



**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE CUENCA**

**CARRERA DE INGENIERÍA DE SISTEMAS**

**Tesis previa a la obtención del Título de:**  
**Ingeniero de Sistemas.**

**Tema:**

**“Análisis de uso de dispositivos celulares para apoyo a la gestión  
empresarial, creación de un prototipo”**

**Autores:**

Jorge Cabrera

Pablo Villa

**Director de Tesis:**

Ing. Diego Quinde F.

Cuenca, julio del 2012



## **DECLARACION DE RESPONSABILIDAD**

Los conceptos desarrollados, análisis realizados y las conclusiones del presente trabajo, son de exclusiva responsabilidad de los autores.

CUENCA, 23 de Julio del 2012.

(f)\_\_\_\_\_

Pablo Daniel Villa

(f)\_\_\_\_\_

Jorge Vinicio Cabrera

Yo, Diego Marcelo Quinde Falconí, Docente de Carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana, CERTIFICO que he leído íntegramente el presente trabajo de grado, comprobando que se cumplen los objetivos planteados en el proyecto. Así mismo declaro que los autores Pablo Villa y Jorge Cabrera son los únicos responsables de la originalidad del trabajo desarrollado.

(f)\_\_\_\_\_

Ing. Diego Quinde Falconí



***Dedicatoria.***

*Durante la vida estudiantil he tenido varias caídas que lo he superado gracias al esfuerzo, dedicación y fe, además cada uno de nosotros somos indispensables gracias a los conocimientos, valores, etc., adquiridos en la Universidad sin tomar presente los prejuicios de algunos docentes que al final fue una fortaleza y reto para demostrar que somos capaces. Estas líneas son escritas con el fin de dedicar a las personas que hicieron posible el cumplimiento de mis objetivos, gracias a mi Mami Rosa ya que fue la persona que estuvo presente en mis penas y alegrías, también a mi papá Carlos que de una u otra manera su apoyo y consejos fueron mi fortaleza. También dedico a mis sobrinos Gaby, Juan y Carlita ya que de ellos he aprendido ver la vida de otra manera, lleno de felicidad, alegrías y compartir buenos momentos.*

*Esto es un paso más para mi vida profesional ya que el recorrido recién comienza.*

*Gracias por su apoyo incondicional.*

*Jorge Cabrera.*



***Dedicatoria***

*Dedico este proyecto a mis padres ya que fue la fuente y fortaleza para seguir adelante y no rendirme, además agradezco por apoyarme y estar en los buenos y malos momentos.*

*Pablo Villa*



***Agradecimiento.***

*Ante todo quiero agradecer a Dios por dar salud a mis padres, hermanos, sobrinos, mi tía María, en general a toda mi familia. Gracias Diosito por creer en mí, porque Tú sabes lo que he pasado y por brindarme tu confianza, además darme oportunidades para demostrar mis capacidades.*

*Agradezco a mis padres ya que con su sacrificio pudieron ayudarme a cumplir mis metas, gracias por sus palabras y brindarme su confianza, aunque he abusado de ellos espero que con esto pueda darles una alegría.*

*También quiero agradecer a mis hermanos Mary, Marco, Adrián, Wilson, por su comprensión, por su confianza, por su amistad, y estar siempre unidos en los buenos y malos momentos.*

*A mi Tía María que en realidad ha sido una segunda madre, gracias por sus consejos, sus regaños que me ha ayudado en mi vida, a mi familia en general que ha estado pendiente en mi éxito.*

*En realidad son contadas las personas que tuvieron pendiente de esta Tesis, ya que siempre me dijeron que siga adelante, que a pesar del tiempo fue una amistad incondicional, y en especial agradezco a Pao por ayudarnos, por aguantarnos y ser parte de “el grupito”.*

*Y para finalizar agradezco a la persona más importante ya que desde el principio hasta el final del proyecto nos brindó su colaboración, comprensión y su confianza. Gracias Ingeniero Diego Quinde.*

*Jorge Cabrera.*



***Agradecimiento.***

*Sin esperar nada a cambio pude culminar con mis estudios agradeciendo primero a Dios ya que me dio salud para seguir adelante, agradezco a mis padres por su apoyo incondicional, a mi hermana, a mi esposa, y a mi familia ya que de una u otra manera me apoyaron para cumplir con mi meta.*

*Y para concluir agradezco sinceramente a la persona que nos ayudó desde el principio hasta el final del proyecto nos brindó su colaboración, comprensión y su confianza. Gracias Ingeniero Diego Quinde*

*Pablo Villa.*



## Índice

1. CAPITULO I.....	11
1.1 INTRODUCCION .....	11
1.1.1. Visión General del proyecto .....	11
1.2 OBJETIVO Y ALCANCE DEL PROYECTO.....	12
1.2.1. Objetivos.....	12
1.2.1.1. Objetivo General .....	12
1.2.1.2. Objetivos Específicos .....	12
1.2.2. Características del proyecto .....	12
1.2.3. Alcance del proyecto .....	13
1.2.4. Restricciones.....	14
1.3 REQUERIMIENTOS DEL PROYECTO/ENTREGABLES .....	14
1.4 ARQUITECTURA DE LA APLICACIÓN PARA CELULARES .....	15
2. CAPITULO II.....	16
2.1 SISTEMA OPERATIVO MOVIL.....	16
2.1.1. Definición .....	16
2.1.2. Capas .....	16
2.1.2.1. Kernel .....	16
2.1.2.2. Middleware .....	17
2.1.2.3. Entorno de ejecución de aplicaciones.....	17
2.1.2.4. Interfaz de usuario .....	17
2.2 IDENTIFICACIÓN DE LOS SISTEMAS OPERATIVOS MÁS COMUNES .....	18
2.2.1. ANDROID.-.....	19
2.2.1.1. Estructura .....	20
2.2.1.2. Características .....	20
2.2.1.3. Arquitectura .....	22
2.2.1.4. Kit de desarrollo .....	24
2.2.1.5. Versiones .....	25
2.2.2. IOS .....	28
2.2.2.1. Características .....	29
2.2.2.2. Arquitectura .....	31
2.2.2.3. Kit de desarrollo .....	32



2.2.2.4.	Versiones .....	32
2.2.3.	SYMBIAN.....	35
2.2.3.1.	Características .....	36
2.2.3.2.	Arquitectura .....	37
2.2.3.3.	Kit de desarrollo .....	40
2.2.3.4.	Versiones .....	41
2.2.4.	BLACKBERRY OS .....	43
2.2.4.1.	Características .....	44
2.2.4.2.	Arquitectura .....	46
2.2.4.3.	Kit de desarrollo .....	46
2.2.4.4.	Plataforma.....	46
2.2.4.5.	Versiones .....	47
2.2.5.	WINDOWS PHONE.- MICROSOFT .....	49
2.2.5.1.	Características .....	49
2.2.5.2.	Arquitectura .....	51
2.2.5.3.	Kit de desarrollo .....	53
2.2.5.4.	Versiones.- .....	53
2.3	MERCADO DE LOS SISTEMAS OPERATIVOS.....	55
3.	CAPITULO III .....	57
3.1	IDENTIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN MÁS COMUNES.....	57
3.1.1	JAVA 2 MICRO EDITION J2ME.....	57
3.1.1.1.	Ventajas .....	60
3.1.1.2.	Desventajas .....	61
3.1.1.3.	Características .....	61
3.1.1.4.	Arquitectura del entorno de ejecución de J2ME .....	62
3.1.1.4.1.	KVM.....	63
3.1.1.4.2.	Configuraciones .....	64
3.1.1.4.3.	Perfiles .....	66
3.1.1.5.	Plataformas .....	68
3.2.1	MICROSOFT COMPACT .NET FRAMEWORK.....	69
3.2.1.1.	Ventajas .NET Compact Framework.....	72
3.2.1.2.	Características .NET Compact Framework .....	73



3.2.1.3.	Descripción del lenguaje C# .....	74
3.2.1.4.	Plataformas .....	76
3.3.1	PYTHON (PYS60).....	77
3.3.1.1.	Ventajas .....	78
3.3.1.2.	Características .....	79
3.3.1.3.	Plataformas .....	79
3.4.1	OBJECTIVE-C .....	80
3.4.1.1.	Ventajas .....	80
3.4.1.2.	Características .....	80
3.4.1.3.	Arquitectura. ....	81
3.4.1.4.	Plataformas .....	82
3.5.1	WML Y XHTML.....	83
3.5.1.1.	Características .....	85
3.5.1.2.	Proceso para desarrollar una aplicación para WAP.....	86
4.	CAPITULO IV .....	88
4.1	PROTOTIPO .....	88
4.1.1.	REQUERIMIENTOS DE SOFTWARE Y HARDWARE.....	88
4.1.1.1.	Selección lenguaje de programación .....	88
4.1.1.2.	Selección de gateway sms.....	89
4.1.1.3.	Selección de Base de datos .....	92
4.1.1.4.	Requerimientos de Hardware.....	92
4.1.2.	INTERFAZ DE USUARIO .....	93
4.1.2.1.	Arquitectura de la pasarela SMS. ....	93
4.1.2.2.	Interfaces, Configuraciones y conexiones. ....	94
4.1.2.2.1.	Servidor Web, Modelo de la Base de Datos y Aplicación Web.....	94
4.1.2.2.2.	Configuración del Gateway kannel .....	107
4.1.2.2.3.	Aplicación móvil .....	112
4.1.3.	PLAN DE PRUEBAS .....	125
4.1.4.	EJECUCIÓN DE PLAN DE PRUEBAS .....	132
	CONCLUSIONES.....	152
	RECOMENDACIONES.....	154
	BIBLIOGRAFIA.....	155



## Índice de Figuras

Figura 1.1: Arquitectura de la Aplicación Móvil .....	15
Figura 2.1: Arquitectura Android.....	22
Figura 2.2: Arquitectura iOS.....	31
Figura 2.3: Arquitectura Symbian .....	37
Figura 2.4: Arquitectura de Windows Phone.....	51
Figura 2.5: Mercado Latinoamérica. ....	55
Figura 2.6: Mercado Ecuador.....	56
Figura 3.1: Arquitectura plataforma Java .....	59
Figura 3.2: Ubicación de las tecnologías java.....	59
Figura 3.3: Arquitectura de J2ME.....	62
Figura 3.4: Entorno de ejecución .....	63
Figura 3.5: Librerías de CLDC .....	65
Figura 3.6: Librerías de MIDP .....	67
Figura 3.7: Arquitectura Objective-C.....	81
Figura 4.1: Arquitectura del Gateway. ....	91
Figura 4.2: Posición lógica de la pasarela SMS entre un teléfono y un servidor de contenido.....	93
Figura 4.3: Case Method.....	96



## 1. CAPITULO I

### 1.1 INTRODUCCION

#### 1.1.1. Visión General del proyecto

El teléfono móvil o celular es un dispositivo de gran importancia en el ámbito de negocios, que permite realizar marketing móvil (m-marketing) y potenciar los servicios de las empresas. Al utilizar el envío de SMS o mensajes cortos de texto, las empresas pueden realizar operaciones de promoción y porque no ver la fidelidad del cliente

Este sistema de comunicación ha evolucionado, tanto en su tecnología como su costo, hoy en día el celular es una cuestión de costumbre, necesidad o simplemente de moda.

En la actualidad existe una gran variedad de celulares para todo tipo de usuario, sin duda este tipo de tecnología es una de los principales medios de comunicación y de un consumo masivo.

Además de realizar el análisis de los Sistemas Operativos Móviles, la tecnología que manejan y todas sus características, el proyecto planteado pretende brindar una herramienta que se pueda usar conjuntamente con dispositivos móviles y que se vincule a la gestión empresarial de manera que ciertos procesos y tareas se realicen de manera ágil, fácil y rápidamente.

La aplicación en si proyecta usar un dispositivo móvil mediante el cual se ingrese información y esta sea enviada de tal manera que el mensaje se almacene en un servidor externo.

También se va a incorporar software amigable para que los consumidores puedan controlarlos de una manera fácil y eficiente.



## 1.2 OBJETIVO Y ALCANCE DEL PROYECTO

### 1.2.1. Objetivos

#### 1.2.1.1. Objetivo General

Análisis de uso de dispositivos móviles con el propósito de proponer alternativas de creación de Software de Gestión como apoyo al quehacer empresarial.

#### 1.2.1.2. Objetivos Específicos

- Analizar las alternativas en Sistemas Operativos para dispositivos celulares.
- Analizar las alternativas en lenguajes de programación para celulares.
- Crear un prototipo de software para ser instalado en dispositivos celulares.
- Realizar pruebas reales de la aplicación.
- Permitir dar un valor agregado a las operaciones empresariales que se encuentran en el mercado.
- Automatizar los servicios orientados al cliente, para que los mismos sean accesibles a través de teléfonos celulares.

### 1.2.2. Características del proyecto

- Investigativo
  - ✓ Definir cuáles son los Sistemas Operativos Móviles que existen en la actualidad.
  - ✓ Definir el mercado que abarca cada uno de estos Sistemas Operativos Móviles.
  - ✓ Definir los lenguajes de programación que se emplean para el desarrollo de aplicaciones.



- Medible en el tiempo.
- Practico:

Se realizara un prototipo de la aplicación para dispositivos móviles y de igual manera un servidor que envíe y reciba SMS.

### **1.2.3. Alcance del proyecto**

El desarrollo de la Sociedad de la Información y el mejoramiento de las comunicaciones han permitido que se desarrollen aplicaciones que han aportado en la forma de realizar negocios y transacciones, permitiendo que las empresas sean más productivas y mejoren sustancialmente la relación con los clientes.

El alcance de este proyecto se define de la siguiente manera, comenzando por el capítulo actual donde se presenta una reseña general del trabajo que se va a realizar, a donde queremos llegar con el mismo y una breve idea de la motivación existente.

En el segundo capítulo se desarrolla con mayor profundidad el estudio sobre los Sistemas Operativos Móviles existentes en la actualidad, que conceptualmente proveen las bases y el punto de partida para el desarrollo del presente proyecto.

En el tercer capítulo se abarcará un estudio de los diferentes lenguajes de programación utilizados para el desarrollo de aplicaciones para dispositivos móviles (celulares), además de definir el más adecuado para el desarrollo de la aplicación prototipo que se llevara a cabo en el cuarto capítulo.

En el cuarto capítulo se realizará una aplicación prototipo, la cual consiste en conectarse a una base de datos para el envío de información desde un dispositivo celular vía SMS y que la misma se registre en un servidor y recibir un mensaje de confirmación, definiendo las herramientas que se van a utilizar como son la base de datos, el lenguaje de programación, el tipo de conexión, el hardware y sus respectivas pruebas y correcciones.



