcp.org](http://www.jcp.org)

### 5.1.3. Implementar la aplicación

El entorno J2ME es bastante sencillo, la base está formada por la configuración del dispositivo conectado Limitado (CLDC) y el perfil de *Internet* para dispositivos móviles (MIDP) los cuales se explicaron en el capítulo 4. Sección 4.2, que son a la vez muy fáciles de entender.

Al igual que las aplicaciones clásicas de java, se extiende la librería *javax*. Adicionalmente se extiende la librería *microedition.midlet.MIDlet* y eso es todo, listo para empezar.

Se puede crear la interfaz de usuario de formas diferentes:

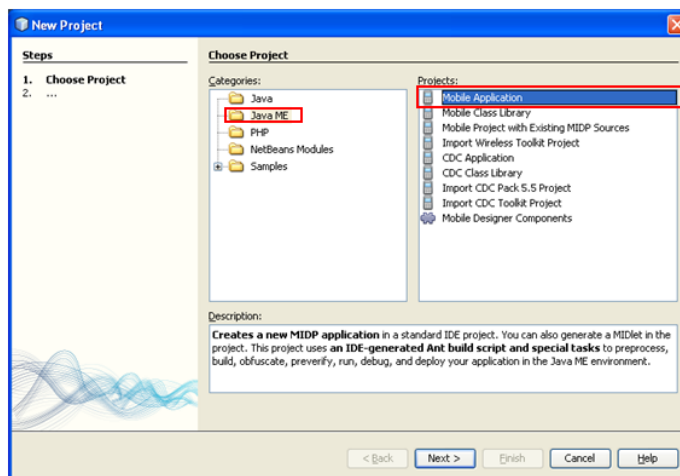
- Componentes LCDUI de alto nivel: utilizando los componentes que se encuentran en la librería: *javax.microedition.lcdui*, por ejemplo, *Form* o *List*

- Componentes LCDUI de bajo nivel: son del tipo: *javax.microedition.lcdui.Canvas* útiles para manipular la interfaz gráfica a nivel de pixel
- SVG: gráficos de vector escalables se define en JSR 287 [HTTP://jcp.org/es/jsr/detalle?id=287](http://jcp.org/es/jsr/detalle?id=287)

Para ejemplificar la fase de implementación se bautiza la aplicación propuesta como: “Confirmación Pago de Boletas de inscripción”

Crear un nuevo proyecto de tipo J2ME en el IDE *Netbeans*:

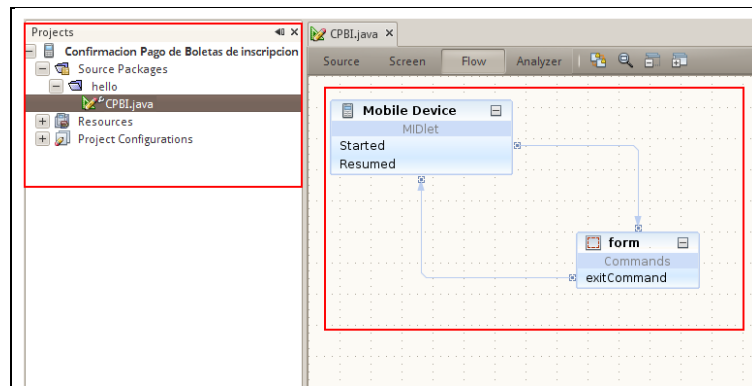
Figura 27. **Crear proyecto J2ME en *Netbeans***



Fuente: elaboración propia

El proyecto, sus componentes y el editor MDA:

Figura 28. **Editor Model Driven Architecture de Netbeans**



Fuente: elaboración propia

Importar las librerías *Javax* e implementar la clase *MIDlet*

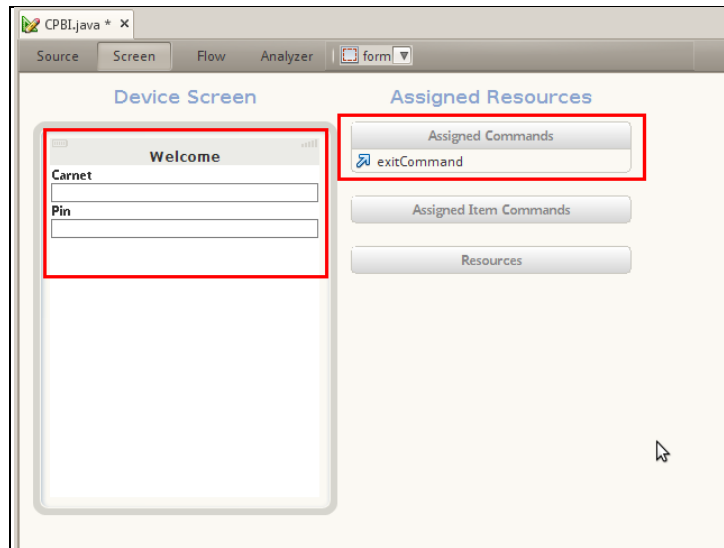
Figura 29. **Código de importación de paquetes**

```
6 package hello;
7
8 import javax.microedition.midlet.*;
9 import javax.microedition.lcdui.*;
10
11 /**
12  * @author ubuntu
13  */
14 public class CPBI extends MIDlet implements CommandListener {
15
16     private boolean midletPaused = false;
17
18     Generated Fields
19
20
21
22
23
24 /**
25  * The HelloMIDlet constructor.
26  */
27 public CPBI() {
28 }
```

Fuente: elaboración propia

Crear los componentes LCDUI

Figura 30. **Ambiente de diseño de componentes LCDUI en *Netbeans***



Fuente: elaboración propia

#### 5.1.4. Probar

Debido a la fragmentación, las pruebas sobre las aplicaciones son totalmente vitales. Las pruebas en emuladores no son sustituto para las pruebas en dispositivos reales, no es posible comparar una máquina con alta capacidad de procesamiento, memoria RAM y conexión de banda ancha a *Internet* contra las típicas funciones de un teléfono celular. Existen emuladores muy buenos como el emulador *Symbian* o el *BlackBerry* pero siempre es considerable realizar las siguientes pruebas en un entorno real:



- UI: la experiencia del usuario con la interfaz puede diferir cuando el producto es lanzado, siempre es preciso recordar que el producto es para teléfonos móviles, debe proveer facilidad de ingreso de datos, ser intuitiva y de fácil acceso.
- Multimedia: la reproducción de audio, vídeo y grabación puede variar dramáticamente entre los dispositivos y sus emuladores respectivos.
- Conectividad a *Internet*: establecer conexiones a *Internet* puede tomar bastante tiempo. La demora en la conexión y el ancho de banda depende de la red, su fuerza actual y el número de usuarios simultáneos.

#### **5.1.5. Compatibilidad, Cubrir todos los equipos**

Uno de los puntos fuertes del entorno *Java* en móviles es ser estándar, así que puede ser aplicado por varios proveedores que compiten entre sí. La mala noticia es que una norma o API debe ser interpretada, y la interpretación causa diferencias. Asimismo, diferentes implementaciones sufren de errores diferentes, lo que hace que las cosas no sean realmente fáciles. En la siguiente sección se esbozan estrategias diferentes para portar aplicaciones J2ME a todos los teléfonos y para diferentes plataformas.

Implementación directa: la mejor pero la más difícil solución es el codificar directamente para diferentes grupos de dispositivos y plataformas. Esto significa crear una aplicación J2ME para dispositivos MIDP, una aplicación nativa de *BlackBerry*, una aplicación nativa de *Windows Mobile*, una aplicación *Symbian*, una aplicación para el *iPhone*, una aplicación *Web OS*, y así sucesivamente.

Como es de imaginarse, este enfoque tiene el potencial de llevar la mejor experiencia al usuario, ya que realmente puede adaptarse la aplicación a cada plataforma. Al mismo tiempo sus costos de desarrollo se elevan por el cielo. Es aconsejable utilizar otra estrategia hasta que su primera idea ha sido probada con éxito.

**Mínimo Común Denominador:** se puede evitar muchos problemas de portabilidad mediante la limitación de las funcionalidades de la aplicación para el mínimo común denominador. En el mundo de J2ME, esto significa CLDC 1.0 y MIDP 1.0. Dependiendo de la región objetivo de la aplicación también se podría considerar la posibilidad de utilizar las extensiones de la tecnología Java para la Industria Inalámbrica (JTWI, JSR 185) o la arquitectura de servicios móviles (MSA, JSR 205). Las dos aproximaciones son compatibles con muchos dispositivos modernos y proporcionan mucha más potencia, aunque en algunas regiones como África, América del Sur o la India debe tenerse en cuenta que se estaría limitando el número de usuarios potenciales ya que estos estándares no están globalizados.