#### **1.2.4. Restricciones**

- La aplicación debe cumplir funcionalidades con limitaciones con respecto al hardware, en cuanto a la capacidad gráfica, memoria, procesadores de baja capacidad, etc.
- Conectividad intermitente y ancho de banda limitado.
- El envío de mensajes no debe ser por internet sino por la tecnología de comunicación SMS.
- Al enviar un mensaje debe ser seguro mediante una codificación.
- El lenguaje de programación sea de libre distribución.

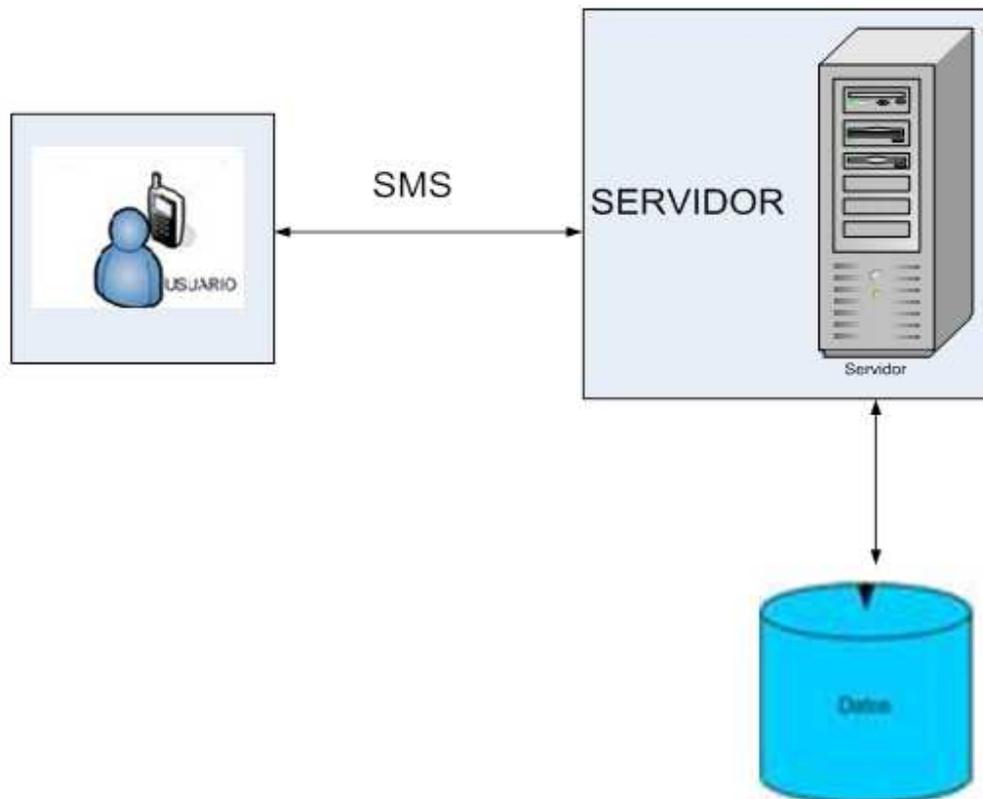
### **1.3 REQUERIMIENTOS DEL PROYECTO/ENTREGABLES**

- La Interfaz de usuario debe proporcionar métodos que son comunes para todos los celulares, la interfaz debe ser básica y fácil de utilizar.
- Guardar la información de los usuarios.
- Interactúe el vendedor con la empresa de modo que se pueda acceder a la información y poder satisfacer las necesidades de los usuarios.
- Se ofrecerá aplicaciones para celulares con la finalidad de aportar ideas innovadoras, para lo cual se desarrollará una aplicación que permita a los celulares con características limitadas brindar apoyo a la gestión empresarial.
- La aplicación del servidor tiene la capacidad de recibir mensajes SMS, procesarlos y remitir mensajes SMS, por lo tanto debe tener la capacidad de interactuar con la base de datos que contiene la información de los usuarios.



## 1.4 ARQUITECTURA DE LA APLICACIÓN PARA CELULARES

Básicamente la aplicación consiste en la interacción del usuario con el Servidor Central. En esta sección se plantean esquemáticamente las funcionalidades de la aplicación.



*Figura 1.1: Arquitectura de la Aplicación Móvil*

*Fuente: El Autor.*

La aplicación servidor es una aplicación con una base de datos, tiene la capacidad de recibir mensajes SMS, procesarlos y remitir mensajes SMS, por lo tanto debe tener la capacidad de interactuar con la base de datos que contiene la información de los usuarios.



## **2. CAPITULO II**

### **2.1 SISTEMA OPERATIVO MOVIL**

En este capítulo conoceremos la importancia de los sistemas operativos para celulares ya que mediante estos pueden funcionar y se puede sacar provechos de los móviles.

#### **2.1.1. Definición**

Un sistema operativo móvil es aquel que controla un dispositivo móvil, partiendo de la capa compleja entre el hardware y el usuario, concebible también como una máquina virtual que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas y determinan las capacidades, conectividad inalámbrica, multimedia de los dispositivos e interactúa con el usuario de una manera fácil.

#### **2.1.2. Capas**

##### **2.1.2.1. Kernel**

La primera capa es la principal para el funcionamiento del móvil, entonces el núcleo o kernel permite el acceso a los distintos elementos del hardware del dispositivo.

El kernel dispone de varios servicios a las capas superiores como son los controladores para el hardware, gestión de procesos, el sistema de los archivos, además del acceso y gestión de la memoria.



### **2.1.2.2. Middleware**

La siguiente capa es el middleware ya que contiene conjunto de módulos permitiendo el uso de las aplicaciones y ejecuciones para móviles.

En este caso el usuario tiene la facilidad de interactuar con el dispositivo ya que es totalmente transparente su funcionamiento, además posibilita la ejecución de varios servicios durante la ejecución de aplicaciones, estos servicios son claves para el normal funcionamiento del móvil, entre los servicios que brinda esta segunda capa es el motor de mensajería y comunicación (Red), códec multimedia, intérpretes de páginas web, gestión del dispositivo y seguridad.

### **2.1.2.3. Entorno de ejecución de aplicaciones**

La tercera capa contiene un gestor de aplicaciones y una serie de interfaces programables APIs o "Application Programming Interface" abiertas, además elementos necesarios para la creación y desarrollo de software para los programadores.

### **2.1.2.4. Interfaz de usuario**

Las interfaces de usuario, la última capa, simplemente nos facilitan la interacción con el usuario, debido a esta capa el usuario se le hace fácil la utilización del dispositivo ya que contiene el diseño de la presentación visual de la aplicación. Los servicios que incluye son componentes gráficos, como pueden ser los botones, pantallas, listas, etc. y por ende el manejo de eventos de cada uno de ellos.

En los dispositivos móviles existen varias aplicaciones nativas que originalmente incluyen los menús, el marcador de números de teléfono y otros.

En la actualidad los móviles poseen aplicaciones avanzadas e incorporadas como puede ser la pantalla táctil, usando simplemente la pantalla sin necesidad de teclado.



También una gran ventaja son los teclados del tipo virtual, que eventualmente remplazarán a los tradicionales dispositivos de entrada, permitiendo de esta manera un mejor y más liviano diseño del móvil.

En fin la personalización de nuestra interfaz es una de las principales utilidades o aplicaciones de los usuarios.

## 2.2 IDENTIFICACIÓN DE LOS SISTEMAS OPERATIVOS MÁS COMUNES



- Android.- Sistema operativo basado en Linux para dispositivos móviles.
- IOS.-Esta en los iPhone, Ipad, Ipod de la marca Apple.
- Symbian.- Propiedad de Nokia, y es el sistema operativo que generalmente se instalan en sus dispositivos.
- Blackberry OS.- Es el sistema operativo de la marca Blackberry.
- Windows Phone.- Microsoft.



### 2.2.1. ANDROID.-



Es un sistema operativo basado en Linux, su creación fue especialmente para dispositivos móviles, como teléfonos inteligentes y tablets, aunque también puede utilizarse en PC.

Es creada por Android Inc, y fue vendida a en el año 2005 a Google. En la actualidad es uno de los más utilizados en los teléfonos inteligentes.

En el año 2007 dieron a conocer de este nuevo sistema operativo junto con la creación de un consorcio de 78 compañías de hardware, software y telecomunicaciones, llamado Open Handset Alliance, con la finalidad de desarrollo de estándares abiertos para dispositivos.

La empresa dueña de Android, Google libero el código bajo la licencia de Apache, una licencia libre y de código abierto.

El sistema permite programar en una variación de java llamada Dalvik, además proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono, como GPS, llamadas, agendas, etc., de una manera muy sencilla en un lenguaje de programación muy conocido como es Java y con un núcleo de sistema operativo libre, gratuito y multiplataforma



### 2.2.1.1. Estructura

Se compone de un conjunto de APIs o aplicaciones, durante la ejecución en un framework Java (aplicaciones orientadas a objetos) sobre un kernel o núcleo de las bibliotecas de Java en una máquina Virtual Dalvik.

Poseen un administrador de interfaz gráfica “Surface manager”, un frameworkOpenCore, una base de datos relacional SQLite, todo esto incluyen en las bibliotecas escritas en lenguaje C.

API gráfica OpenGL es 2.0 3D, un motor de renderizado WebKit, motor gráfico SGL, SSL y una biblioteca estándar de C Bionic.

Las líneas de código del sistema operativo están compuestas por 12 millones, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

### 2.2.1.2. Características

- Android soporta tecnologías de conectividad GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, and WiMAX.
- Soporta SMS y MMS para mensajería y ahora la Android Cloud to Device Messaging Framework (C2DM) es parte del servicio de PushMessaging de Android.
- El soporte a J2ME se puede correr mediante el J2ME MIDP Runner.
- Soporta formatos multimedia como puede ser, WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, OggVorbis, WAV, JPEG, PNG, GIF, BMP.
- Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML



(HTML5 <video>tag).

- Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player.
- Soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, termómetro, aceleración 2d y 3d.
- Soporta pantallas multi-táctiles que inicialmente hicieron su aparición en dispositivos como el HTC Hero.
- El soporte para A2DF y AVRCP fue agregado en la versión 1.5, el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0 y el marcado por voz junto con el envío de contactos entre teléfonos fueron en la versión 2.2.
- Algunos dispositivos pueden actualizar el sistema operativo que lo soporta video llamada ya sea por la red del operador como el Samsung i9000 Galaxy S o sobre IP.
- Tethering también es soportado por Android, el cual permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico, todos los teléfonos con la versión 2.2.
- Para utilizar la conexión 3G del móvil Android en un computador portátil se requiere la instalación de software adicional.



### 2.2.1.3. Arquitectura

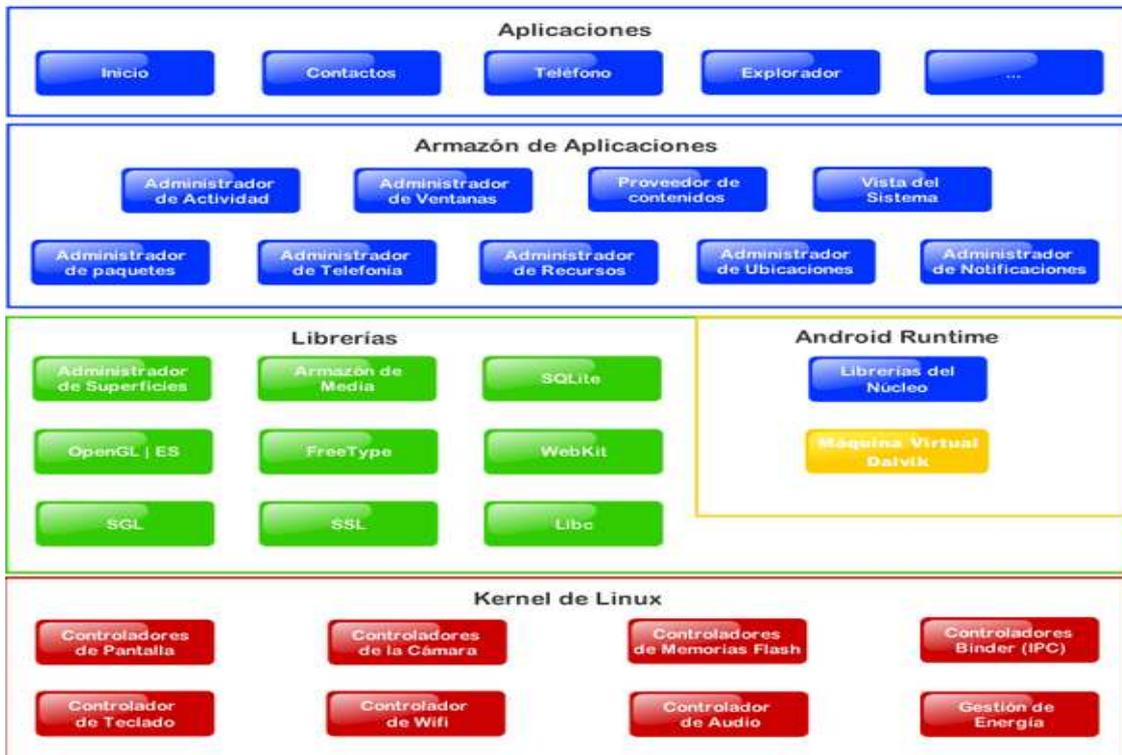


Figura 2.1: Arquitectura Android

Fuente: [http://commons.wikimedia.org/wiki/File:Diagrama\\_android.png](http://commons.wikimedia.org/wiki/File:Diagrama_android.png)

Autor: Álvaro Fuentes Vasquez (Kronox)

#### Aplicaciones.-

En la plataforma Android existe varias aplicaciones creadas bajo el lenguaje Java, como base importante incluye un cliente de email o correo electrónico, calendario, programas de sms, navegador, contactos y algunos otros servicios mínimos.

Además vienen pre instaladas aplicaciones de Google y permite acceder a los servicios de una forma muy integra, además lo más interesante incluye aplicación Market, este permite instalar aplicaciones desarrollados por terceros.



También la búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.

Se puede desarrollar aplicaciones Android, ya que está abierto para programadores que deseen crear sus propias aplicaciones mediante Android SDK, ya que pueden ser completamente personalizados de acuerdo a los requerimientos del usuario bajo la utilización del lenguaje Java y por último se instala en los dispositivos móviles.

### **Framework o Marco de trabajo de aplicaciones.-**

Como es código abierto está disponible para todos los desarrolladores de aplicaciones android, tienen acceso a todo el código usado en las aplicaciones de base, esto para que no se genere cientos componentes de aplicaciones distintas que realizan la misma acción, esto da la posibilidad de modificar o reemplazar programas por cualquier usuario sin empezar a programar desde cero.