Con el mínimo común denominador normalmente es más fácil resolver problemas de portabilidad, ya que hay menos funcionalidades en uso. Sin embargo, la experiencia del usuario puede verse afectada al limitar la aplicación de esta manera. Además, este enfoque es obstáculo al momento de portar la aplicación para diferentes plataformas como *Android* o *iPhone*. En pocas palabras: usar el mínimo común denominador para aplicaciones sencillas, pero tener cuidado de utilizar este método para aplicaciones complejas.

Portar *Frameworks*: implementar *Frameworks* dentro de la aplicación permite adaptar automáticamente la aplicación a diferentes dispositivos y plataformas. Estos *Frameworks* usualmente se pueden portar llevando al proceso de compilación los siguientes elementos:

- La librería cliente que simplifica el desarrollo
- Compilar herramientas de cadena que convierte el código y los recursos en paquetes de aplicaciones
- Base de datos de dispositivos móviles que proporciona información sobre los dispositivos donde se instala la aplicación
- Compiladores de cruce que porten la aplicación a diferentes plataformas

En el mundo J2ME existen diversos *Frameworks* para elegir: *Xpress Suite* de *JavaGround* proporciona portabilidad en contra de compartir los ingresos. *Celsius* de *Mobile Distillery* se licencia por mes. *Bedrock* de *Metismo* proporciona un conjunto de compiladores cruzados en un pago anual. J2ME *Polish* de *Enough Software* está disponible tanto en versión GPL y comercial. En otras palabras: un buen *Framework* no se deshace de la fragmentación de dispositivos, pero hace que la fragmentación sea más manejable.

#### **5.1.6. Certificar la aplicación**

El estándar móvil de aplicaciones *Java* detecta cuando una aplicación esta firmada o esta sin firmar. Algunas funcionalidades del teléfono sólo están disponibles para las aplicaciones de confianza o firmadas.

¿Qué características se ven afectadas y qué ocurre si la aplicación no está firmada?, pero utiliza en gran medida una de estas características, depende en gran medida de la implementación de la aplicación. En un teléfono el usuario puede recibir un mensaje en el cual se le solicite permitir esta funcionalidad, este mensaje puede ser solicitado una sola vez, en otro se le pedirá cada vez que la función sea utilizada y en una tercera el usuario no será capaz de utilizar la función.

La mayoría de las implementaciones también se diferencian entre los certificados y las autoridades que han firmado el certificado para una aplicación:

- Las aplicaciones firmadas por el fabricante de un dispositivo disfrutan de el más alto nivel de protección y pueden acceder a todas las funciones del teléfono que deseen
- Las aplicaciones firmadas por *JavaVerified*, *Verisign* o *Thawte* gozan el nivel más bajo de seguridad

Las funcionalidades que más a menudo se encuentran restringidas por las aplicaciones son las de conectividad, como por ejemplo: SMS, GPRS, *Bluetooth*, infrarrojo, etc.

### **5.1.7. Distribuir la aplicación Java ME**

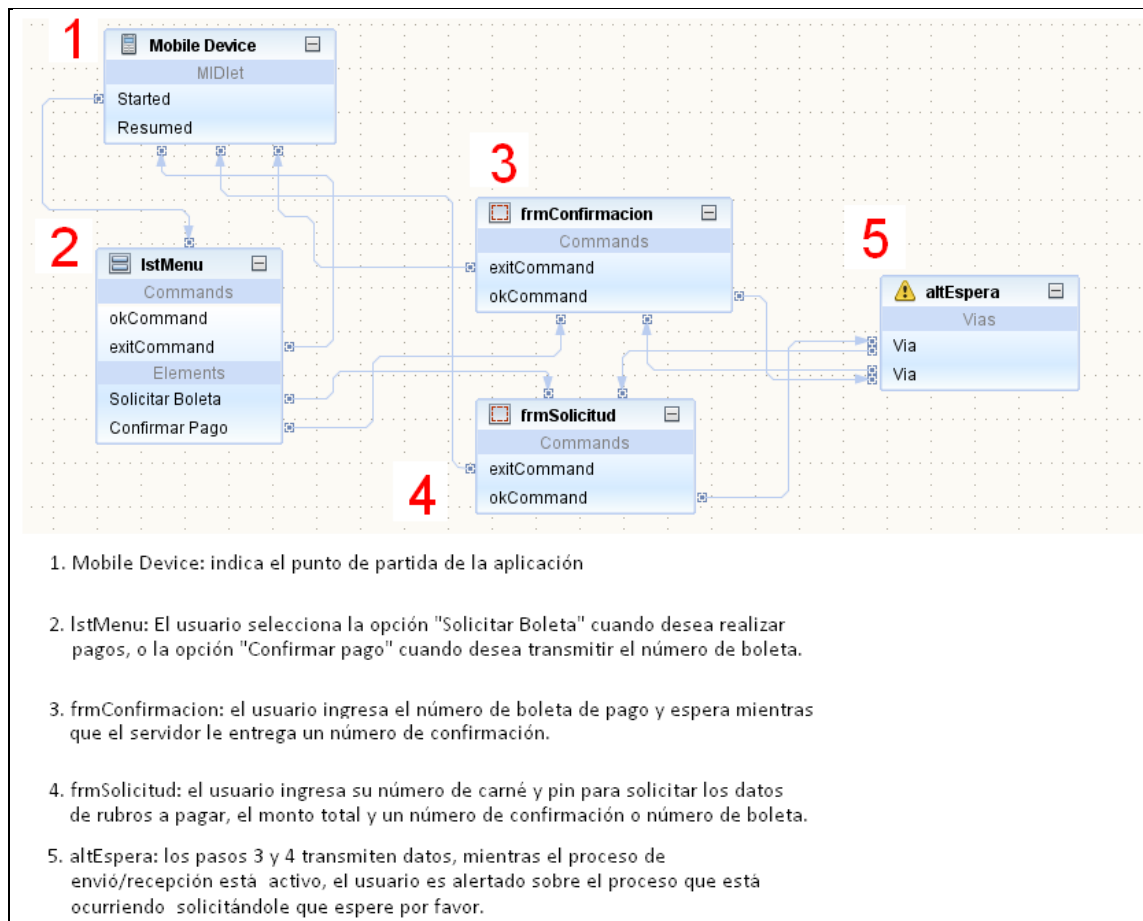
A diferencia del *iPhone*, es posible instalar aplicaciones J2ME directamente en el teléfono o bien a través de *Bluetooth* o por GPRS (OTA) utilizando un servidor *web*. Gracias a los almacenes de aplicación (*App Stores*), en la actualidad la distribución es más fácil que nunca.

Las *App Stores* se encargan de la administración de pagos, *hosting* y publicidad, y al hacerlo reciben cierta remuneración.

- *Ovi Store* se centra en usuarios de *Nokia* en todo el mundo y proporciona un 70% los ingresos por acción para el promotor (de las ventas brutas, netas de reembolsos y devoluciones, menos los impuestos aplicables y, en algunos casos, los costos de facturación del operador).
- *Java Store* planea ser vendedor exclusivo de aplicaciones basadas en *Java*.
- *Samsung Application Store* distribuye aplicaciones exclusivas para teléfonos *Samsung*.
- LG también distribuye aplicaciones en [www.lgapplication.com](http://www.lgapplication.com).
- *Handmark* proporciona un portador y proveedor independiente de *móvil* *Stores*.
- *GetJar* es uno de los más antiguos distribuidores para móviles con aplicaciones gratuitas.

## 5.2. Diagrama de aplicación "Confirmación Pago de Boletas de Inscripción"

Figura 31. MDA de *Netbeans* con los componentes de la aplicación Móvil



Fuente: elaboración propia

### 5.2.1. Caso de uso extendido "Confirmación Pago de Boletas de Inscripción"

- A. El usuario es presentado con un menú en el cual se le solicitará la selección de dos opciones:  
Solicitar Boleta y Confirmar Pago

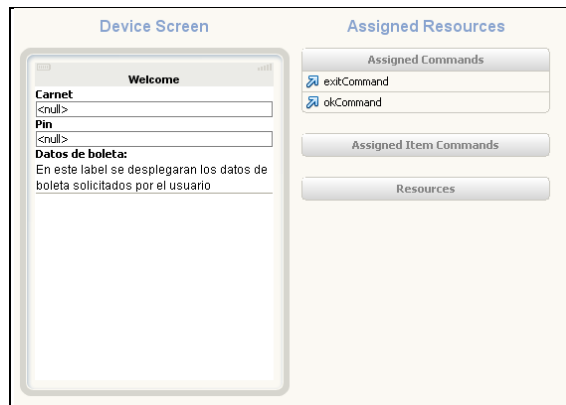
Figura 32. Menú principal desde el editor de componentes LCDUI



Fuente: elaboración propia

- B. Si selecciona Solicitar Boleta, el usuario es presentado con dos cajas de ingreso de datos, una para el número de carné y otra para el *pin*. El usuario presiona el botón *Ok* para transmitir los datos o el botón *Exit* para abortar la operación.

Figura 33. **Solicitud de datos de boleta desde el editor de componentes LCDUI**



Fuente: elaboración propia

- C. Al seleccionar *Ok* la aplicación captura el número de carnét y *pin*, y envía un mensaje de texto SMS al servidor de la Universidad el cual se encarga de capturar el mensaje de texto y recuperar las variables de consulta. Se despliega la pantalla de alerta mientras se recibe respuesta del servidor.

Figura 34. **Código para capturar el ingreso de usuario**

```

113     }
114     } else if (displayable == frmSolicitud) {
115         if (command == exitCommand) {
116             // write pre-action user code here
117             exitMIDlet();
118             // write post-action user code here
119         } else if (command == okCommand) {
120             consultarBoleta = txtCarne.getString() + txtPin.getString();
121             switchDisplayable(getAltEspera(), getFrmSolicitud());
122             // write post-action user code here
123         }
124     } else if (displayable == lstMenu) {

```

Fuente: elaboración propia

- D. El servidor recibe la solicitud SMS, recupera los datos de consulta y realiza un *login* en la cuenta del estudiante, se consulta a la base de datos y recuperar la información de rubros pendientes de pago.



- E. Crea una cadena de texto con el mensaje a mostrar al estudiante y envía un mensaje de texto SMS de vuelta al teléfono celular del solicitante.
- F. Al recibir la señal de respuesta del servidor, la aplicación finaliza la pantalla de alerta y despliega el contenido del mensaje de texto con la información que se ha consultado en el servidor.
- G. Si se selecciona Confirmar Pago, el usuario es presentado con una caja de ingreso de datos para introducir el número de boleta de pago. El usuario presiona el botón *Ok* para transmitir los datos o el botón *Exit* para abortar la operación.

Figura 35. **Ingreso de datos para confirmar el pago de boleta desde el editor de componentes LCDUI**



Fuente: elaboración propia

- H. Al seleccionar Ok la aplicación captura el número de boleta de pago, y envía un mensaje de texto SMS al servidor de la Universidad el cual se encarga de capturar el mensaje de texto y recupera las variables de consulta. Se despliega la pantalla de alerta mientras se recibe respuesta del servidor.
  
- I. El servidor recibe la solicitud SMS, recupera los datos de consulta y realiza un *login* en la cuenta del estudiante, se consulta a la base de datos y se verifica que el número de solicitud concuerde, se graba un registro en la base de datos indicando que el estudiante realizó el pago de los rubros pendientes, crea una cadena de texto con el mensaje a mostrar al estudiante y envía un mensaje de texto SMS de vuelta al teléfono celular del solicitante.
  
- J. Al recibir la señal de respuesta del servidor, la aplicación finaliza la pantalla de alerta y despliega el contenido del mensaje de texto con la información que se ha consultado en el servidor.

### **5.3. Servicios del proveedor de contenido**

Los servicios del proveedor de contenido se almacenan del lado del servidor, y reciben información del *Gateway*. Estos servicios pueden ser interfaces realizadas en cualquier lenguaje de programación como por ejemplo: Ejecutables de *Windows* (*exe*), archivos PHP o *Servlets* de *Java*. El *Gateway* se encarga de recibir el mensaje de texto del cliente y traducir esta información en parámetros que después serán utilizados para ejecutar la interfaz y poner a trabajar el servicio, ejemplos de ejecución sería:

- Miprograma.exe 200217925 1234
- HTTP://localhost/mipagina.php?carne=200217925&pin=1234

El programa "Confirmación Pago de Boletas de Inscripción" cuenta con dos acciones que requieren la funcionalidad de envío de mensajes de texto SMS:

- Confirmación de Boleta: cuando el usuario ha pagado la boleta y desea confirmar el pago.
- Solicitud de datos de Boleta: cuando el usuario desea consultar los rubros pendientes de pago.

En ambos casos, la aplicación envía un mensaje de texto el cual viaja a la central de mensajes SMSC de la empresa que provee el servicio al usuario y posteriormente este mensaje viaja a otros SMSC o bien directamente al destinatario. Según lo expuesto en el capítulo 2, sección 2.5. Proceso de transmisión de un SMS entre terminales.

Para montar un servidor que reciba solicitudes vía mensaje de texto se necesita del siguiente *hardware* y *software*:

### ***Hardware***

- Un PC con sistema operativo *Linux* de preferencia distribución *Debian*, *Ubuntu* o *Redhat*

- Un módem GSM con conexión de puerto serial, actualmente cualquier teléfono celular funciona como módem GSM a través del cable USB que viene con el dispositivo, pero es más confiable utilizar un *módem* ya que los teléfonos celulares tienden a bloquearse
- Una tarjeta SIM de cualquier proveedor para insertarla en el *módem*

### **Software**

- *Software Gateway SMS* el cual se encargará de traducir los mensajes de texto SMS que lleguen al *módem* y de ejecutar los servicios
- Un servidor de páginas dinámicas como lo es *Apache* o *Tomcat*
- Un programa escrito en un lenguaje de programación dependiendo del servidor de páginas seleccionado, este programa será la interface o servicio
- Una base de datos (si es que el servicio requiere consultar o actualizar información)

#### **5.3.1. Instalación del Gateway SMS *Kannel***

*Kannel* es un *software Gateway SMS*, la función de un *Gateway* ya se ha explicado en el capítulo 3, sección 3.1. *Gateway* de mensajería SMS a continuación se utilizará *Kannel* para recibir las peticiones y poner en marcha los servicios de la aplicación del lado del servidor. En esta sección se explica cómo compilar el *Gateway* y dejar instalados los componentes necesarios en el lugar adecuado.

## Instalación de *Kannel* utilizando paquetes DEB

Los paquetes de *kannel* están disponibles en el repositorio oficial de *Debian* ([HTTP://packages.debian.org/kannel](http://packages.debian.org/kannel)) pero se recomienda obtener el paquete con extensión *.DEB* directamente de la página de *Kannel* [www.kannel.org](http://www.kannel.org) para obtener la última versión.

Antes de iniciar con la instalación se debe verificar que esté instalado *libxml2* en el sistema, utilizando la siguiente instrucción en la línea de comando.

```
# dpkg -l libxml2
```

### A. Instalar o actualizar *Kannel* desde los repositorios de *Kannel* o *Debian*

- Instalar o actualizar el paquete: `# Apt-get install kannel`

Vea [http://kannel.org/download.shtml#debian\\_repository](http://kannel.org/download.shtml#debian_repository) para información acerca de la lista de repositorios *kannel*.

### B. Instalar o actualizar *Kannel* desde un archivo *.DEB*

- Descargar el archivo binario DEB desde el sitio de *Kannel*
- *Login* como *root*: `# su -`
- Instalar o actualizar con el paquete DEB:  
`# dpkg -i kannel-VERSION.deb`

### C. Eliminar *Kannel*

- Login como root: `# su -`
- Eliminar el paquete conservando los archivos de configuración  
`# dpkg --remove kannel`
- Eliminar el paquete y los archivos de configuración  
`# dpkg --purge kannel`

Después de la instalación se puede iniciar *Kannel* con la siguiente instrucción, *Kannel* se ejecutará como un servicio:

```
# /etc/init.d/kannel start
```

O para detener el servicio:

```
# /etc/init.d/kannel stop
```

Para ejecutar *Kannel* como un *Gateway* SMS es necesario crear un archivo de configuración el cual será explicado en la siguiente sección.

La documentación de *Kannel* se instala en `/usr/share/doc/kannel/`. Dentro del cual se encuentra un archivo de nombre `control.html`. El cual contiene un ejemplo sobre cómo administrar *Kannel* mediante HTTP.

*Kannel* es una herramienta que permite interactuar con páginas dinámicas PHP, *Java* o cualquier otra tecnología *web*, por lo tanto las instrucciones para enviar y recibir mensajes se pueden configurar por medio de instrucciones

HTTP. Motivo por el cual se propone utilizar un servicio basado en páginas *web* JSP de *Java* que se ejecuten en un servidor *Tomcat*.

### 5.3.2. Configuración de *Kannel* como un *Gateway SMS*

El *software Kannel* tiene dos funcionalidades, una como *Gateway WAP* y otra como *Gateway SMS*, a continuación se describe los pasos de configuración sobre los componentes *Gateway SMS* de *Kannel*.

#### Componentes requeridos

Para configurar *Kannel* como un *Gateway SMS*, además de una máquina ejecutando *Kannel*, se necesita acceso a una o varias centrales de mensajería de texto (SMSC) de uno o varios operadores de telefonía móvil.

Si no se cuenta con acceso al SMSC de un operador debido a los costos discutidos en el capítulo 3, aun es posible utilizar el *software Kannel* haciendo uso de un teléfono celular o un *módem GSM* el cual será configurado como SMSC virtual.