### **Librerías.-**

En su base de datos incluye un conjunto de librerías C/C++, que son expuestas a todos los desarrolladores a través del framework de las aplicaciones Android System C library, librerías de medios, librerías de gráficos, 3D, Sqlite, etc.

### **Runtime de Android.-**

Android incluye bibliotecas base que proporcionan las funciones disponibles en las bibliotecas base del lenguaje Java. La máquina virtual Dalvik es la encargada de correr aplicaciones android.

La ventaja es que en esta máquina virtual puede correr múltiples máquinas virtuales de forma eficiente, el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corren clases compiladas por el compilador de Java que han sido transformadas al formato.dex (DalvikExecutable) por la herramienta incluida "dx".



### **Núcleo Linux.-**

Linux es la base para Android ya que depende para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red, y modelo de controladores. El kernel actúa como una capa de abstracción entre el hardware y software.

#### **2.2.1.4. Kit de desarrollo**

Para desarrollar aplicaciones tenemos:

##### **Android Software Development Kit**

- Se programa sobre un subconjunto del lenguaje Java 5.
- El código Java de Android es 100% retro-compatible con Java, pero no a la inversa.
- El código Java se compila en bytecode para la DVM automáticamente.
- Las aplicaciones andan sobre la DVM.

##### **Android Native Development Kit**

- Permite construir librerías en C\C++.
- Es un complemento del SDK, y NO se pueden construir aplicaciones sólo con el NDK
- El código C\C++ es compilado en lenguaje máquina ARM del procesador
- Cuando se utiliza librerías construidas con el NDK, NO ejecutamos directamente en el procesador, seguimos dentro de la DVM.
- Todos los dispositivos Android son táctiles, y tienen los siguientes botones característicos BACK, HOME, MENU.



### 2.2.1.5. Versiones

- Android 1.5 (Cupcake) en abril 2009
- Android 1.6 (Donut) en septiembre 2009
- Android 2.x (Éclair) en noviembre 2009
- Android 2.1 en enero 2010
- Gingerbread. 2.3 2011
- Para tabletas esta la versión 3.0 / 3.1 / 3.2 conocida como honeycomb en castellano panal de abeja. 2011.

#### **Versión Resiente**

Google presento el Galaxy Nexus y sus características pero también deja datos de la última versión de Android, es decir 4.0 conocida como Ice Sandwich Cream.

Android 4.0 llega con un nuevo diseño, ha sido refinado, y trae muchas mejoras y novedades acerca del mejor sistema operativo.

A continuación detallaremos sobre la última versión de Android.

#### **Características Android 4.0.-**

Tres botones virtuales.- Tres nuevos botones, estos son Atrás, Inicio y Aplicaciones Recientes, estos están en la parte inferior donde se agregan en un estilo barra, estos nuevos botones están presentes en todas las aplicaciones, aunque pueden desaparecer para permitir la vista de pantalla completa (en videos, por ejemplo).

Llega con una barra de acción nueva para acceder a opciones contextuales de cada aplicación.

Desbloqueo mediante el uso de la cara.- Ahora android reconoce el rostro y permite desbloquear su teléfono con sólo hacerle una sonrisa. También se puede utilizar un PIN de seguridad.



Carpetas en la pantalla principal.- Las aplicaciones y accesos directos ahora se pueden agrupar en carpetas en la pantalla de inicio.

AndroidBeam.- Disponible para dispositivos preparados para NFC, la cual ahora va a permitir compartir contactos, artículos, fotos, videos con tan solo tocar 2 dispositivos.

Mejora la multitarea.- Hay un nuevo “botón aplicaciones recientes” que permite a los usuarios saltar al instante de una tarea a otra con la lista en la barra del sistema. La lista aparece para mostrar imágenes en miniatura con las aplicaciones utilizadas recientemente y tocando esa miniatura nos vamos a la aplicación.

Nuevo motor de entrada de voz.- Esto nos proporciona hablar y que el dispositivo escriba todo lo que hablamos.

#### **Mejoras Android 4.0 Ice SandwichCream.-**

- Mejora el rendimiento y la renderización de páginas, hasta 16 diferentes pestañas y el modo de lectura fuera de línea que permite guardar páginas completas que queramos.
- Mejora el teclado y corrección ortográfica.
- El correo electrónico (e-mail) y calendario son mejorado.
- Características mejoradas de la cámara, incluyendo el modo de panorama y timelaps de vídeo de grabación.



### **Principales Celulares que usan el SO Android.-**

Motorola: Motorola Atrix 4G, Quench, Milestone, Milestone 2, Droid X, Motorola i1, Blackflip, Charm, Devour, XT720, XT800, CLIQ, Motorola Defy, Motorola FlipOut, Motorola Quench 2, Motorola Dext.

Samsung: Behold II, Samsung i7500 Galaxy, M900 Moment, Samsung i5700 Spica, Samsung i5800 Teos, Samsung Galaxy A, Samsung i9000 Galaxy S, Samsung Acclaim, Samsung M910 Intercept, Nexus S, Samsung Galaxy Tab, Samsung i5500 Galaxy.

Sony Ericsson: Xperia X10, Xperia X10 Mini, Xperia X10 Mini Pro, Xperia X8, XperiaArc, Xperia Play, Xperia Neo, Xperia Pro.

LG: GW620 Eve, Optimus GT540, GS290 Cookie Fresh, KH5200, LU2300, VS740, KU9500.



### 2.2.2. IOS



Este sistema operativo fue creado por la empresa Apple, fue lanzado al mundo por Steve Jobs el 9 de Enero del 2007 y salió al mercado en EEUU el 29 de Junio del mismo año.

IOS lanzó la versión 1.0, en ese momento no era aún denominada como iPhone OS pero incorporaba aplicaciones comunes: Mail, Fotos, iPod, Calculadora y muchas otras aplicaciones, principalmente fue implementado en iPhone, debido al gran avance tecnológico después fue usado en dispositivos como el iPod, Touch, iPad y el Apple TV. Apple, Inc. además no permite la instalación de iOS en hardware de terceros.

La interfaz de usuario de iOS se basa en la manipulación multitáctiles. Los componentes de la interfaz se ajustan por deslizadores, interruptores y botones. La pantalla principal (“SpringBoard”) es donde se encuentran los iconos de Aplicaciones y el Dock en la parte inferior de la pantalla donde se pueden agregar aplicaciones de uso frecuente, aparece al desbloquear el dispositivo o presionar el botón de inicio.

La pantalla tiene una barra de estado en la parte superior para mostrar datos, tales como la hora, el nivel de batería, y la intensidad de la señal.



### 2.2.2.1. Características

A partir de iOS 4, dispositivos de tercera generación y posteriores soportan el uso de API's para multitarea, específicamente:

Audio

- Voz IP
- Notificaciones push
- Notificaciones locales
- Cambio rápido de aplicaciones

Multitarea (Multitasking): publicaron en el año 2007 sin ningún éxito, luego presentaron oficialmente, su principal característica es navegar por Internet y al mismo tiempo utilizar las redes sociales.

Múltiples cuentas de Exchange: Integra cuentas de Microsoft Exchange como el usuario lo requiera, también sincroniza cuentas de correo electrónico como GMail y MobileMe, además el firmware IOS4 es compatible con Exchange 2010.

Cámara de 5x y Zoom incorporado: ahora puede realizar acercamiento o alejar la imagen de una forma gratuita, en las aplicaciones anteriores tocaba pagar por esta característica.

Wi-Fi estable en modo de espera: En los dispositivos iPod Touch se puede recibir llamadas de Skype y otros servicios de VoIP en modo de suspensión.

Stream en los videos de YouTube con Safari: Si existe algún problema con la aplicación de YouTube o se elimine por cualquier motivo, ahora pueden ver los videos en el navegador Safari.

Soporte de Bluetooth en el teclado: IOS 4 es compatible con el teclado Bluetooth Tethering, antes no tenía esta mejora.



Mejora de archivos adjuntos en el correo electrónico: Se puede tener una vista de archivos adjuntos, si el usuario lo requiere, antes de descargarlos en su dispositivo móvil.

Las aplicaciones principales son Phone, Safari, Ipod y Mail. Entre las secundarias están YouTube, iTunes, AppStore, Nike+Ipod y GameCenter (red social al estilo Xbox Live).

Este sistema de Apple no se deja modificar, no existe ningún tipo de cambio en la forma del funcionamiento del dispositivo y tener un control de sus aplicaciones existentes, además de que el iPhone tiene un costo muy elevado en el mercado, debido a su único fabricante y un modelo único en todo el mundo.

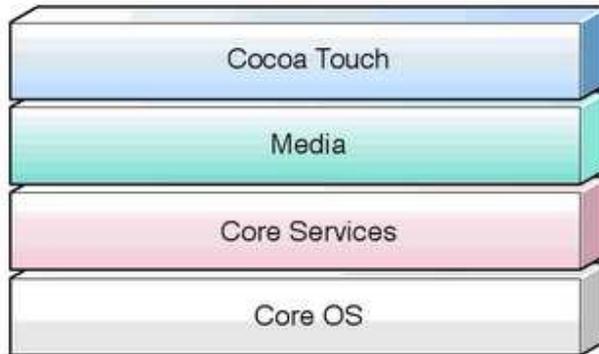
IOS no soporta Adobe Flash o Java, lo que afecta a los sitios web que utilizan estas tecnologías. Steve Jobs dio un comunicado criticando a Flash por ser inseguro, con errores, consumir mucha batería, ser incompatible con interfaces multitouch e interferir con el servicio App Store.

iOS soporta HTML5 como una alternativa a Flash.



### 2.2.2.2. Arquitectura

La arquitectura es similar a Mac OS X, al más alto nivel. iOS actúa como un intermediario entre el hardware subyacente y las aplicaciones que aparecen en la pantalla.



*Figura 2.2: Arquitectura iOS*

*Fuente: <https://sites.google.com/site/tecnologiaiostm/desarrollo-de-aplicaciones/arquitectura-ios>*

Está basado en una variante del Mach kernel que se encuentra en Mac OS X. Este sistema operativo contiene 4 principales capas de abstracción:

- Capa del núcleo del sistema operativo
- Capa de servicios principales
- Capa de medios de comunicación
- Capa de Cocoa Touch (pantalla multitáctil)

Las 2 últimas capas del sistema son los servicios y las tecnologías principales en las que se apoyan todas las aplicaciones, en el nivel alto se ven las capas que sujetan servicios y tecnologías más sofisticadas para el sistema operativo.

Las estructuras de alto nivel proporcionan abstracciones orientadas a la construcción de objetos de nivel inferior, lo efectúan para poder escribir el código de manera más sencilla y fácil ya que disminuye la cantidad de código que se tiene que escribir y



encapsular, con características potencialmente complejas. La estructura de los niveles bajos está accesible para los usuarios o desarrolladores que prefieren utilizar.

Sin duda este sistema operativo para dispositivos móviles con marca Apple, es una de las mejores para el usuario debido a su tecnología basado en su Touch Screen e implementado mediante el framework Cocoa Touch desarrollado por Apple.

### **2.2.2.3. Kit de desarrollo**

Para desarrollar aplicaciones tenemos:

En octubre de 2007, el fundador Steve Jobs anunció un Kit de desarrollo de software o SDK para desarrolladores, además este SDK fue liberado el 6 de marzo de 2008, teniendo como objetivo permitir a los desarrolladores hacer aplicaciones para el iPhone e iPod Touch, así como probarlas en el "iPhone simulator". De cualquier manera, solo es posible utilizar el app en los dispositivos después de pagar la cuota del iPhone Developer Program.

Los programadores pueden poner un precio mínimo \$0.99 dólares a sus aplicaciones para distribuir las en el App Store, de donde recibirán el 70% del dinero que produzca la aplicación. En alternativa, el desarrollador puede optar por lanzar la aplicación gratis, y de esta forma no pagar ningún costo por distribuir la aplicación (excepto por la cuota de la membresía).

### **2.2.2.4. Versiones**

- Versión y Dispositivos que la soportan
- 3.1.3 iPhone e iPod touch (1ª generación)
- 4.2.1 iPhone 3G e iPod touch (2ª generación)
- 5.0 iPhone 3GS, iPhone 4, iPhone 4S, iPod touch (3ª y 4ª generación), iPad e iPad 2
- 5.0 Apple TV (2ª generación)



**Versión Resiente.-**

A continuación detallaremos sobre la última versión Iphone

**iOS 5.x: Quinta versión del OS**

Apple utilizó el concepto de reproductor de música portátil. Después de la aparición del iPod Touch vieron la necesidad de convertirlo en un teléfono. Unido a iTunes y la AppStore, el iPhone se convirtió en el terminal de referencia, millones de programas de bajo costo que convertían el teléfono móvil en lo que quisiéramos: teléfono, navegador, mp5, videoconsola o meteorólogo lo hicieron realidad.

El 6 de junio de 2011 fue el lanzamiento de la última versión de iOS. Desde la penúltima versión es imposible la actualización en los dispositivos antiguos, en este caso: el iPhone 3G y el iPod touch 2G, fue lanzado para iPhone 3GS, iPhone 4, iPhone 4S, iPod touch 3G, iPod touch 4G, iPad e iPad 2 el 12 de octubre de 2011.

Estas son las principales características de iOS 5:

- Notification Center, en un solo lugar combinas todos los tipos de notificaciones, es igual que Android, se baja el panel de arriba y te salen todas las notificaciones.
- La cotización de las acciones de la app Bolsa aparece en la barra de arriba, como si fuera un canal de economía de la web, se puede eliminar una notificación con solo hacer clic en el botón de la “x”.
- Newstand: para suscribirse a periódicos y revistas.
- Se ha creado un sitio en la App Store que combina periódicos, revistas y se puede descargar y leerlas off-line.
- Twitter integrado dentro iOS 5. Se integra con la mayoría de las aplicaciones de iOS (YouTube, Fotos, Maps, Contactos, etc.).
- Safari Reader de Mac OS X disponible en iOS 5.
- Pestañas en iOS 5.
- Si se hace doble clic en el botón Home bloqueas la pantalla.