Adicionalmente al SMSC (real o virtual) es necesario tener un servidor que se encargue de administrar los servicios requeridos por los SMS recibidos.

Básicamente el servidor deberá poseer un serie de *Scripts* o bien programas CGI-BEANS los cuales darán servicio a las peticiones HTTP requeridas por *Kannel*. Estos servicios también tendrán la capacidad de iniciar mensajería SMS gracias a la interfaz SMSBOX provista por *Kannel* vía HTTP.

## El archivo de configuración

Para configurar *Kannel* como *Gateway SMS* es necesario crear un archivo de configuración. El archivo debe ser de texto plano sin importar su extensión o codificación de caracteres, puede poseer cualquier nombre y estar ubicado en cualquier sector del disco duro.

El objetivo del archivo de configuración es indicarle a *Kannel* por medio de las variables, cuales son los métodos de conexión con el SMSC, cuales son los servicios de los cuales dispone, cuales son los usuarios administradores y cuáles son los usuarios de envío de SMS.

El archivo de configuración de *Kannel* es muy similar a los archivos de configuración de servidores HTTP, como *Tomcat*, *Apache*, *GlassFish*, etc. Únicamente que no viene con una configuración por defecto, cada administrador de equipo debe de configurar su propio archivo desde el comienzo.

El archivo de configuración está dividido por secciones denominadas "Grupos", cada grupo se encarga de realizar una tarea específica y cada grupo cuenta con determinadas "Variables de Grupo" las cuales deben ser configuradas de forma correcta para realizar cada tarea.

El archivo de configuración debe contar con las siguientes secciones configuradas:



## CORE

Establece la configuración básica del sistema, activa el software cliente que interactúa con el equipo servidor a nivel de sistema operativo. Indica cuáles son los puertos de comunicación, el usuario administrador y la ubicación de los archivos de LOG.

Figura 36. Configuración de variables del grupo Core

```
34 # CORE
35 #
36 # There is only one core group and it sets all basic settings
37 # of the bearerbox (and system). You should take extra notes on
38 # configuration variables like 'store-file' (or 'store-dir'),
39 # 'admin-allow-ip' and 'access.log'
40
41 group = core
42 admin-port = 13101
43 smsbox-port = 13102
44 admin-password = connext
45 #status-password = foo
46 #admin-deny-ip = ""
47 #admin-allow-ip = ""
48 #log-file = "/LOGS/MODEM/kannel.log"
49 log-level = 1
50 box-deny-ip = "*. *.*.*"
51 box-allow-ip = "127.0.0.1"
52 #unified-prefix = "+358,00358,0;+,00"
53 #access-log = "/kannel/access.log"
54 #store-file = "kannel.store"
55 #ssl-server-cert-file = "cert.pem"
56 #ssl-server-key-file = "key.pem"
57 #ssl-certkey-file = "mycertandprivkeyfile.pem"
58
```

Fuente: elaboración propia

Las variables a configurar son:

- *Group*: indica el nombre del grupo, su valor es Core. Línea 41
- *Admin-port*: número de puerto asignado a *Kannel* para trabajar directamente con el sistema operativo, puede ser cualquier número de puerto libre

- *SMSbox-port*: número de puerto asignado para el envío/recepción de mensajes SMS traducidos por *kannel* hacia la interface *SMSbox*
- *Admin-password*: *password* de acceso a operaciones administrativas del sistema como por ejemplo. Detener *Kannel*, ingresar al *log* de eventos de *Kannel* y otros
- *Log -file* y *log-level*: indica la ubicación del archivo de *log* y el nivel de detalle a almacenar en el mismo
- *Box-deny-ip* y *box-allow-ip*: indican las direcciones IP desde las cuales se puede tener acceso al servidor que ejecuta *Kannel*, por defecto se le permite acceso al servidor local: 127.0.0.1

## **SMSC CONNECTIONS**

Maneja el protocolo del SMSC, en este caso un módem GSM que actúa como un SMSC virtual. La conexión SMSC es la que se encarga de enviar y recibir SMS entre terminales Mviles. Indica el nombre de usuario del SMSC, la dirección IP del SMSC, el puerto de acceso, para el caso del SMSC virtual, solicita los parámetros AT necesarios para indicarle al módem que su funcionalidad será administrada por el *Gateway*.

Figura 37. Configuración de variables del grupo SMSC

```
59 #-----
60 # SMSC CONNECTIONS
61 #
62 # SMSC connections are created in bearerbox and they handle SMSC specific
63 # protocol and message relaying. You need these to actually receive and send
64 # messages to handset, but can use GSM modems as virtual SMSCs
65 #
66 group = smsc
67 smsc = at
68 modemtype = multitech
69 device = /dev/ttyS0
70 speed = 9600.
71
72 group = modems
73 id = multitech
74 name = "multitech"
75 detect-string = "multitech"
76 init-string = "AT+CNMI=2,3,0,0,0"
77 broken = true
```

Fuente: elaboración propia

Las variables a configurar son:

- *Group*: indica el nombre del grupo, su valor es SMSC. Línea 66
- *SMSC*: es un *Id* opcional que puede servir en otras secciones para hacer referencia a los parámetros de configuración SMSC, tiene el mismo uso que un alias, línea 67
- *Modemtype*: las características de los *módem* varían dependiendo del fabricante, en esta variable se indica la marca del fabricante del *módem* GSM o teléfono celular conectado a la computadora para recibir los mensajes SMS línea 68

- *Device*: indica el tipo de conexión o puerto serial/USB donde está conectado el módem a la computadora. Dado que el sistema operativo es *Linux*, se le indica la ubicación del archivo de configuración del dispositivo E/S esto nos obliga a que el archivo de configuración sea ejecutado por un usuario con permisos a la carpeta */dev* línea 69
- *Speed*: indica la velocidad en bits de procesamiento del *módem*, indicar la velocidad exacta optimiza el rendimiento línea 70. Dado que se utiliza un SMSC virtual, se debe de configurar el grupo *módems*, el cual sirve para indicarle a *Kannel* de alguna configuración inicial que deba hacer en el dispositivo
- *Group*: indica el nombre del grupo, su valor es *módems*. Línea 72
- *Id*: es un *Id* opcional que puede servir en otras secciones para hacer referencia a los parámetros de configuración de este *módem* en específico, funcionamiento similar a un alias
- *Name*: el nombre que se le asignará a esta configuración, es útil para detectar problemas en los archivos de *log* ya que la información se agrupa por el nombre de los dispositivos
- *Detect-string*: es un segundo *Id* para hacer referencia al *módem* tal y como lo hace *Id*

- *Init-string*: si el *módem* requiere de una configuración inicial, es decir enviarle un comando AT que permita inicializar el *módem*, por lo regular es necesario para indicarle al *módem* que *Kannel* tomará control exclusivo y que debe enviar y recibir información a los puertos designados en el grupo *Core*
- *Broken*: algunos *módem* necesitan que esta variable sea configurada con el valor *true* para funcionar, algunos *módems* GSM tienden a bloquear su puerto o procesamiento de SMS cuando han recibido cierta cantidad de mensajes en determinado intervalo de tiempo, este comportamiento puede analizarse de forma análoga al *spam* de correo electrónico. *Kannel* le indica al *módem* con esta variable que al momento de ocurrir el bloqueo reinicie el dispositivo

### ***SMSBOX SETUP***

En esta sección se configura el cliente que decide qué hacer con los SMS entrantes, éste se comunica vía HTTP con las interfaces locales diseñadas para interactuar con los mensajes de los clientes de mensajería corta.

Figura 38. Configuración de variables del grupo **SMSBOX**

```
79 #-----
80 # SMSBOX SETUP
81 #
82 # Smsbox(es) do higher-level SMS handling after they have been received from
83 # SMS centers by bearerbox, or before they are given to bearerbox
84 #for delivery
85 #
86 group = smsbox
87 bearerbox-host = localhost
88 sendsms-port = 13013
89 global-sender = 1551
90 #log-file = "/LOGS/MODEM/smsbox.log"
91 #sendsms-chars = "0123456789 +-"
92 #log-file = "/kannel/smsbox.log"
93 #log-level = 0
94 #access-log = "/kannel/accessbox.log"
```

Fuente: elaboración propia

Las variables a configurar son:

- *Group*: indica el nombre del grupo, su valor es *SMSbox*. Línea 86
- *Bearerbox-host*: le indica a *Kannel* la máquina donde se está ejecutando la interfaz *Bearerbox*
- *SendSMS-port*: indica el número de puerto que utiliza la interfaz *Bearerbox* para comunicarse con la interfaz *SMSbox* para el envío de mensajes SMS
- *Global-sender*: se configura para enviar un número corto como remitente del mensaje

## **SEND-SMS USERS**

Para enviar mensajes de texto desde la computadora haciendo uso de *Kannel*, es necesario definir un usuario específico para esta tarea, para enviar SMS *Kannel* utiliza interfaces de alto nivel que operan bajo el protocolo HTTP, de esta manera se puede enviar un mensaje de texto a cualquier terminal con tan sólo ejecutar una petición *Web*.

A esta operación se le conoce como *Push SMS* y se realiza escribiendo la siguiente URL en el navegador *web*:

- [HTTP://kannel.machine:13013/cgi-bin/sendSMS?username=tester&password=foobar...](http://kannel.machine:13013/cgi-bin/sendSMS?username=tester&password=foobar...)

Figura 39. **Configuración de variables del grupo *SENDSMS-USER***

```
96 #-----
97 # SEND-SMS USERS
98 #
99 # These users are used when Kannel smsbox sendsms interface is used to
100 # send PUSH sms messages, i.e. calling URL like
101 # http://kannel.machine:13013/cgi-bin/sendsms?username=tester&password=foobar
102 #
103 group = sendsms-user
104 username = connext
105 password = connext
106 #user-deny-ip = ""
107 #user-allow-ip = ""
```

Fuente: elaboración propia

Las variables a configurar son:

- *Group*: indica el nombre del grupo, su valor es *sendSMS-user*. Línea 103
- *Username*: nombre que identifica al usuario en los parámetros de la URL para iniciar *push SMS*
- *Password*: contraseña para el envío de SMS en los parámetros de la URL para iniciar *push SMS*

## SERVICES

Son los responsables de responder a los mensajes de texto enviados por el cliente, se activan mediante una frase específica y se comunica con el sistema local a través del protocolo HTTP.

Figura 40. Configuración de variables del grupo **SMS-SERVICE**

```
110 #-----
111 # SERVICES
112 #
113 # These are 'responses' to sms PULL messages, i.e. messages arriving from
114 # handsets. The response is based on message content. Only one sms-service is
115 # applied, using the first one to match.
116 #
117 #
118 #-----
119 group = sms-service
120 keyword = usacc
121 post-url="http://localhost:8084//servicios/usac/confirmacionPago.jsp?cp=%r&sender=%p"
122 accept-x-kannel-headers = true
123 max-messages = 1 #if max-messages is set to 0, no reply is sent, except for error messages.
124
125 #-----
126 group = sms-service
127 keyword = usacs
128 post-url="http://localhost:8084//servicios/usac/solicitudDatos.jsp?nc=%s&pin=%s&sender=%p"
129 accept-x-kannel-headers = true
130 max-messages = 1 #if max-messages is set to 0, no reply is sent, except for error messages.
```

Fuente: elaboración propia



El sistema a desarrollar cuenta con dos servicios: la consulta de rubros de pago y la confirmación de pago.

- *Group*: indica el nombre del grupo, su valor es SMS-service, línea 119
- *Keyword*: indica una palabra clave que permite identificar los servicios prestados por *Kannel*, esta debe ser incluida en el mensaje de texto para identificar qué servicio se desea activar
- *Post-URL*: indica el URL del servicio que ejecutará el SMS, el servicio reside en una página dinámica del servidor *web*, *Kannel* se encarga de ejecutar la página JSP mediante una interfaz HTTP, los valores de los parámetros están precedidos por el símbolo %, cada parámetro es una palabra en el mensaje de texto separada por un espacio en blanco

Para activar el servicio de solicitud de datos de pago se necesitan dos variables: el número de carné y el número de pin, así mismo el nombre del servicio, es decir enviar un mensaje de texto con la siguiente estructura:

- SMS: << *Keyword* >> << carné >> <<pin>>
- SMS: usacs 200217925 1234

El encargado de enviar el mensaje de texto será el programa J2ME, el código de envío se modifica de la siguiente manera:

Figura 41. **Código fuente de captura del texto a enviar por la aplicación**

```
consultarBoleta = "usacc"+" "+txtCarne.getString() +" "+ txtPin.getString();
```

Fuente: elaboración propia

*Kannel* recibe el mensaje de texto SMS y mediante el *Bearerbox* recupera los parámetros de consulta y envía al *SMSbox* el valor de los mismos buscando el servicio y sustituyendo los valores de los parámetros de la siguiente manera:

Figura 42. **Variable Post-URL**

```
post-url="http://localhost:8084//servicios/usac/solicitudDatos.jsp?nc=%s&pin=%s&sender=%p"
```

Fuente: elaboración propia

- *Nc=%s* es la primera palabra después de la palabra clave
- *Pin=%s* es la palabra siguiente es decir el orden de los parámetros se toma en la misma secuencia del mensaje enviado
- *%p* es una constante que indica el número del teléfono que envió la solicitud SMS

De este modo *Kannel* ejecuta en el servidor una llamada HTTP que ejecuta el código del *script* JSP.

Adicionalmente se debe configurar un servicio que por defecto se ejecute cuando *Kannel* no es capaz de encontrar una coincidencia, es decir, cuando el usuario no escriba correctamente la palabra clave en su mensaje de texto, en caso de que la consulta se haga manualmente vía la bandeja de salida de mensajes del celular.

Figura 43. **Configuración de servicio *Default***

```
153 #-----  
154 # There should be always a 'default' service. This service is used when no  
155 # other 'sms-service' is applied.  
156 group = sms-service  
157 keyword = default  
158 text = "No se ha especificado un servicio, los servicios disponible son usacc y usacs"
```

Fuente: elaboración propia

*Kannel* enviará un mensaje de texto con el contenido de la variable **text** al usuario que inicio el servicio.

### 5.3.3. Ejecución de *Kannel* como *Gateway SMS*

Para poner a funcionar *Kannel* como *Gateway SMS* se necesita ejecutar la interfaz *Bearerbox* y la interfaz *SMSbox* indicándole a ambas interfaces la ubicación del archivo de configuración:

- `#!/Bearerbox /home/Ubuntu/usac/kannel-config`
- `#!/SMSbox /home/Ubuntu/usac/kannel-config`

Si el archivo de configuración es correcto, entonces la ejecución de *Bearerbox* y *SMSbox* no dará errores y el sistema estará listo para enviar y recibir mensajes de texto. Esto dicho sintácticamente, se debe asegurar que la dirección *web* del servicio esté disponible y funcione correctamente.

Así mismo que el *módem* cuente con tarjeta SIM y que la tarjeta esté activada y con saldo para enviar mensajes de respuesta.

#### **5.3.4. Enviar mensajes usando la interfaz HTTP**

Es posible administrar *Kannel* por medio de un navegador *web* mediante URL por ejemplo para verificar que *Kannel* se comunica con el *módem* se puede enviar un mensaje de prueba con la siguiente URL en el navegador *web*.

```
http://localhost:13013/cgi-bin/sendSMS?username=connext&password=123&to=52098888&text=texto
```

Técnicamente se está ejecutando una petición HTTP vía *Get* en la cual toda la información está siendo contenida en la URL y enviada al *SMSbox* para que el *Gateway* envíe el mensaje a través del *módem*.

#### **5.4. Crear los servicios del proveedor de contenido**

La siguiente etapa es la creación del programa que se encargará de ejecutar la acción en el servidor. Para lo cual se ha seleccionado el servidor de páginas *web Tomcat* y se ha bautizado a los servicios como *confirmacionPago.JSP* y *solicitudDatos.JSP*.

En el ejemplo de la sección anterior se ha realizado una llamada al programa *solicitudDatos.JSP* de la siguiente manera:

Figura 44. **Variable *Post-URL* con los parámetros sustituidos por valores**

```
post-url="http://localhost:8084//servicios/usac/solicitudDatos.jsp?nc=200217925&pin=1234&sender=%p"
```

Fuente: elaboración propia

Es decir, se ha solicitado la información de rubros pendientes de pago para el número de carné 2002 17925, el cual se firma al sistema con el número de pin 1234 así mismo nos proporciona su número de teléfono para que se le responda.

El *script* JSP debe de realizar lo siguiente:

- Iniciar sesión en la base de datos con el nombre de usuario y *pin*;
- Lanzar una consulta a la tabla de la base de datos que contenga la información de rubros pendientes de pago;
- Recuperar la consulta como texto y enviarla al usuario.