- El botón de volumen sirve para disparar la foto.
- Puedes hacer zoom con el gesto de dos dedos.
- Se puede bloquear el enfoque y la exposición de la cámara en iOS 5.
- Cortar, pegar y rotar integrado dentro de la app cámara.
- Reducción de ojos rojos.
- Mail en iOS 5 con texto en formato enriquecido.
- PC Free: sincronización en online sin ordenador de por medio.
- Game Center: se pueden comprar juegos directamente que te haya recomendado un amigo en Game Center.
- Nuevo servicio de mensajería entre usuarios de iOS5. Gratis entre usuarios de iOS 5. Un Whatsapp para iOS 5.
- El iMessages soporta 3G y WiFi.
- Las fotos y vídeos en iMessages se envían en alta calidad.
- Nueva aplicación de música para el iPad.
- Patrones de vibración personalizables.
- Soportará iPhone 3GS.
- AirPlayMirroring, vía inalámbrica, visualiza las acciones que realiza el iPad 2 directamente en la pantalla de un TV HD, a través del Apple TV.
- Un teclado reformado, para facilitar aún más la escritura en el iPad.

### **Disponibilidad**

- La versión beta del software iOS 5 y del SDK están disponibles para los miembros del iOS Developer Program en [developer.apple.com](http://developer.apple.com).



### 2.2.3. SYMBIAN



La compañía británica Psion comenzó con el desarrollo del predecesor de Symbian llamado EPOC OS en 1987.

Este sistema operativo móvil utiliza empresas importantes de dispositivos móviles como son los Smartphones tales como Nokia, Sony Ericsson, Motorola, Samsung, Panasonic.

Este sistema operativo es utilizado para varias tareas a la vez, creado principalmente para que la batería tenga un mayor rendimiento, una de las ventajas, se debe a largos periodos de tiempo sin necesidad de ser reseteado y tiene una gran capacidad de integración de aplicaciones lo cual lo hace un sistema operativo versátil o adaptable y de alta capacidad por su memoria y los recursos gestionados.

En 1998 los fabricantes de smartphones se aliaron para desarrollar y reforzar un software único en sus teléfonos, con la primera versión de Symbian lanzada en 2000. En el año 2001 fue lanzado el primer teléfono cuyo sistema operativo fue el Nokia 9210 Communicator, que permitía a los usuarios instalar sus propias aplicaciones.

Symbian es conocido sustancialmente por ser el primer sistema operativo de smartphones que permitía una especie de multitasking, es decir, trabajar con varias aplicaciones a la vez sin la necesidad de cerrarlas, como por ejemplo abrir la calculadora y dejarlo en pantalla consumiendo un mínimo de recursos.

Durante los años 2009 y 2010 la empresa Nokia decidió trasladar el soporte y desarrollo del sistema operativo Symbian a la consultora Accenture, es decir que a finales de septiembre de 2011 terminaron con su operación, una vez terminado el desarrollo de la nueva versión Symbian Belle y última versión de Symbian en la que Nokia participó de forma exclusiva. Además oficialmente Symbian tendrá soporte hasta el año 2016, al no ser competencia para la nueva versión de smartphones con sistemas operativos de última generación como Android, iOS o Windows Phone.



### 2.2.3.1. Características

- Symbian OS posee un núcleo de tiempo real.
- Telefonía móvil multi-modo integrada.- Symbian OS integra la potencia de computación con la telefonía móvil, aportando servicios avanzados de datos al mercado masivo.
- Entorno abierto de aplicación.-Permite a los teléfonos móviles ser una plataforma de aplicaciones y servicios (programas y contenido) que pueden ser desarrollados en una amplia gama de lenguajes y diferentes formatos.
- Estándares libres e interoperabilidad.-Con una implementación flexible y modular, Symbian OS proporciona un sistema esencial de APIs (Application Programming Interface) y de tecnologías compatibles para todos los teléfonos Symbian. Soportan los estándares más comunes de la industria.
- Multi-Tarea.- Está basado en una arquitectura de micro-kernel e implementa funciones de tipo multi-tarea. Los servicios del sistema como la telefonía, las redes middleware y las aplicaciones, funcionan en sus propios procesos.
- Orientado a objeto y Basado en componente.- Está diseñado y pensado exclusivamente para los dispositivos móviles, usando las técnicas avanzadas de OO (orientado a objetos), para construir una arquitectura flexible basada en componentes.
- Diseño flexible del interfaz de usuario.- Permitiendo un diseño gráfico flexible del interfaz de usuario en el sistema y usando el mismo sistema operativo como base en diferentes diseños, facilita el proceso de desarrollo de la aplicación.
- Robustez.- mantiene el acceso inmediato a los datos del usuario.
- Asegura la integridad de los datos, incluso en la presencia de comunicación no fiable y de falta de recursos como la memoria, el almacenamiento y la energía.



### 2.2.3.2. Arquitectura

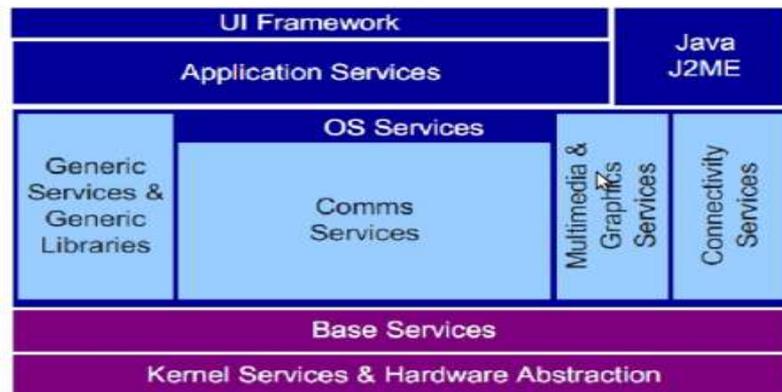


Figura 2.3: Arquitectura Symbian

Fuente: <http://es.scribd.com/doc/56182084/Sistema-operativo-symbian>

#### Framework de la interfaz de usuario

- Es la capa superior de Symbian OS.
- El sistema operativo entrega una interfaz de prueba llamado Tech View a los fabricantes de móviles, esta interfaz no es ni completa ni de calidad.
- A final los fabricantes instalan una interfaz de calidad de producción propia.

#### Servicios de aplicación

- Application engines (Motores de Aplicación)
- Los motores de aplicación se basan en la biblioteca del usuario, que proporcionan una funcionalidad para la manipulación de un tipo específico de datos como por ejemplo la agenda. Los motores son utilizados frecuentemente por aplicaciones diferentes.
  - ✓ Cliente Sync MLData
  - ✓ Web engine (Motor de Web)
  - ✓ Mensajería
  - ✓ SMS (Short Message Service)
  - ✓ EMS (Enhanced Messaging Service)



- ✓ MMS (Multimedia Messaging Service)
- ✓ Email
- ✓ Fax

### **Servicios del sistema operativo**

- Multimedia
- Entorno de Aplicación
  - ✓ Entorno de GUI (Interfaz gráfico de usuario)
  - ✓ Soporte de servicios de aplicación
  - ✓ Support Internacional
  - ✓ Utilidades de textos de gráficos
  - ✓ Redes de Comunicación Área Personal
  - ✓ Bluetooth
  - ✓ Infrarrojo
  - ✓ USB
- Infraestructura de Comunicación
- Networking
  - ✓ HTTP
  - ✓ WAP
  - ✓ Conectividad de PC
  - ✓ Gestión de conexión
  - ✓ Herramientas de conexión
- Telefonía
  - ✓ Telefonía GSM/EDGE
  - ✓ GSM



- ✓ GPRS
- ✓ EDGE
- ✓ Telefonía CDMA
- ✓ CDMA (IS-95)
- Seguridad
  - ✓ Módulo de Criptografía
  - ✓ Entorno de Criptografía
  - ✓ Certificado de gestión de módulo
  - ✓ Instalación de Software

### **Servicios base**

Acceso a dispositivos, provee el framework para drivers de los dispositivos.

- Librerías de usuario
- Arquitectura de Target CPU
- Drivers de dispositivos
- Servidor de ficheros
- Librería Standard

### **Kernel (núcleo)**

- Abstracción del Hardware del lado del usuario.
- Framework para drivers de media.
- Provee las primitivas del SO y firmware de abstracción de Hardware.
- DBNS: Acceso en API a BD Relacional
- Los servicios del Kernel se encargan de:
  - ✓ Bootstrapping.
  - ✓ Hilos, procesos, direcciones de memoria, timers.



- ✓ Manejo de Interrupciones.

### 2.2.3.3. Kit de desarrollo

Kit de desarrollo de Software.

- Equipos Symbian OS.
- Herramientas de Desarrollo de C++.
- Eliminación de fallos Sobre aplicación.
- Eliminación de fallos Sobre objetivo núcleo.
- Pasarela de integración de Hardware.
- Integración y Test de Telefonía.
- Desarrollo de Aplicación.
- Licencias SDKs de Symbian OS.
- C++
- Java
  - ✓ Personal Java J2ME
  - ✓ Java Phone
  - ✓ MIDP



#### 2.2.3.4. Versiones

Existen varios tipos de Symbian, Nokia y otros distribuyen móviles con la serie S60 la que cuenta con una aceptación considerable y avance en desarrollo, y la S80, S90, de los cuales hay unos pocos terminales. **UIQ**, usado principalmente en Sony Ericsson y Motorola es una versión ajustada a estos terminales. Estas series también se distinguen en diferentes versiones.

EPOC32 1.0 5.1, con el nombre de EPOC32 en año 1997 con el handledPsion Series 5.

Symbian OS 6.0, con el nombre de Symbian OS en el año 2001 con el Nokia 9210 Communicator.

Symbian OS 6.1, con el nombre de Symbian S60 en el año 2002 con el Nokia 7650 .

Symbian OS 6.1, con el nombre de Symbian S60, Feature Pack 1 en el año 2002 con el Nokia 3650.

Symbian OS 7.0, con el nombre de Symbian S60 2nd Edition en el año 2003 con el Nokia 6600.

Symbian OS 7.0, con el nombre de Symbian S60 2nd Edition, Feature Pack 1 en el año 2004 con el Nokia 3230.

Symbian OS 8.0, con el nombre de Symbian S60 2nd Edition, Feature Pack 2 en el año 2004 con el Nokia 6630.

Symbian OS 8.1, con el nombre de Symbian S60 2nd Edition, Feature Pack 3 en el año 2005 con el Nokia N70.

Symbian OS 9.1, con el nombre de Symbian S60 3rd Edition en el año 2006 con el Nokia 3250

Symbian OS 9.2, con el nombre de Symbian S60 3rd Edition, Feature Pack 1 en el año 2007 con el Nokia 5700 XpressMusic

Symbian OS 9.3, con el nombre de Symbian S60 3rd Edition, Feature Pack 2 ene el año 2008 con el Nokia 5320 XpressMusic



Symbian OS 9.4, con el nombre de Symbian S60 5th Edition en el año 2008 con el Nokia 5800 XpressMusic.

Symbian OS 9.5, con el nombre de Symbian^3 2010 con el Nokia N8.

Symbian OS 9.5, con el nombre de Symbian Anna en el año 2011 con el Nokia X7 y el Nokia E6 y en forma de actualización.

### **Versión Resiente.-**

A continuación detallaremos la versión Symbian Belle (Symbian OS 10.1).

Como estrategia Nokia considero que debía cambiar su nombre hacia al público de Symbian Belle a Nokia Belle, para los programadores o desarrolladores el sistema operativo se denomina Symbian. Symbian OS 10 viene instalada por defecto en los nuevos Nokia 603, 700 y 701 y también estaría disponibles para los teléfonos que funcionan con Symbian^3 y Symbian Anna. Entre las nuevas características se dispone de estabilidad, consumo mínimo de recursos y la gestión de la memoria RAM del sistema, además se incorporó nuevas aplicaciones oficiales de Microsoft (Microsoft Apps), entre las que se incluyen Word, Excel, PowerPoint y OneNote, una interfaz gráfica de usuario totalmente cambiado por sus menús, widgets, barra de navegación, menú de notificaciones deslizable y nueva barra de notificaciones entre otras que deja más espacio y alternativas para las aplicaciones. Podríamos decir que sería la extinción de este sistema operativo ya que terminaría el soporte y desarrollo mediante la consultora Accenture.

Esta versión fue lanzada para todos los terminales compatibles menos el Nokia 500 el 7 de febrero de 2012, pero para el Nokia 500 se lanzó el 15 de febrero de este año.



#### 2.2.4. BLACKBERRY OS



Es un sistema operativo móvil desarrollado por Research in Motion para sus dispositivos BlackBerry.

A finales de la década de los 90 el internet y el uso de correo electrónico se convirtieron en los componentes primordiales para la comunicación informática, las empresas AOL y Yahoo fueron los pioneros para dar un realce a estas tecnologías de la información, por lo cual los fundadores de RIM se dedicaron a elaborar una visión para habilitar la mensajería de correo electrónico de los usuarios finales.

En 1999, aparecieron los primeros handheld además de hacer las funciones básicas de un teléfono móvil como por ejemplo realizar llamadas, enviar SMS, etc. estos dispositivos permiten el acceso a correo electrónico, navegación web y sincronización con programas como Microsoft Exchange o Lotus Notes.

Este sistema operativo permite multitarea y soporta diferentes tecnologías particularmente la trackwheel, trackball, touchpad y pantallas táctiles.



#### 2.2.4.1. Características

Tiene pantalla grande, una compatibilidad óptima con modernos servicios inalámbricos tales como redes General Packet, Radio Service (GPRS) y Code Division Multiple Access (CDMA) y la integración con el correo electrónico corporativo a través del BlackBerry Enterprise Server (BES).

Este sistema operativo fue desarrollado bajo la plataforma Java, ya que en base a este las comunicaciones son sólidas y seguras, así como permitir a los desarrolladores de software crear aplicaciones adicionales de software y soluciones.

El SO BlackBerry se enfoca a la orientación del uso profesional como gestor de correo electrónico y agenda. Desde la versión cuarta, se considera la sincronización del dispositivo móvil con el correo electrónico, el calendario, tareas, notas y contactos de Microsoft Exchange Server.

BlackBerry Enterprise Server (BES) brinda servicios como el acceso, control y organización del correo electrónico a organizaciones grandes identificando a cada usuario con un único valor BlackBerry PIN. El software BlackBerry Internet Service facilita acceso a internet y a correo POP3 / IMAP / Outlook Web Access sin utilizar BES para los usuarios pequeños que utiliza este servicio.

Los desarrolladores tienen la facilidad de crear programas para BlackBerry pero existen restricciones para ciertas funciones ya que necesitan tener una firma digital para poder ser asociados a una cuenta de programador de RIM.

Además BlackBerry se diferencia de los demás por sus teclados QWERTY al estilo de un teclado de PC.

También se caracteriza por sus políticas de seguridad ya que cuenta con un archivo "policy.bin", este se encuentra a un nivel muy bajo del BlackBerry lo cual pertenece exclusivamente al sistema sin la necesidad de encontrar en el sistema de archivos y menos aún copiarlo o eliminarlo.