*Kannel* proporciona 2 formas de enviar mensajes de texto, un método explícito mediante la interfaz HTTP:

```
http://localhost:13013/cgi-bin/sendSMS?username=connext&password=123&to=52098888&text=texto
```

Para ejecutar HTTP en java no basta con la ejecución del método *response.redirect (URL)*, es necesario utilizar una librería especial de nombre *HTTPClient* la cual al ser instanciada permite ejecutar HTTP en el *script* de la siguiente manera:

Figura 45. **Código fuente de la conexión HTTP por medio de la librería *HTTP\_CLIENT***

```
public String urlNavigate(String URL){  
    /* Crear la instancia del cliente HttpClient.*/  
    HttpClient client = new HttpClient();  
  
    /* Create la instancia GET*/  
    GetMethod method = new GetMethod(URL);  
  
    try {  
        /* Ejecutar la llamada*/  
        int statusCode = client.executeMethod(method);  
  
        if (statusCode != HttpStatus.SC_OK) {  
            return("Method failed: " + method.getStatusLine());  
        }  
  
        /* Read the response body.*/  
        byte[] responseBody = method.getResponseBody();  
  
        /* Deal with the response.*/  
        /* Use caution: ensure correct character encoding and is not  
        return (new String(responseBody));  
    } catch (HttpException e) {  
        return("Fatal protocol violation: " + e.getMessage());  
    }  
}
```

Fuente: elaboración propia

Otros lenguajes como PHP pueden requerir métodos menos complejos pero que obtienen el mismo resultado, hay que recordar que *Kannel* funciona mediante peticiones HTTP por lo que un programa no puede hacer uso de el método de envío de SMS re direccionando la dirección URL.

El segundo método consiste en efectuar una salida estándar del tipo:

Figura 46. **Código fuente de la salida Estándar**

```
public void urlNavigate() {  
    System.out.println("Salida Estandar");  
}
```

Fuente: elaboración propia

*Kannel*, envía como respuesta al usuario todo lo que el *script* escriba en la página *web*, este método es menos complejo y por lo general más útil porque lo que se desea enviar es un mensaje de texto al remitente.

La desventaja es que no se puede especificar el número de teléfono de destino, siempre toma el número del remitente.

Adicionalmente el servicio configurado en *Kannel* debe establecer la siguiente variable de grupo en 1:

Figura 47. **Configuración de la variable *Max-Messages***

```
150 accept-x-kannel-headers = true
151 max-messages = 1 #if max-messages is set to 0, no reply is sent, except for error messages.
152
```

Fuente: elaboración propia

Ya que si se coloca en 0, el segundo método no funcionará.

#### 5.4.1. Elegir el servidor *web*

Existen varios servidores de páginas HTTP los cuales pueden ser utilizados para montar los servicios *web* que se ofrecen a los clientes SMS. Lo importante no es el servidor de páginas, sino más bien el servicio que se ofrece y los conocimientos en los lenguajes de programación:

## **Apache**

El servidor HTTP *Apache* es un servidor web HTTP de código abierto para plataformas *Unix* (BSD, GNU/Linux, etc.), *Microsoft Windows*, *Macintosh* y otras, que implementa el protocolo HTTP/1.1 *Apache* es usado principalmente para enviar páginas *web* estáticas y dinámicas en la *World Wide Web*.

Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a *Apache*, o que utilizarán características propias de este servidor *web*.

## **Tomcat**

También llamado *Jakarta Tomcat* o *Apache Tomcat*, funciona como un contenedor de *Servlets* desarrollado bajo el proyecto *Jakarta* en la *Apache Software Fundación*. *Tomcat* implementa las especificaciones de los *Servlets* y de *JavaServer Pages* (JSP) de *Sun Microsystems*.

## **GlassFish**

Es un servidor de aplicaciones desarrollado por *Sun Microsystems* que implementa las tecnologías definidas en la plataforma *Java EE* y permite ejecutar aplicaciones que siguen esta especificación. La versión comercial es denominada *Sun GlassFish Enterprise Server*. Es gratuito y de código libre, se distribuye bajo un licenciamiento dual a través de la licencia CDDL y la GNU GPL.



#### **5.4.1.1. Base de Datos**

Se requiere una base de datos para almacenar la información del contenido que se va a presentar, consultar o alterar mediante las peticiones SMS, es necesario tomar en consideración que la elección del servidor HTTP, el lenguaje de programación y la base de datos van de la mano, esto por temas de configuración, ya que en el caso de la base de datos se debe poseer un controlador específico para que el servidor HTTP se comunique de forma correcta y eficiente con la base de datos.

En entornos con bases de datos existentes, suponiendo de *Microsoft*, se recomienda crear bases de datos de paso o de transferencia, estas bases de datos estarán separadas de la base de datos de producción y mediante procedimientos internos se deberán de actualizar diariamente.

#### **5.5. Ciclo del HTTP Server**

La ejecución de una petición al servidor debe contener los siguientes pasos:

##### **Recepción de parámetros**

Los parámetros de ejecución, es decir, los que servirán más adelante para consultar la información:

Figura 48. **Código fuente de recepción de parámetros**

```
39  /*RECEPCION DE PARAMETROS*/
40
41  String pin = request.getParameter( "pin" );
42  String carne = request.getParameter( "nc" );
```

Fuente: elaboración propia

### Conectar a la base de datos

Antes de consultar es necesario conectarse a la base de datos:

Figura 49. **Código fuente del proceso de conexión**

```
52
53  /*CONECTAR A LA BASE DE DATOS*/
54  Connection conn;
55  conn = dbman.dbConnect( "localhost", "3306", "buro_gt", "root", "admin" );
56
```

Fuente: elaboración propia

Posteriormente crear una consulta SQL que devuelva la información requerida:

Figura 50. **Código fuente de la consulta SQL**

```
58  String SQL;
59  SQL = "SELECT rubro, precio " +
60        "FROM gastos" +
61        "WHERE carnet = '"+carne+"' " +
62        "AND pin like '"+pin+"%' " +
63        " ";
64  out.print(SQL+"<BR>");
```

Fuente: elaboración propia

Por cada registro leído acumular el total de la deuda:

Figura 51. **Código fuente de la lectura de registros de SQL**

```
69  if (consulta.next()==true){
70
71      /* RECUPERAR LOS DATOS DE LA CONSULTA*/
72      Long total;
73      String rubro      = consulta.getString("rubro"  );
74      Long Valor       = consulta.getLong("valor");
75      total = total + Valor;
76
77  }else{
78      System.out.println("Total a pagar:" + total)
79  }
```

Fuente: elaboración propia

## Responder

Al finalizar la lectura de los registros se ha sumado el total a pagar por el estudiante:

Figura 52. **Código fuente del despliegue de la respuesta vía salida Estándar**

```
69  if (consulta.next()==true){
70
71      /* RECUPERAR LOS DATOS DE LA CONSULTA*/
72      Long total;
73      String rubro      = consulta.getString("rubro"  );
74      Long Valor       = consulta.getLong("valor");
75      total = total + Valor;
76
77  }else{
78      System.out.println("Total a pagar:" + total)
79  }
```

Fuente: elaboración propia

De este modo se concluye la realización de las etapas de desarrollo en el cual se ha cubierto la totalidad de las distintas tecnologías que hacen posible la realización de Sistemas de Servicio Móvil basados en mensajería de texto SMS.



## CONCLUSIONES

1. Los elementos de infraestructura de *software* para montar un centro de servicios consumibles por teléfonos móviles son: un servidor con sistema operativo *Linux*, una conexión con SMSC virtual, la cual requiere un módem GSM y una tarjeta SIM.
2. Los elementos de *software* involucrados en la arquitectura para montar un centro de servicios consumibles por teléfonos móviles son: *Gateway SMS*, para interpretar la red GSM hacia la red IP, Servidor HTTP de páginas dinámicas, los cuales alojan los servicios programados en un lenguaje que pueda interpretar el servidor HTTP y una base de datos.
3. La arquitectura de la red GSM está comprendida por las torres de red inalámbrica, los proveedores de telefonía con sus SMSC y los usuarios de telefonía móvil; para ampliar las capacidades de la red se cuenta con los métodos de transporte de datos SMS y conexiones HTTP por medio de GPRS o WLAN respectivamente.

4. Los sistemas operativos móviles más destacados son *Symbian*, *Windows Mobile*, *Android* *iPhone* y *BlackBerry*; los cuales permiten ser programados en lenguajes de tercera generación como: *Java*, *C* y *C++* y *Python*. Todos los teléfonos celulares soportan la plataforma *J2ME*, la cual permite ejecutar cualquier tipo de programa escrito en *Java* que haga uso de las funciones nativas del teléfono celular.
  
5. Un servicio consumible por el teléfono móvil, es aquel que mediante el *Software Gateway SMS*, es ejecutado por el envío de un mensaje de texto, ya sea esté originado desde una aplicación celular o desde el buzón de salida de mensajes de texto del teléfono celular. El servicio debe ser capaz de interpretar la información enviada por el usuario y consultar, o ejecutar, una acción que devuelva una respuesta al solicitante por medio de *SMS*.
  
6. Una aplicación móvil se ejecuta dentro de un teléfono celular como un programa y ejecuta el envío de *SMS*, solicitando al usuario la información de consulta o los parámetros de ejecución del servicio, además debe implementar una interfaz para el envío de mensajes ya sea de forma nativa o bien por medio de la plataforma *J2ME*.

## RECOMENDACIONES

1. Se debe considerar un análisis de mercado para estimar el número de mensajes de texto que la aplicación va a generar, con base en este número, se debe determinar si la aplicación puede ser soportada por un SMSC virtual, o bien, si se debe contratar el servicio de un operador de telefonía móvil.
2. La tecnología de *software* puede dividirse en dos (2) ramas principales, el *Open Source* y el mercado privado. Se debe considerar que el *software Open Source* provenga de una organización que actualice periódicamente su *software*, que tenga participación en el mercado y contar siempre con las fuentes más actualizados para asegurar la calidad del servicio.
3. Se recomienda hacer uso de investigación técnica sobre el *hardware* utilizado por los proveedores de servicio móvil, ya que en diversas ocasiones, los empresarios bautizan las tecnologías con nombres modernos como el caso de GSM; la cual en realidad aun era una tecnología previa (TDMA).

4. Las diferentes prestaciones de las terminales móviles hacen necesario el uso de plataformas estándar de desarrollo, plataformas como J2ME permiten el acceso a las APIS más comunes de los teléfonos celulares. Se recomienda el uso de SDK específicos de las plataformas, siempre y cuando, los desarrollos abarquen un mercado de consumidores con terminales en común, en caso de no conocer las terminales objetivo, se recomienda usar SDK genéricos y seleccionar las APIS por el método de menor común denominador; el cual permite la mayor compatibilidad con las funciones nativas de los dispositivos móviles independientemente del sistema operativo.
  
5. Los *Gateway* que permiten el uso de HTTP como interfaz de comunicación entre el servidor IP y la red GSM, son por lo general más eficientes y fáciles de mantener que aquellos que sólo ejecutan acciones de terminal; es decir, programas ejecutables. Se recomienda utilizar interfaces web y servidores HTTP como *Apache* o *Tomcat* y desplegar los servicios como páginas *web* dinámicas, ya que pueden incluso ser reutilizados por otros medios de acceso.
  
6. Las aplicaciones móviles deben solicitar la menor cantidad de información al usuario, es más eficiente utilizar entradas predeterminadas para las categorías, preseleccionar las selecciones de acuerdo a la mayor probabilidad de uso, restringir las entradas numéricas y las textuales, en lugar de mezclar ambos. Toda aplicación debe ser probada en una gama amplia de terminales en búsqueda de *bugs* antes de salir a producción.



## BIBLIOGRAFÍA

1. Asociación Hispanoamericana de Centros de Investigación y Empresas de Telecomunicaciones. *Las telecomunicaciones y la movilidad en la sociedad de la información*. España: Albadalejo, 2005. 430 p.
2. AYUN, Roy Ben. *Java ME on Symbian OS: Inside the Smartphone Model*. England: Wiley, 2009. 409 p.
3. Nokia Corporation. *Series 40 Platform. Introductory White Paper*. Versión 3.1. s.l.: Nokia Corporation, 2008. 33 p.
4. ROJAS GÁLVEZ, José Miguel; DÍAZ ORTEGA, Lucas. *Java a Tope: J2me (Java 2 Micro Edition)*. Edición electrónica. Málaga: Universidad de Málaga, 2003. 148 p.
5. VIRKUS, Robert. *Pro J2ME Polish: Open Source Wireless Java Tools Suite*. New York: Apress, 2005. 450 p.



## APÉNDICE

### A.1.1. Plan de contingencia

#### Fallas del *Hardware*

#### Mantener partes adicionales de *hardware*<sup>1</sup>

En su forma más simple, las exposiciones debidas a fallas del *hardware* se pueden reducir manteniendo repuestos de *hardware* adicionales. Por supuesto, este enfoque asume dos cosas:

- Alguien está en el sitio con suficientes habilidades para diagnosticar el problema, identificar la parte defectuosa y reemplazarla
- Está disponible un repuesto para el *hardware* defectuoso
- Estos problemas se cubren con más detalles en las secciones siguientes

---

<sup>1</sup> [web.mit.edu/rhel-doc/4/RH-DOCS/rhel.../ch-disaster.html](http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel.../ch-disaster.html)

## **Contratos de servicios<sup>2</sup>**

Los contratos de servicios pasan el problema de las fallas de *hardware* a alguien más. Lo único que necesita hacer es confirmar que ha ocurrido una falla y que no parece estar relacionado a un problema de *software*. Usted simplemente hace la llamada y alguien más aparece para encargarse de que las cosas estén en funcionamiento otra vez.

Parece muy simple, pero como con la mayoría de las cosas en la vida, hay mucho más de lo que el ojo puede ver. He aquí algunas cosas que debería considerar cuando esté revisando contratos de servicios:

Horas de cobertura, tiempo de respuesta, partes disponibles, presupuesto disponible y *hardware* cubierto.

## **Fallas De Software**

### **Caídas del sistema**

Las caídas ocurren cuando el sistema operativo experimenta una condición de error desde la cual no se puede recuperar. La razón de las caídas puede variar. Cuando un sistema operativo falla, se debe reiniciar el sistema para poder continuar la producción.

---

<sup>2</sup> [web.mit.edu/rhel-doc/4/RH-DOCS/rhel.../ch-disaster.html](http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel.../ch-disaster.html)

## **Bloquearse**

Cuando el sistema operativo deja de manejar estos eventos, el sistema se detiene aparatosamente. Esto se conoce como un bloqueo. Los interbloqueos cuando dos recursos se disputan los recursos que el otro posee pueden provocar que el sistema se bloquee, con el mismo resultado final — una falta total de productividad, se debe terminar el proceso que este causando el bloqueo o en el peor de los casos se debe reiniciar el sistema para poder continuar la producción.

## **Obteniendo ayuda — Asistencia de *software***

De la misma forma que los fabricantes de *hardware* ofrecen soporte para sus productos, muchos proveedores de *software* colocan paquetes de soporte disponibles a sus clientes. Excepto por las diferencias obvias, los contratos de soporte de *software* pueden ser bastante similares a los contratos de *hardware*.

El nivel de soporte suministrado por un fabricante de *software* puede variar. He aquí algunas de las estrategias de soporte más comunes empleadas hoy día.

- Documentación
- Auto-asistencia
- Soporte de *Web* o de correo electrónico
- Soporte telefónico
- Soporte *in situ*

## **Documentación**

Aunque a veces es ignorada, la documentación del *software* puede servir como una herramienta de soporte de primer nivel. Bien sea en línea o impresa, la documentación a menudo contiene la información necesaria para resolver muchos problemas.

## **Auto asistencia**

La auto-asistencia confía en que el cliente utiliza los recursos en línea para resolver sus propios problemas relacionados al *software*. A menudo estos recursos toman la forma de *FAQ's* (Preguntas más frecuentes) basadas en el *Web* o bases de datos de conocimientos.

Las *FAQ's* tienen poca o ninguna capacidad de selección, dejando que el cliente se desplace pregunta por pregunta con la esperanza de encontrar una que mencione el problema que tiene. Las bases de conocimiento tienden a ser un poco más sofisticadas, permitiendo la entrada de términos para realizar búsquedas. Las bases de conocimientos también pueden ser bien completas en ámbito, convirtiéndola en una buena herramienta para resolver problemas.

## **Soporte *Web* o de correo electrónico**

Muchas veces lo que a veces parece un sitio de auto asistencia, también incluye algunas formas basadas en *web* o correo electrónico que permiten que la persona pueda enviar preguntas al personal de soporte.

Mientras que esto puede parecer a primera vista una mejora de un buen sitio *web* de auto asistencia, realmente depende de la gente que contesta los correos.

Si el personal de soporte está saturado de trabajo, es difícil obtener la información necesaria de ellos, pues su principal preocupación es la de responder rápidamente a cada correo y moverse al siguiente. La razón de esto es que casi todo el personal de soporte usualmente es evaluado por el número de problemas que pueden resolver. Los problemas de escalada también son complicados porque hay poco que hacer dentro de un correo electrónico para promover respuestas con mejores tiempos de respuestas y más útiles.

La forma de obtener el mejor servicio es asegurarse de que su correo electrónico responde todas las preguntas que un técnico podría preguntarle, tales como:

- Claramente describa la naturaleza del problema
- Incluya todos los números de versión pertinentes
- Describa lo que ya ha hecho en un intento de resolver el problema (aplicó los últimos parches, reinició con la configuración mínima, etc.)
- Al darle al técnico de soporte más información, tiene más oportunidades de obtener la asistencia que necesita

### **Soporte telefónico**

El soporte telefónico implica hablar con un técnico a través del teléfono. Este estilo de soporte es más similar al soporte de *hardware*; en que pueden haber diferentes niveles de soporte disponible (con diferentes horas de cobertura, tiempo de respuesta, etc.)