Puede asignar políticas de TI a dispositivos BlackBerry para satisfacer los requisitos de la política de seguridad de su empresa y para reflejar las necesidades de los



usuarios que utilicen los dispositivos BlackBerry. Por ejemplo, puede crear una política de TI, configurar las reglas de política de TI para la característica de nivel ejecutivo y los requisitos de seguridad, agregar ejecutivos a un grupo y asignar la política de TI al grupo.

**Algunas políticas de seguridad del dispositivo como tal son:**

- No guardar mensajes enviados
- Contraseña necesaria
- Desactivar el centro de aplicaciones
- Desactivar Buscar actualizaciones
- Modo de introducción de PIN mínimo
- Desactivar el acceso telefónico a redes
- Desactivar perfil de acceso a SIM
- Desactivar JavaScript en el explorador
- Desactivar mensajería sin conexión para Enterprise Messenger
- Desactivar la cámara de fotos y cámara de video.
- Desactivar MMS (Servicios Multimedia)
- Habilitar el PIN de mensajería
- Permitir SMS
- Desactivar GPS
- Desactivar el almacenamiento masivo USB



#### **2.2.4.2. Arquitectura**

- Escrito en: Java, C++
- Licencia propietaria
- Tipo de núcleo: Basado en Java
- Interfaz gráfica por defecto: GUI
- Plataformas soportadas: Línea de Smartphome de BlackBerry

#### **2.2.4.3. Kit de desarrollo**

Herramientas Necesarias para desarrollar:

- Sun JDK 1.6.x
- Eclipse SDK
- Blackberry JDE Plug-in para Eclipse(BlackBerry Java SDK 5.0.0)

#### **2.2.4.4. Plataforma**

QNX es una plataforma de sistema operativo de tiempo real, que se ejecuta bajo procesadores INTEL: 386, 486, Pentium y clones como AMD, Nat Semiconductor, Cyrix y SGS Thompson, que agrega la arquitectura del modelo de proceso universal (UPM). Esta arquitectura de UPM le permite disminuir el tiempo de desarrollo ya que permite realizar la recopilación sin la necesidad de cambiar el kernel e implantar nuevos drivers, esta plataforma QNX es totalmente escalable, así que se puede usar el mismo sistema operativo para todo, desde la electrónica hasta los sistemas de control del BlackBerry.

El sistema operativo QNX está basado en estándares como HTML, CSS, Java y soporta HTML5, Flash y OpenGL. Las nuevas características presentan facilidades para desarrollar, QNX ayudará a RIM a dar ventajas competitivas frente a las competencias e ir más allá en el mercado empresarial y el de consumo, ya que si desea permanecer en el mundo de las empresas telefónicas.



Blackberry realiza una combinación con lo mejor de su sistema operativo con lo mejor de QNX (Sistema operativo para la tablet Playbook) por lo que será compatible con aplicaciones desarrolladas para BlackBerry OS 6 y OS 7.

También existe la manera de desarrollar software usando estas API y propiedad de las API de BlackBerry. Cualquier aplicación que hace uso de ciertas funciones restringidas debe ser firmada digitalmente para que pueda estar asociado a una cuenta de desarrollador de RIM, además garantiza la autoría de una aplicación pero no garantiza la calidad y seguridad del código, esta empresa RIM facilita herramientas para el desarrollo de aplicaciones y temas para la BlackBerry, los temas pueden ser descargados en los dispositivos BlackBerry a través de BlackBerry App World, OverThe Air (OTA) a través del navegador de BlackBerry, o a través de BlackBerry Desktop Manager.

#### **2.2.4.5. Versiones**

En el BlackBerry DevConAmericas 2011, Research In Motion anunció su plataforma BlackBerry BBX, consiste en lo mejor de la plataforma BlackBerry y lo mejor de la plataforma QNX para amplificar servicios como conectar a personas, dispositivos y contenidos. Ello incluirá desde los equipos y tabletas junto con su contenido y la sincronización en la nube.

#### **BBX**

La plataforma BBx soportara los servicios en la nube de blackberry, también soporta aplicaciones desarrolladas utilizando cualquiera de las herramientas para blackberryentre ellas el sdk nativo, adobe air/flash y webworks/HTML5.

También incluye Blackberry UI framework para contenidos gráficos mejorados, una herramientas visual y aportará capacidades de “Super App” para la integración profunda entre aplicaciones, los servicios push estarán activos.

RIM facilita herramientas para los desarrolladores y presentó el SDK nativo 1.0 “Gold” para la BlackBerry PlayBook, el cual permitirá crear aplicaciones nativas en C/C++, además de poder desarrollar juegos en 2D y 3D, así como usar OpenGL ES



2.0 y Open AL.

Para el Blackberry PlayBook, se cuenta con soporte para Adobe Air 3.0 además se anunció la versión del PlayBook SO 2.0 para desarrolladores que contiene el BlackBerry Runtime y el complemento BlackBerry para Android Development Tools para ejecutar aplicaciones Android.

Al inicio del año 2012 anunciaron oficialmente sobre su nombre de esta plataforma llamado blackberry 10 luego que una corte de EE.UU dicto una orden de restricción que impide a RIM utilizar el nombre de BBX, ya que va ser la evolución de sus sistemas operativos.



## 2.2.5. WINDOWS PHONE.- MICROSOFT



La empresa Microsoft dueña de este producto Windows Phone, conocido como

Windows Mobile es un sistema operativo móvil diseñado para dispositivos móviles o teléfonos inteligentes Smartphone, Windows Phone tiene una interfaz natural de usuario con la finalidad del uso fácil.

Este sistema operativo se basa en el kernel del Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows.

Está basada en paneles personalizables y los únicos botones físicos son la flecha para ir atrás, el botón central de inicio con el logo de Windows y el botón de búsqueda.

Al ser desarrolladores de la consola XBOX, se podrá acceder a los mejores juegos online y tendrá integración por medio de Windows Live y Microsoft Office por lo cual es considerada uno de los mejores sistemas operativos más completos del mercado.

### 2.2.5.1. Características

La interfaz presenta novedades importantes las cuales se presenta en la pantalla "Hoy" la fecha actual, la información del usuario, las citas, mensajes, E-mail, y las tareas. En si el usuario puede cambiar o eliminar el tema predeterminado configurando nuevos complementos, como por ejemplo, alarma, temperatura, estado de la batería.

La seguridad en Windows Phone 7 es muy importante durante el desarrollo de las aplicaciones y es por eso que tiene unas características que son muy indispensables en el desarrollo de las mismas ya cuenta con seguridad al enviar y recibir datos por internet.



Durante el desarrollo se ha proporcionado nuevas funciones y aspectos de seguridad mediante la estructura Silverlight para aplicaciones web, bajo el control de Sandbox las aplicaciones son controladas y delimitadas al acceso de ficheros u otras aplicaciones.

Cuando los desarrolladores programan, deben tener en cuenta como invocan las clases para desarrollar la tareasdeterminadas utilizando código con permisos, en este momento es cuando los Launchers y Choosers se usan permitiendo a los usuarios ejecutar tareas comunes, pueden diferenciar las aplicaciones y facilitar una funcionalidad adicional.

Existen otras clases o librerías diferentes para controlar el acceso a los datos del el teléfono desde las aplicaciones y a los servidores donde estén almacenados, las cuales son:

- System.Security.Principal: Esta clase da información relativa a la gestión de los usuarios y sus roles.
- System.Security.Permissions: Esta clase permite características para el acceso a ciertos recursos.
- System.Security.Cryptography: Esta clase proporciona características de encriptación y funciones como: AES, SHA1, SHA256 y HMAC.

Ahora con la versión de Windows Phone Mango 7.5 aparecen nuevas características de seguridad como:

- Inclusión de passwords alfanuméricos.
- Soporte para derechos de administración para la visualización y envío de correos cifrados.



2.2.5.2. Arquitectura

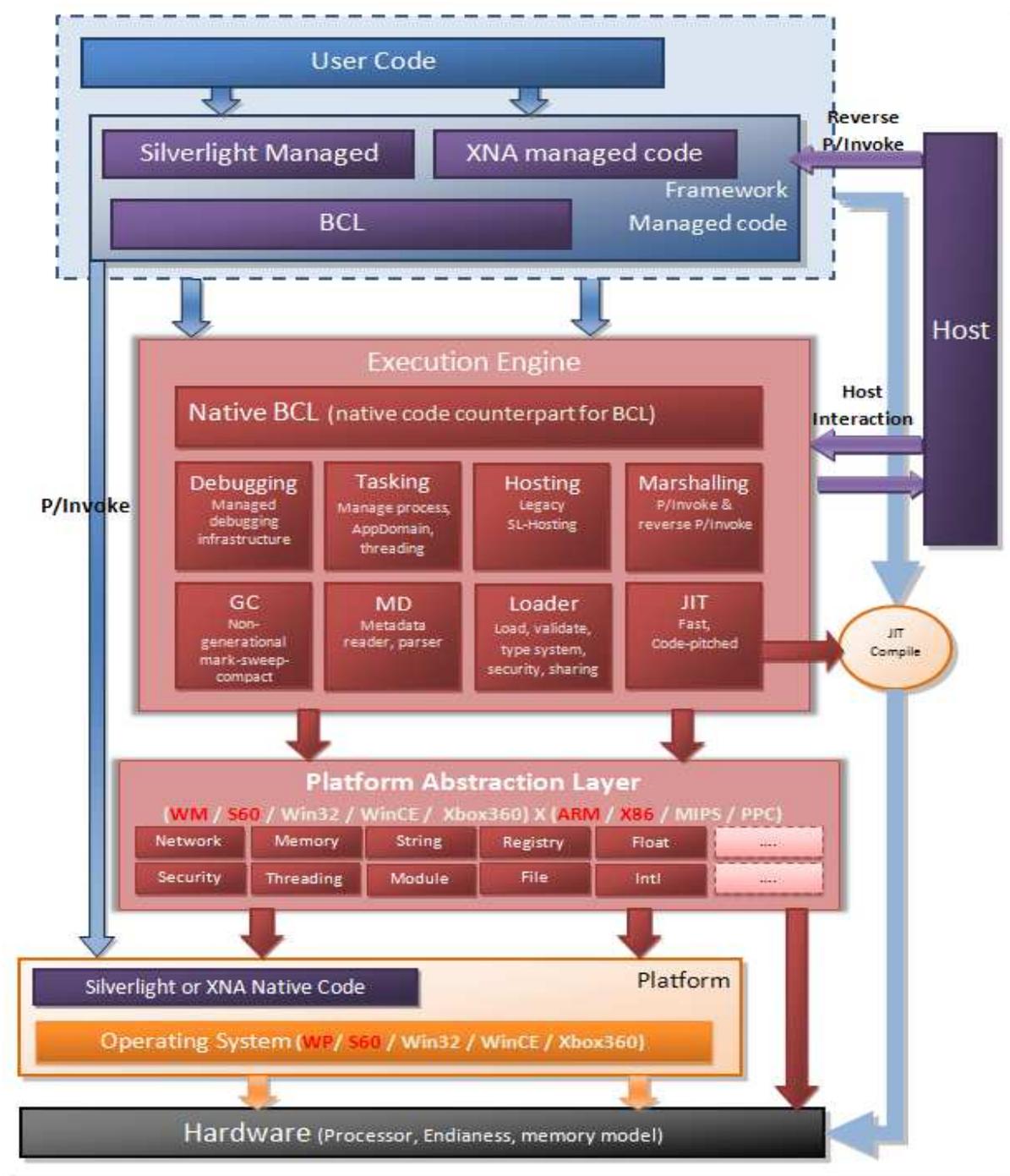


Figura 2.4: Arquitectura de Windows Phone

Fuente: <http://msdn.microsoft.com/es-es/library/jj130729.aspx>



En esta arquitectura de Windows Phone el usuario tiene la facilidad de utilizar el código que son: Silverlight y XNA, estos códigos se encuentran agregados en una plataforma de ejecución (Framework Managed Code) donde se administrase aísla, este aislamiento o limitación de procesos llamado Sandbox ejecuta aplicaciones con seguridad y de manera separada, cada aplicación se ejecuta en su propio entorno limitado, aislado, para evitar que los archivos de la aplicación sean dañados por otras, además previene cualquier tipo de acceso directo a la plataforma, por lo tanto la única salida al sistema operativo es por el framework de código administrado.

En la arquitectura se encuentra el código administrado que contiene biblioteca de base de código (BCL), la cual se puede acceder al código del framework Silverlight y el código del framework XNA, estos códigos se comunican con la plataforma a excepción del código de usuario, también puede llamar al sistema subyacente haciendo un previo análisis de seguridad.

La Máquina Virtual está codificada contra una capa de abstracción llamada PAL (PlatformAbstractionLayer) que habilita el framework a ejecutar sobre diferentes dispositivos móviles que tenga una PAL para dicha plataforma.

El compilador JIT (Just In Time) es llamado como compilación en tiempo de ejecución y es una técnica para optimizar el rendimiento de sistemas de programación en el momento de su compilación. Durante implementación de esta arquitectura, cada procesador que se soporte debe haber un compilador JIT separado.

Todas estas características de arquitectura están incluidas en un host, este utiliza los servicios en tiempo de ejecución para poder ejecutar el código administrativo.

El host para este sistema operativo es conocido como Windows PhoneTask Host, este corresponde a un proceso que funciona en segundo plano. Este host se comunica con el motor de ejecución a través de interfaces del hosting.

Además una de las ventajas es la portabilidad con el diseño de .NET Compact Framework.



Los procesadores soportan formatos de tipos de datos Big-Endian, donde los datos de más de un byte se almacenan en memoria considerando el orden de llegada para realizar su escritura, también existe otro tipo de formato llamado Little-Endian, donde los datos se almacena de mayor a menor, por ejemplo: la arquitectura de Intel i386 almacena datos en la memoria en orden haciendo más sencilla la interpretación binaria de los datos, a esto se le denomina Little-Endian, de mayor a menor. Existe varios tipos de procesadores que manejan los dos formatos, estos son: ARM, PowerPC, AlphaDEC, HP-PA, RISC y MIPS. Microsoft es el único que soporta la XBOX 360 ya que tiene ese tipo de procesador.

### **2.2.5.3. Kit de desarrollo**

Para desarrollar aplicaciones para este sistema operativo móvil tenemos la herramienta .NET Compact Framework, ya que permite desarrollar aplicaciones para dispositivos móviles tanto para dispositivos con tecnologías modernas y con recursos limitados, además al mínimo consumo de energía, también existe la herramienta Pitching la cual se encarga en el Rutine .NET de liberar espacios de RAM ya que existen códigos ya compilados y no servibles con el objetivo de reutilizar esos espacios de memoria con nuevos objetos.