## **Soporte *in situ***

También conocido como consultorías *in situ*, el soporte de *software in situ* normalmente está reservado para resolver problemas específicos o para efectuar cambios críticos, tales como la instalación y configuración inicial, actualizaciones importantes, entre otros. Como se puede esperar, este es el tipo de soporte más costoso.

Sin embargo, hay situaciones en las que las consultorías *in situ* tienen sentido. Como ejemplo considere una pequeña organización con un único administrador de sistemas. La organización va a implementar su primer servidor de bases de datos, pero la implementación (y la organización) no es lo suficientemente grande como para justificar la contratación de un administrador de bases de datos dedicado. En esta situación, a menudo puede ser más económico traer a un especialista de un proveedor de bases de datos para que maneje la implementación inicial (y ocasionalmente más adelante, si surge la necesidad), que entrenar al administrador de sistemas con una habilidad que será utilizada muy de vez en cuando.

### **A.1.2. Fenómenos naturales y ambientales**

#### **Fallos de electricidad**

La energía suministrada desde la compañía eléctrica normalmente no satisface los estándares necesarios para un centro de datos. Por lo tanto, usualmente se requiere un nivel de condicionamiento de la energía. Hay varios enfoques para hacer esto posible:



## **Protectores de corriente**

Los protectores de corriente hacen exactamente lo que su nombre indica, filtran el oleaje de la fuente de poder. La mayoría no hacen nada más que esto, dejando los equipos vulnerables a otros problemas relacionados con la energía.

## **Acondicionadores de energía**

Los acondicionadores de energía tratan de lograr un enfoque más completo; dependiendo de lo sofisticado que sea la unidad, los acondicionadores de energía a menudo cubren la mayoría de los problemas descritos arriba.

## **Conjuntos de moto-generadores**

Un moto-generador esencialmente es un motor eléctrico grande activado por su suministro normal de poder. El motor está conectado a una rueda voladora, la cual a su vez está conectada a un generador.

El motor mueve la rueda y el generador, lo cual produce la electricidad en suficientes cantidades para correr el centro de datos. De esta forma, la energía del centro de datos está separada de la electricidad externa, lo que significa que se eliminan una gran parte de los problemas relacionados con la electricidad. La rueda voladora también permite la habilidad de mantener energía durante interrupciones eléctricas cortas, pues toma varios segundos para que la rueda se detenga al punto en que ya no genere energía.

## **Fuentes de alimentación continuas**

Algunos tipos de fuentes de alimentación continuas, más conocidos como UPS, incluyen la mayoría (si no es que todas) las funcionalidades de un acondicionador de energía.

## **Independencia de fuentes de energía**

Cada edificio de la Universidad de San Carlos de Guatemala cuenta con su propio suministro de electricidad, para cuando un edificio en específico no paga por el servicio de luz únicamente ese edificio se ve afectado por el corte de energía, o si el servicio es interrumpido por diferentes razones únicamente es afectado ese edificio.

## **Calefacción, ventilación y aire acondicionado**

Las fallas en el sistema HVAC pueden ser devastadoras para el funcionamiento continuo de su centro de datos. Dada su complejidad y la naturaleza electromagnética, las posibilidades de una falla son muchas y variadas.

He aquí algunos ejemplos:

- Las unidades de manejo de aire acondicionado (esencialmente grandes ventiladores eléctricos) pueden fallar debido a la sobrecarga eléctrica, una falla, falla de la correa/polea, etc.

- Las unidades de enfriamiento pueden perder su refrigerante debido a filtraciones o tomar sus compresores y/o motores.
- La reparación y mantenimiento de un HVAC es un campo muy especializado - un campo que el administrador de sistemas promedio debería dejar a los expertos. En cualquier caso hacer esta reparación.

## **Humedad**

La humedad puede afectar los equipos estropeándolos y afectar la salud de las personas que residen en el edificio.

Es necesario mejorar la ventilación para evitar los problemas por acumulación de humedad. En caso de no poder mejorar la ventilación puede escogerse la vía de la deshumificación artificial, con la compra de deshumificadores eléctricos que extraen la humedad del ambiente condensándola en un recipiente específico. Nótese que los deshumificadores deben usarse cuando no haya personal o algún ser viviente en la misma área durante tiempos prolongados por que pone en riesgo la salud, debido a la extracción de toda el agua del ambiente.

### **A.1.3. Fenómenos sociales y humanos<sup>3</sup>**

Puesta en marcha de una política de seguridad.

Actualmente las legislaciones de algunos países, obligan a las empresas, instituciones públicas a implantar una política de seguridad.

---

<sup>3</sup> [es.wikipedia.org/wiki/Seguridad\\_informática](http://es.wikipedia.org/wiki/Seguridad_informática)

Generalmente se ocupa exclusivamente a asegurar los derechos de acceso a los datos y recursos con las herramientas de control y mecanismos de identificación. Estos mecanismos permiten saber que los operadores tienen sólo los permisos que se les otorgó.

La seguridad informática debe ser estudiada para que no impida el trabajo de los operadores en lo que les es necesario y que puedan utilizar el sistema informático con toda confianza.

Por eso en lo referente a elaborar una política de seguridad, conviene:

- Elaborar reglas y procedimientos para cada servicio de la organización
- Definir las acciones a emprender y elegir las personas a contactar en caso de detectar una posible intrusión
- Sensibilizar a los operadores con los problemas ligados con la seguridad de los sistemas informáticos

Los derechos de acceso de los operadores deben ser definidos por los responsables jerárquicos y no por los administradores informáticos, los cuales tienen que conseguir que los recursos y derechos de acceso sean coherentes con la política de seguridad definida. Además, como el administrador suele ser el único en conocer perfectamente el sistema, tiene que delegar a la directiva cualquier problema e información relevante sobre la seguridad, y eventualmente aconsejar estrategias a poner en marcha, así como ser el punto de entrada de la comunicación a los trabajadores sobre problemas y recomendaciones en término de seguridad.

## **Técnicas de aseguramiento del sistema<sup>4</sup>**

### **Codificar la información**

Criptología, Criptografía y Criptociencia, contraseñas difíciles de averiguar a partir de datos personales del individuo.

### **Tecnologías repelentes o protectoras**

Cortafuegos, sistema de detección de intrusos - *antispyware*, *antivirus*, llaves para protección de *software*, etc. Mantener los sistemas de información con las actualizaciones que más impacten en la seguridad.

### **Consideraciones de *software*<sup>5</sup>**

Tener instalado en la máquina únicamente el *software* necesario reduce riesgos.

Así mismo, tener controlado el *software* asegura la calidad de la procedencia del mismo (el *software* obtenido de forma ilegal o sin garantías aumenta los riesgos). En todo caso un inventario de *software* proporciona un método correcto de asegurar la reinstalación en caso de desastre. El *software* con métodos de instalación rápidos facilita también la reinstalación en caso de contingencia.

---

<sup>4</sup> [www.revistaitnow.com/pdfs/.../seguridad-informatica-symantec.pdf](http://www.revistaitnow.com/pdfs/.../seguridad-informatica-symantec.pdf)

<sup>5</sup> [es.wikipedia.org/wiki/Seguridad\\_informática](http://es.wikipedia.org/wiki/Seguridad_informática)

Existe *software* que es conocido por la cantidad de agujeros de seguridad que introduce. Se pueden buscar alternativas que proporcionen iguales funcionalidades pero permitiendo una seguridad extra.

### **Consideraciones de una red<sup>6</sup>**

Los puntos de entrada en la red son generalmente el correo, las páginas *web* y la entrada de ficheros desde discos, o de ordenadores ajenos, como portátiles.

Mantener al máximo el número de recursos de red sólo en modo lectura, impide que ordenadores infectados propaguen virus. En el mismo sentido se pueden reducir los permisos de los usuarios al mínimo.

#### **A.1.4. CONCLUSIONES**

- La reducción del riesgo de perder vidas, propiedades, bienes comunes y ambientales, causados por las causas descritas se lograran en la manera que todas las personas relacionada a la institución conozcan y apliquen las normas de seguridad.
- Los pasos y recomendaciones descritos en el presente documento son maneras de mitigar y reducir el impacto causado por los desastres, estos nos dicen cómo actuar después de ellos.

---

<sup>6</sup> [www.mitecnologico.com/.../SeguridadInformatica](http://www.mitecnologico.com/.../SeguridadInformatica)

- No se cuenta actualmente con una estructura organizativa, pero se plantea que cada persona debe conocer las normas de seguridad y posibles riesgos para actuar de forma individual y ordenada.
- Los riesgos actuales son mínimos, pero debe crearse conciencia de su existencia.
- Debido a que en las mencionadas escuelas no se cuenta con manuales de evacuación propios se tomara el normativo del edificio, en caso éste no existiera, el presente documento desarrolla los normativos individuales que se deben practicar para la evacuación de algún establecimientos en los casos pertinentes.