### **2.2.5.4. Versiones.-**

- Windows Phone 6.5 en 6 de octubre de 2009
- Windows Mobile 6.1 en 1 de abril de 2008
- Windows Mobile 6 en 12 de febrero del 2007
- Windows Mobile 5.0 en 9 de mayo del 2005
- Windows Mobile 2003 Second Edition en 24 de marzo de 2004
- Windows Mobile 2003 es conocido también como Windows CE 4.20.
- PocketPC 2002



### **Versión Resiente.-**

Windows phone 8

Microsoft encontró alternativas para la integración que pretende lograr con sus sistemas operativos tanto móvil como la de escritorio ya que será única y bastante innovadora. Esta compatibilidad consiste en aplicaciones que sean sumamente fáciles de portar o instalar del escritorio al celular o incluso compatibles hasta cierto punto, por lo tanto los programadores podrían desarrollar una sola vez y lanzar en distintas plataformas, posiblemente no nada más escritorio y móvil sino también en el Xbox.

Hardware.- Existirá una mayor optimización y mejor aprovechamiento de los recursos como doble núcleo, nuevas resoluciones de pantalla y utilización de tarjetas microSD. Adicionalmente, el NFC tendrá su debut en la plataforma.

Software.- Se estima que haya más de 100,000 aplicaciones en el Windows Marketplace, por lo que se ofertara opciones bastante sólida para las necesidades básicas.

Recordemos que Microsoft compró a Skype, la cual integra esta herramienta al sistema operativo siempre y cuando sean controladas ya que iría en contra de los intereses de las operadoras telefónicas quienes siguen aferradas a sus servicios de voz.

Al igual que en Android 4.0 "Ice Cream Sandwich", Microsoft incorporará una herramienta de control de consumo de datos llamado DataSmart que permitirá determinar y ajustar la manera en la que las aplicaciones abusan de los pocos MB que las operadoras dan.

El navegador web Internet Explorer tiene sus novedades ya que proporcionará un sistema de compresión de datos desde los servidores de Microsoft. También ofrecerá servicios nuevos que podrán interesar a clientes corporativos como cifrado de datos y aplicaciones especialmente diseñadas para la administración de redes empresariales, ya que podrán competir con BlackBerry 10.



## 2.3 MERCADO DE LOS SISTEMAS OPERATIVOS

En los últimos años el sistema operativo móvil más difundido en Latinoamérica, sigue siendo Symbian con un 31.74% al último mes (febrero) y lo ha sido desde hace varios años. Sin embargo, es sólo cuestión de meses o semanas para que Android lo supere de una vez por todas pues va subiendo rápidamente con un 30.93%. Asimismo, iOS sube establemente con un 12.81%.

Además las tendencias a disminuir vienen sistemas operativos de Samsung con un 7.53%, BlackBerry OS con 3.43%, sistema de Sony Ericsson con 1.61% cayendo en picada.

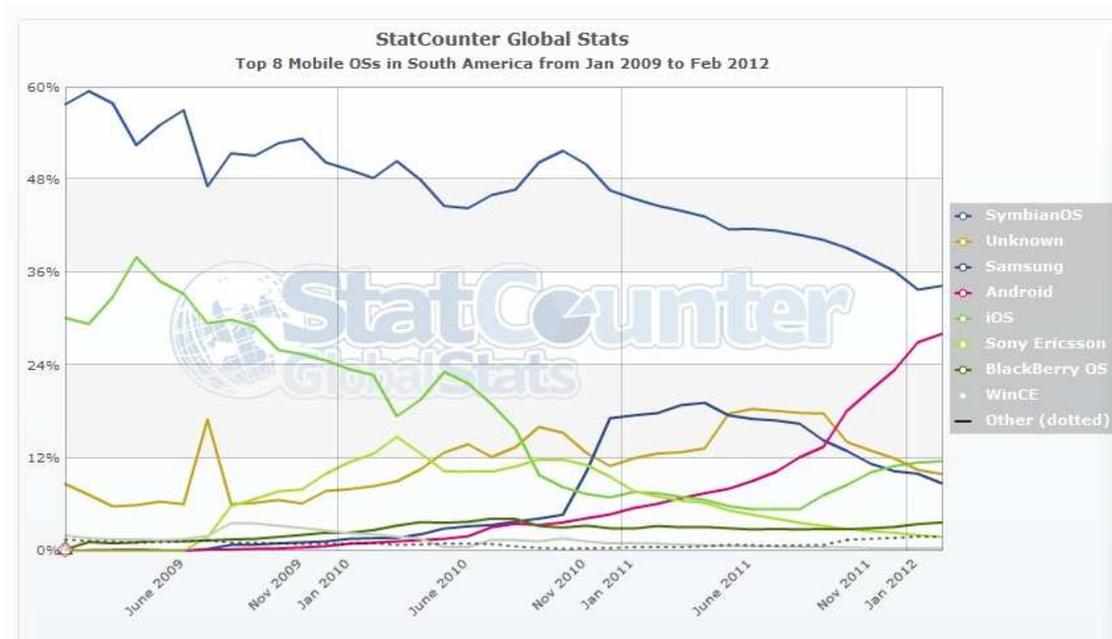


Figura 2.5: Mercado Latinoamérica.

Fuente: [http://gs.statcounter.com/#mobile\\_os-sa-monthly-200901-201202](http://gs.statcounter.com/#mobile_os-sa-monthly-200901-201202)



A continuación especificaremos los porcentajes de mercado de los sistemas operativos en Ecuador.

Symbian ocupa la gran mayoría (31.98%) pero cede lugar hacia las tres competencias principales del momento: Android (25%), iOS (19.55%) y BlackBerry OS (15.35%). El resto se ubica por debajo del 4%.

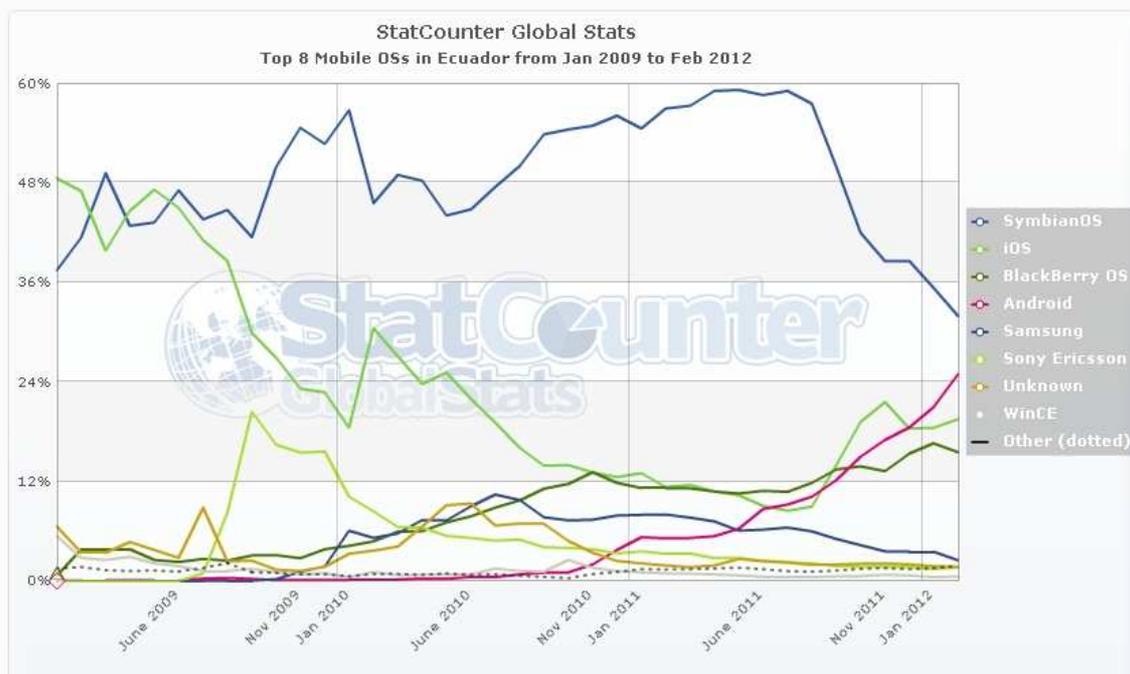


Figura 2.6: Mercado Ecuador

Fuente: [http://gs.statcounter.com/#mobile\\_os-EC-monthly-200901-201202-bar](http://gs.statcounter.com/#mobile_os-EC-monthly-200901-201202-bar)



### 3. CAPITULO III

#### 3.1 IDENTIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN MÁS COMUNES

Existe una gran variedad de lenguajes de programación para celulares, hemos identificado los más importantes y utilizados para el desarrollo de aplicaciones móviles. A continuación presentamos las principales plataformas:

- Java - J2ME
- Microsoft Compact .NET FrameWork (VB.NET o C#)
- Python- PyS60
- Objective-C
- WML y XHTML.

##### 3.1.1 JAVA 2 MICRO EDITION J2ME



La edición Java 2 Micro Edition fue presentada en 1999 por Sun Microsystems con el propósito de habilitar aplicaciones Java para dispositivos pequeños.

En Java podemos encontrar tres ediciones principales.

- J2SE (Java Standard Edition) orientada al desarrollo de aplicaciones independientes de la plataforma.
- J2EE (Java Enterprise Edition) orientada al entorno empresarial.
- J2ME (Java Micro Edition) orientada a dispositivos con capacidades restringidas.



### **Características de cada una de las versiones.**

#### **Java 2 Platform, Standard Edition (J2SE).-**

Es la base de las versiones de Java ya que es el inicio de este lenguaje, además es basado bajo C++, pero desarrollado con componentes de alto nivel, como soporte nativo de strings y recolector basura.

Posee código autónomo de la plataforma pre compilado a bytecodes y ejecutado en el cliente por una JVM (Java Virtual Machine).

También posee herramientas para desarrollar Applets, ya que las APIs son utilizadas en Interfaz gráfica de usuario, multimedia, redes de comunicación, etc.

#### **Java 2 Platform, Enterprise Edition (J2EE).-**

Esta versión está orientada al entorno empresarial, además posee características importantes: es considerado una de las principales ya que el objetivo es ejecutarse sobre una red de ordenadores de manera distribuida y remota mediante EJBs (Enterprise Java Beans). En sí, el software empresarial es capaz de realizar integración de datos provenientes de ambientes heterogéneos.

También esta versión se enfoca especialmente al desarrollo de servicios web, persistencia de objetos, XML, APIs para la gestión de transacciones, etc. La importancia de esta especificación es extender la J2SE para brindar soporte a los requisitos de las aplicaciones de empresa.

#### **Java 2 Platform, Micro Edition (J2ME).-**

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 versión tiene varios componentes básicos a diferencia de las otras, como el uso de una máquina virtual llamada KVM (Kilo Virtual Machine, ya que su utilización necesita una reducida capacidad de memoria para su funcionamiento) en vez del uso de la JVM clásica.



Figura 3.1: Arquitectura plataforma Java

Fuente: <http://es.scribd.com/doc/72213082/113/Figura-6-1-Arquitectura-de-la-Plataforma-Java-2-de-Sun>

### Relación entre las APIs de la plataforma Java

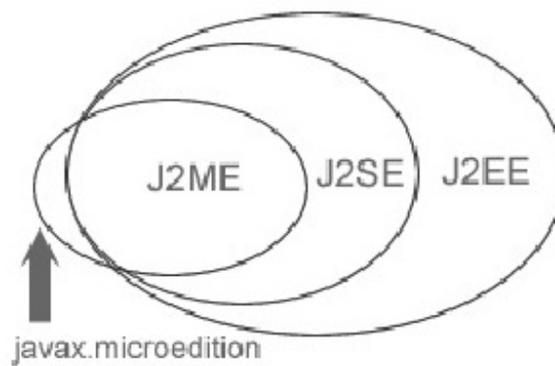


Figura 3.2: Ubicación de las tecnologías java

Fuente: <http://es.scribd.com/doc/72213082/113/Figura-6-1-Arquitectura-de-la-Plataforma-Java-2-de-Sun>.



De acuerdo a esta figura, J2ME constituye una versión reducida de J2SE, además esta versión fue separada ya que fue pensada para dispositivos móviles con capacidades limitadas e interfacesgráfica reducida. También separó J2SE de J2EE por motivos exigentes ya que cuenta con características muy pesadas o especializadas de E/S, trabajo en red, etc., por razones de eficiencia. J2EE es un súper conjunto de J2SE pues contiene toda la funcionalidad de éste y todas las características, así como J2ME es un subconjunto de J2SE (excepto por el paquete `javax.microedition`).

### **3.1.1.1. Ventajas**

- Para desarrollar, instalar y ejecutar aplicacionespara dispositivos móviles de capacidad limitada con acceso a redes de información.
- De acuerdo a sus configuraciones y perfiles se puede desarrollar varias aplicaciones orientadas a diversosdispositivos móvilesde acuerdo a las necesidades.
- Se utiliza software de libre distribución.
- La tecnología JAVA hoy en día es lenguaje universal y para una persona que conozca el lenguaje JAVA, el aprendizaje de la tecnología J2ME lo hace fácil.
- Java 2 Micro Edition posee un entorno de desarrollo para la creación deaplicaciones para dispositivosmóviles.
- Portabilidad de código. Significa que las aplicaciones se desarrollan una vez y al final se puede implementar cualquier dispositivo.
- Seguridad. Son aplicaciones auto contenidas ya que cuenta con restricciones al sistema.
- En la actualidad existe la comunidad de desarrolladores para aportar ideas o ya sea por necesidades del lenguaje.



### 3.1.1.2. Desventajas

- La tecnología J2ME presenta desventajas que no pueden ser controladas o no puede realizar funciones, como leer una guía telefónica, los sistemas de mensajería de contenido, etc.
- Teniendo en cuenta las limitaciones en el hardware del teléfono móvil, la velocidad de operación es relativamente lenta, pero la velocidad de los usuarios todavía puede aceptar.
- Además existe una limitación en su funcionalidad de acuerdo al tamaño, en general es de 64 KB en el J2ME.
- Limita el número de exception disponibles para control de errores.
- Excluye operaciones con punto flotante y los tipos de datos que manejan esa información.

### 3.1.1.3. Características

Java 2 Micro Edition, “J2ME” posee características específicas de este tipo de dispositivos, ya que los desarrolladores de Java vieron la obligación de desarrollar un subconjunto del lenguaje y configurar sus principales bibliotecas para permitir su adaptación a un entorno con limitaciones de memoria, velocidad de proceso y pantallas de reducidas dimensiones.

Máquina virtual (Kilobyte virtual Machine).- Encargada de interpretar código intermedio (bytecode) de los programas java pre compilados a código máquina ejecutable por la plataforma.

Configuración.- Conjuntos de APIs básicas de java que define un entorno generalizado de ejecución.

Existen 2 configuraciones definidas en J2ME:

- ConnectedLimitedDeviceConfiguration (CLDC) enfocada a dispositivos con restricciones de procesamiento y memoria.



- ConnectedDeviceConfiguration (CDC) enfocada a dispositivos con más recursos.

Perfil.- Conjuntos de APIs orientados a un ámbito de aplicación determinado, además cuenta con bibliotecas Java orientadas a realizar funcionalidades de más alto nivel para familias específicas de dispositivos.

#### 3.1.1.4. Arquitectura del entorno de ejecución de J2ME

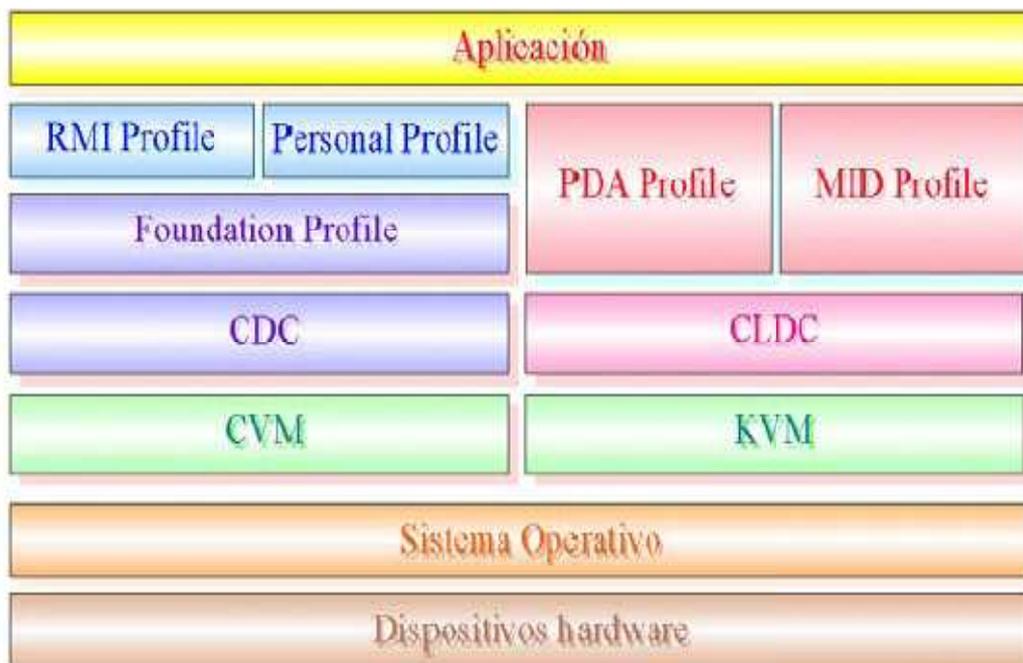


Figura 3.3: Arquitectura de J2ME

Fuente: <http://profesores.elo.utfsm.cl/~agv/elo323/2s06/projects/LoyolaCastillo/tecnologia.htm>



*Figura 3.4: Entorno de ejecución*

*Fuente: <http://es.scribd.com/doc/7136584/J2ME#>*

### **Máquinas Virtuales J2ME**

Una máquina virtual de Java (JVM) interpreta el código intermedio (bytecode) de los programas Java pre compilado a código máquina ejecutable por la plataforma, realiza las llamadas importantes al sistema operativo y además observa las reglas de seguridad y realiza corrección de código.

La VM (Virtual Machine) de la configuración CLDC se denomina KVM y la de la configuración CDC se denomina CVM Compact Virtual Machine.

A continuación nos centraremos en la configuración CLDC:

#### **3.1.1.4.1. KVM**

Es una Máquina Virtual pequeña desarrollada por Sun, su abreviatura es KVM proviene de Kilobyte haciendo referencia a la baja capacidad de memoria, entre 40Kb y 80Kb.

KVM está especialmente orientada a dispositivos móviles con capacidades limitadas computacionales y de memoria.

Sin embargo, estas limitaciones hacen que exista desventajas con respecto a la clásica Java Virtual Machine (JVM):



1. No hay soporte para tipos en coma flotante, esta limitación está presente porque los dispositivos carecen del hardware necesario para realizar operaciones difíciles.
2. Carece de soporte para JNI (Java Native Interface) ya que no cuenta con suficientes recursos de memoria.
3. No existe la finalización de instancias de clases. No existe el método `Object.finalize()`.
4. Con respecto al manejo de excepciones existe una limitación de control ya que depende de las APIs de cada dispositivo móvil por lo que son éstos los que controlan la mayoría de las excepciones.

#### **3.1.1.4.2. Configuraciones**

##### **Configuración de dispositivos limitados con conexión, CLDC (ConnectedLimitedDeviceConfiguration).**

A continuación se presentan características básicas, comunes a todos los dispositivos:

- CLDC está orientada a dispositivos móviles limitados con conexión y con limitaciones en pantalla gráfica y memoria.

Ejemplo de dispositivos móviles:

Teléfonos móviles, buscapersonas (pagers), PDAs, organizadores personales, etc., debido a sus restricciones se ejecutan bajo el uso de la KVM, mediante las configuraciones CLDC.

- Los dispositivos que utilizan CLDC deben cumplir los siguientes requisitos:
  - ✓ Disponer entre 160 Kb y 512 Kb de memoria total disponible, además como requerimiento mínimo es de 128 Kb de memoria no volátil para la Máquina Virtual.
  - ✓ Procesador de 16 o 32 bits con al menos 25 Mhz de velocidad.



- ✓ Consumo de batería, debido a que estos dispositivos trabajan con energía limitado, normalmente baterías, se ofrece un bajo consumo.
- ✓ Tener conexión a algún tipo de red inalámbrico, con conexión intermitente y ancho de banda limitado (unos 9600 bps).
- La CLDC aporta las siguientes funcionalidades a los dispositivos:
  - ✓ Un subconjunto del lenguaje Java (J2ME) 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.

### Librerías de Configuraciones CLDC

Nombre de paquete CLDC	Descripción
java.io	Clases y paquetes estándar de E/S. Subconjunto de J2SE.
java.lang	Clases e interfaces de la Máquina Virtual. Subconj. de J2SE.
java.util	Clases, interfaces y utilidades estándar. Subconj. de J2SE.
javax.microedition.io	Clases e interfaces de conexión genérica CLDC

Figura 3.5: Librerías de CLDC

Fuente: <http://es.scribd.com/doc/7136584/J2ME#>

Sin embargo, la Configuración no se encarga del mantenimiento del ciclo de vida de la aplicación, interfaces de usuario o manejo de eventos, sino que estas son manejadas por los **perfiles**.



### 3.1.1.4.3. Perfiles

Para la configuración CLDC tenemos los siguientes perfiles:

- PDA Profile.
- Mobile Information Device Profile (MIDP).

De igual manera nos centraremos en la configuración CLDC.

#### **PDA Profile:**

- Está construido sobre CLDC.
- Pretende abarcar PDAs de gama baja, tipo Palm, con una pantalla y algún tipo de puntero (ratón o lápiz) y una resolución de al menos 20000 pixels.

#### **Mobile Information Device Profile (MIDP):**

Este perfil está desarrollado sobre la configuración CLDC, además MIDP fue el primer perfil definido para esta plataforma.

Este perfil está enfocado a dispositivos con las siguientes características:

- Reducida capacidad computacional y de memoria.
- Conectividad limitada (en torno a 9600 bps).
- Capacidad gráfica muy reducida (mínimo un display de 96x54 pixels monocromo).
- Entrada de datos alfanumérica reducida.
- 128 Kb de memoria no volátil para componentes MIDP.
- 8 Kb de memoria no volátil para datos persistentes de aplicaciones.
- 32 Kb de memoria volátil en tiempo de ejecución para la pila Java.

Los tipos de dispositivos que se adaptan a estas características son: teléfonos móviles, buscapersonas (pagers) o PDAs de gama baja con conectividad.

En este apartado se define los perfiles que controla el ciclo de vida de la aplicación, interfaz de usuario, etc. más detalladamente, un perfil es un conjunto



de APIs orientado a un ámbito de aplicación determinado. Estas APIs reconocen la funcionalidad de los dispositivos móviles que proporcionan y el tipo de aplicaciones que se ejecutarán en ellos.

Las librerías de la interfaz gráfica son un componente importante en la definición de un perfil ya que cuenta con librerías que se diferencian entre las interfaces, el menú textual de los teléfonos móviles hasta los táctiles de los PDAs.

El perfil establece unas APIs que definen las características de un dispositivo, mientras que la configuración hace lo propio con una familia de ellos. Esto implica que al desarrollar una aplicación el perfil como la configuración cuenta con sus propias APIs.

Tenemos que tener en cuenta que un perfil siempre se construye sobre una configuración determinada.

### **Librerías de perfil MIDP**

El perfil MIDP establece capacidades del dispositivo, por lo tanto, especifica las APIs relacionadas con:

- La aplicación (semántica y control de la aplicación MIDP).
- Interfaz de usuario.
- Almacenamiento persistente.
- Trabajo en conexión.

Paquetes del MIDP	Descripción
javax.microedition.lcdui	Clases e interfaces para GUIs
javax.microedition.rms	<i>Record Management Storage</i> . Soporte para el almacenamiento persistente del dispositivo
javax.microedition.midlet	Clases de definición de la aplicación
javax.microedition.io	Clases e interfaces de conexión genérica
java.io	Clases e interfaces de E/S básica
java.lang	Clases e interfaces de la Máquina Virtual
java.util	Clases e interfaces de utilidades estándar

*Figura 3.6: Librerías de MIDP*

Fuente: <http://es.scribd.com/doc/7136584/J2ME#>



### 3.1.1.5. Plataformas

#### **Multiplataforma**

- Windows.
- Blackberry OS
- Android OS



### 3.2.1 MICROSOFT COMPACT .NET FRAMEWORK

Microsoft .NET es una plataforma de desarrollo y ejecución de aplicaciones, ya que posee componentes necesarios para el desarrollo de aplicaciones de software y todos los mecanismos de seguridad y eficiencia para asegurar su óptima ejecución.

MicroSoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años para competir con la plataforma Java.

Esta plataforma se ha desarrollado con los siguientes objetivos:

- Mejorar su modelo de componentes COM+.
- Obtener un entorno específico para el desarrollo y ejecución de la aplicación en forma de servicios.
- Para el desarrollo de aplicaciones en esta plataforma, Microsoft posee un conjunto de herramientas conocidas como .NET Framework SDK, que incluye compiladores de lenguajes como C#, Visual Basic.NET, Managed C++ y Jscript.NET específicamente diseñados para crear aplicaciones para él.

Entre los principales servicios encontramos:

- La programación lo realiza de una manera consistente y sencilla, completamente orientado a objetos.
- Eliminación del problema de compatibilidad entre DLL's.
- Aislamiento de memoria entre procesos y comprobaciones automáticas de seguridad de tipos en las conversiones.
- Gestión del acceso a objetos remotos.
- Seguridad avanzada.

El modelo de programación de Microsoft .Net se creó especialmente para desarrollar aplicaciones de tipo Web, cliente y servicios Web XML.



Las herramientas que incluye esta plataforma son:

- CommonLanguageRuntime (CLR): administra los servicios en tiempo de ejecución.

El corazón de la plataforma.NET es el CLR ya que administra la memoria, ejecución de subprocesos, ejecución de código, comprobación de la seguridad del código, compilación y demás servicios del sistema. Podríamos decir que esta es la máquina virtual de .NET.

Este componente es el encargado de controlar el ciclo de vida de cualquier aplicación .NET, le inicia, le detiene e interactúa o tiene una comunicación directa con el Sistema Operativo y provee servicios y recursos en tiempo de ejecución.

- Librerías de clases: Estas bibliotecas proveen funciones estándar de entradas y salidas, manipulación de cadenas, gestión de la seguridad, comunicaciones en redes e hilos, etc.
- También agregan acceso a bases de datos con ADO.Net y páginas Web dinámicas y Servicios Web con ASP.Net

En esta plataforma presenta un conjunto de lenguajes de programación de alto nivel, junto con sus compiladores simplificando las tareas más comunes en el desarrollo de aplicaciones, (VB, C++, C#,20+).

Las características principales de la plataforma Microsoft .NET:

- Es una plataforma de ejecución intermedia, ya que no realiza directamente con el sistema operativo, como se presentaba en las versiones anteriores o el modelo tradicional de desarrollo. En su lugar, las aplicaciones .NET son ejecutadas bajo un componente de software llamado Entorno de Ejecución (“Runtime” o “Máquina Virtual”).
- La plataforma Microsoft .NET está basado en el modelo de programación orientado a Objetos.
- La plataforma Microsoft .NETes multilenguaje,consiste en escribir o desarrollar varias aplicaciones bajo esta plataforma sin la necesidad de saber un único



lenguaje específico de programación de alto nivel, sino que se puede elegir de una amplia gama de opciones.

- La plataforma Microsoft .NET fue diseñado para realizar un único modelo de programación, en la cual se desarrolla para todo tipo de aplicaciones ya sean de formularios Windows, de consola, aplicaciones Web, aplicaciones móviles, etc. y considerando a qué tipo de dispositivo se va a ejecutar dichas aplicaciones ya sean PC's, Pocket PC's, Teléfonos Celulares Inteligentes o "SmartPhones", Tablet PC's, etc...

Ventajas de desarrollar con Microsoft .NET:

- Es más eficiente que las anteriores plataformas de Microsoft
- La instalación de aplicaciones sobre plataforma .NET es más sencillo.
- La migración desde Visual Basic a Visual Basic.NET, aunque no es inmediata, es factible.
- Microsoft ha desarrollado herramientas 'visuales' de desarrollo (Visual C# y Visual Studio .Net)
- Simplifica aún más el desarrollo de nuevas aplicaciones.
- Entrega un entorno de ejecución robusto y seguro.
- Es independiente del lenguaje de programación.
- Interoperabilidad con código existente.
- Simplifica la administración de las aplicaciones.



## **Microsoft .NETCompact Framework**

Desarrollada por Microsoft para el desarrollo de aplicaciones móviles tales como PDAs, teléfonos móviles, además permite reutilizar conocimientos y experiencia de desarrollo sin necesidad de aprender nuevos lenguajes.

Es un subconjunto de .NET Framework orientado a dispositivos móviles, muchas de las funcionalidades de .NET se ven disminuidas en esta versión debida a temas de rendimiento y tamaño del CommonLanguageRuntime.

### **3.2.1.1. Ventajas .NET Compact Framework**

- Acceso a las bibliotecas que se encuentran en el NET Compact Framework.
- Entorno de desarrollo muy bueno. Visual studio .NET
  - ✓ Smart DeviceExtensions (SDE). Tiene el objetivo de dar soporte a los desarrolladores que utilizan el Compact Framework y es la parte que permite generar aplicaciones móviles de forma similar al desarrollo de aplicaciones de Windows de escritorio. Cuando el programador desea realizar pruebas de sus aplicaciones existe la posibilidad de ejecutarlas sobre el emulador o sobre el dispositivo real.
- MMIT (Microsoft Mobile Internet Toolkit)
  - ✓ El MMIT es otro complemento a Smart DeviceExtensions mediante esta tecnología los desarrolladores pueden construir una misma aplicación web móvil para una amplia variedad de dispositivos. Esto quiere decir que la misma aplicación se adapta automáticamente dependiendo del dispositivo.
  - ✓ La forma en que funciona MMIT es: el código de la aplicación reside en un servidor y a su vez el MMIT posee una serie de controles ASP .NET que también se encuentran en el servidor. Estos últimos son



encargados de presentar la página web utilizando WML, HTML o XHTML según el tipo de dispositivo que quieran acceder la aplicación. Por último, el dispositivo visualiza la aplicación mediante un web browser.

- Podemos desarrollar ya sea en VB.NET o C#
- Posee sus propios emuladores.
- Es muy fácil la programación. (Arrastrar y colocar).
- Se encuentra software de libre distribución: VB.NET Express.
- Tiene su propio motor de base de datos: Compact SQLServer.NET.

### **3.2.1.2. Características .NET Compact Framework**

- Clases: .NET Compact Framework acepta un subconjunto de la Biblioteca de clases de .NET Framework, ya que es apropiado para desarrollar aplicaciones para dispositivos con limitaciones de recursos.
- Datos: Esta plataforma acepta un subconjunto de ADO.NET e incluye el proveedor de datos de SQL Server CE .NET. No está soportado ODBC.
- Entrada / Salida: Entre los sistemas operativos, existen restricciones y limitaciones en el modelo de E/S, además esta plataforma no proporciona notificaciones de cambios en los archivos. Dado que las entradas y salidas en los dispositivos ocurre en la RAM, en la cual es imposible definir y obtener acceso a los atributos de archivos y directorios.
- Memoria: .NET Compact Framework está optimizado para sistemas alimentados por baterías y evita el uso intensivo de ciclos de RAM y de CPU.
- Redes: .NET Compact Framework proporciona clases IrDA (Infrared Data Association) para establecer conexiones por infrarrojos y clases de escucha de Web para atender peticiones de HTTP al dispositivo.



- Formularios Windows Forms y gráficos: Estos controles de formularios están diseñados especialmente para .NET Compact Framework.

### 3.2.1.3. Descripción del lenguaje C#

Lenguajes de desarrollo de aplicaciones .NET Compact Framework

- C# es el lenguaje nativo de .NET y carece de elementos heredados innecesarios.
- La sintaxis y estructuras de C# es muy parecida al lenguaje de C++ o Java, por lo tanto es brindar una facilitación a la hora de la migración de código y su aprendizaje a los desarrolladores familiarizados a ellos. Además, su sencillez y el alto nivel de productividad son comparables a los de Visual Basic.
- C# es un lenguaje de Microsoft introducido en la plataforma .Net considerando las mejores características de lenguajes existentes como Visual Basic, Java o C++ y las combina en uno solo.

**Ventajas de C# son:**

- Realiza una gestión de memoria automática, aunque permite utilizar punteros y además gestiona manualmente la memoria dentro de un bloque de código convenientemente marcado.
- Se pueden escribir comentarios en XML para generar luego documentación automática.
- Permite herencia simple y definición de interfaces.
- Dispone de soporte nativo para **COM** y otros API's de Microsoft.



### **Características de C#**

- **Sencillez:** C# Simplifica su desarrollo eliminando componentes innecesarios en .NET. por lo tanto el código escrito en C# es autocontenido, ya que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL.
- **Independencia de Tipos:** El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad de código.
- **Modernidad:** C# incorpora elementos necesarios y útiles para el desarrollo de aplicaciones, como por ejemplo un tipo básico decimal que permita efectuar operaciones de alta precisión con reales de 128 bits, la inclusión de una instrucción Foreach que permita recorrer colecciones o Arrays con facilidad y ampliable a tipos definidos por el usuario, la inclusión de un tipo básico String para representar cadenas o la inclusión de un tipo Bool para representar valores lógicos.
- **Orientación a Objetos:** C# es un lenguaje con orientación a objetos y además soporta características propias de este paradigma. C# es un lenguaje más puro ya que no permite funciones ni variables globales sino que todo el código y datos se escribe dentro de definiciones de tipos de datos, lo que implica reducción de problemas por conflictos de nombres y ayuda a la legibilidad del código.
- **Encapsulación, herencia y polimorfismo.** En este apartado este lenguaje cuenta con los modificadores existentes public, private y protected, además este lenguaje cuenta con un modificador adicional llamado internal, que se puede combinar con protected, por lo tanto indica que al elemento a su correspondiente definición precede sólo puede accederse desde su mismo ensamblado.
- **Seguridad de tipos:** C# detecta los acceso a los tipos de datos mediante mecanismos para asegurar su correcto funcionamiento, lo que evita que se



produzcan fallas o problemas que sean imposibles de detectar por acceso a memoria no perteneciente a ningún objeto ya que es necesario en un entorno gestionado por un recolector de basura.

#### **3.2.1.4. Plataformas**

- Plataforma soportada solo para el sistema operativo Microsoft Windows Mobile y Microsoft Pocket PC.



### 3.3.1 PYTHON (PYS60)



Creado por Guido Van Rossum en el año 1990. Este lenguaje está modelado a la programación orientada a objetos, su desarrollo y mantención lo realiza la comunidad global de desarrolladores de código abierto bajo la responsabilidad de la Python Software Foundation.

Se puede ejecutar en varias plataformas Windows, Linux/Unix, Mac OS X, OS/2, además son ejecutados en varios dispositivos como Palm Handhelds, y teléfonos celulares Nokia.

Esta plataforma se desarrolla sobre scripts, competencia directa con Perl. Python permite realizar sub módulos para posteriormente ser reutilizables desde otros programas Python, lo que permite ahorrar el proceso de compilado.

También viene con una gran colección de módulos estándar que proporcionan E/S de ficheros, llamadas al sistema, sockets, interfaces GUI, etc.

Python tiene la capacidad de acceder a la API de C++ permitiendo el desarrollo de prototipos y pruebas de conceptos que de otra manera resultarían más costosos.

#### **La plataforma S60 (serie 60)**

S60 es una plataforma para terminales móviles (teléfonos inteligentes o PDAs) que utilizan el sistema operativo Symbian. S60 se encuentra actualmente entre las plataformas líderes de terminales del tipo Smartphone en el mundo.

S60 tiene una gran variedad de bibliotecas y APIs, tales como telefonía, herramientas de gestión de información personal, y reproductores multimedia, ya que fue desarrollada para dispositivos móviles modernos de amplias características, con pantallas a color, son conocidos Smartphone.

Nokia anunció el 11 de febrero del 2005 el lanzamiento de Python para la Plataforma Serie 60 llamada como Py60,



## **PyS60**

Para los dispositivos móviles existe una versión de Python modificada, ajustada para la Serie S60 del sistema operativo Symbian(PyS60).

PyS60 está basado en Python 2.2.2. Este apoya a módulos estándares de la biblioteca de Python, dentro de estas bibliotecas existen módulos específicos:

- Widgets nativos del GUI
- Bluetooth
- Calendario, grabación de sonidos y jugar con el audio, contactos.
- Establecimiento de una red de GPRS
- Mensajería de SMS,
- Acceso a la cámara fotográfica, etc.

### **3.3.1.1. Ventajas**

- Portabilidad.
- Versatilidad.
- Simplicidad.
- Interactividad.
- Sintaxis clara y legible.
- Productividad.
- Popularidad.
- Facilidad y rapidez de aprendizaje.
- Open Source.



### 3.3.1.2. Características

- Simple y sencillo: Es en lenguaje simple, el pseudo-código natural de Python es una de sus ventajas importantes ya que permite concentrarse en la solución del problema en lugar de la sintaxis, es decir el propio lenguaje, también es sencillo de aprender a programar ya que ofrece una codificación simple.
- Libre y Fuente Abierta: Es un FLOSS (Free/Libre and Open Source), su principal concepto es ser una comunidad que comparte conocimiento.
- Lenguaje de Alto Nivel: Cuando se programa en Python no se preocupa por detalles de bajo nivel, como manejar la memoria empleada por el programa.
- Portable: Como este lenguaje es Open Source, su código es portable, es decir, funciona en diversas plataformas, como Linux, Windows, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE y PocketPC.
- Interpretado: Esta característica similar a la de Java, ya que esta plataforma Python convierte el código fuente en una forma intermedia llamada bytecode, después los traduce en el lenguaje nativo de la computadora y procede a la ejecución de la aplicación.
- Orientado a Objetos: Este lenguaje admite la programación orientada a procedimientos incluyendo la programación orientada a objetos, es decir, el programa está construido sobre procedimientos o funciones los cuales son programas reutilizables.

### 3.3.1.3. Plataformas

- Python para la Serie S60 (PyS60) es una versión modificada del Python adaptada para la Serie S60 del sistema operativo Symbian usado en dispositivos móviles.



### 3.4.1 OBJECTIVE-C



Es un lenguaje orientado a objetos desarrollado como un súperconjunto de C con un estilo muy parecido al de Smalltalk. Es bajo licencia GNU para el compilador gcc. Actualmente se usa como lenguaje principal de programación en Mac OS X y GNUStep.

Podemos obtener un compilador de Objective-C en:

- Mac OS X se tiene el gcc de Apple.
- En otro SO podemos conseguir el compilador gcc de GNU.

#### 3.4.1.1. Ventajas

- Convierte sus programas en software muy ligero óptimo para dispositivos móviles.
- Compila directamente para que el código se ejecute contra la máquina, lo que le da una versatilidad y una eficiencia muy elevadas.
- Es un lenguaje orientado a objetos que extiende el lenguaje estándar ANSI C.
- La sintaxis básica así como el diseño están basados en Small Talk, uno de los primeros lenguajes orientados a objetos.

#### 3.4.1.2. Características

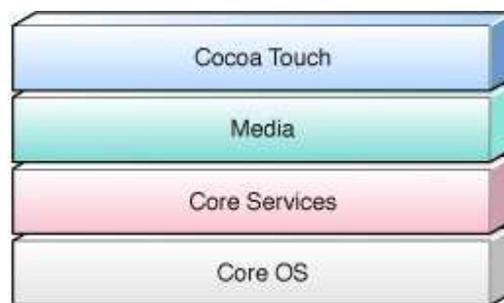
- Sentencias de control de flujo.
- Los tipos de datos fundamentales, estructuras y punteros.
- Conversiones implícitas y explícitas entre tipos.
- Los ámbitos de las variables: Globales, estáticas y locales.
- Las funciones y su sintaxis.
- Las directivas del preprocesador, aunque veremos que Objective-C añade más directivas del preprocesador y también añade las llamadas directivas del compilador.



- Provee sintaxis para la definición de clases, métodos, propiedades, protocolos e interfaces.
- Xcode es el IDE que proporciona Apple para poder desarrollar aplicaciones de IOS para empezar a programar con este lenguaje de programación.
- CocoaTouch crea aplicaciones no solo para el iphone o ipod, sino para toda la familia Mac OS.

### 3.4.1.3. Arquitectura.

Capas con la que trabaja iphone o ipod.



*Figura 3.7: Arquitectura Objective-C*

*Fuente: <http://www.cristalab.com/tutoriales/fundamentos-de-programacion-para-iphone-c260/>*

- Core OS: Es el núcleo del sistema operativo y responsable de gestionar los archivos, la memoria, seguridad y la comunicación.
- CoreServices: Contiene los servicios fundamentales de la aplicación, unos de estos servicios acceder a la lista de contactos, al SQLite o a las preferencias.
- Media: Hace referencia a Multimedia: video, audio, archivos de imágenes, Coreanimation, OPENGL ES, etc.
- CocoaTouch: Es Framework de desarrollo para el Iphone, en la cual presta varias bibliotecas o APIs para el desarrollo de aplicaciones ya que en realidad es el más importante.



**Posee 3 frameworks:**

- ✓ UIKit, proporciona todos los elementos gráficos para nuestra aplicación mediante interfaces a su framework, la estructura de la aplicación, el control de los eventos, el manejo de la interfaz, la representación de las vistas y controles así como el soporte de texto y contenido web.
- ✓ Foundation Framework, es el conjunto de clases que utiliza para representar estructuras de datos complejas. Define el acceso y manejo de objetos, provee acceso a los tipos de datos primitivos, colecciones y servicios del sistema operativo.
- ✓ Address Book UI Framework, ayuda a acceder a los datos de nuestros contactos.

**3.4.1.4. Plataformas**

- La plataforma iPhone OS usa el lenguaje Objective-C



### 3.5.1 WML Y XHTML

Los lenguajes utilizados para el desarrollo de aplicaciones Wap son el WML y el XHTML.

El protocolo WAP fue desarrollado por el Forum WAP, grupo formado por Nokia, Ericsson, Motorola y Phone.com.

Es una tecnología considerada como un estándar mundial orientado al desarrollo de aplicaciones, presentación y envío de información para Internet Móvil a través de los dispositivos móviles.

De esta manera, un sitio web puede ser visto usando un teléfono celular habilitado para WAP (WirelessApplicationProtocol) debido a que es un protocolo para aplicaciones móviles.

Esta plataforma ofrece a los usuarios las mismas funcionalidades que se ofrecen en la Web.

#### **Ventajas**

- WAP presenta un estándar de licencia libre a la industria inalámbrica ya que es posible crear aplicaciones basados en WAP para cualquier dispositivo móvil.
- Este protocolo no depende de ningún tipo de comunicación en la red, esto puede ser implementado en redes CDMA, GSM, TDMA, IDEN, FLEX, TETRA, DECT, DATA, TACT, entre otras.

#### **Desventajas**

- La baja transmisión: actualmente entre 14.4 kbps y 19kbps.
- Los altos costos de las comunicaciones: Dependiendo de los servicios de los proveedores.
- Herramientas limitadas de navegación.
- La dificultad de cargar contenidos HTML. El teléfono celular está pensado para comunicaciones cortas y urgentes.



## Componentes

Wap une dos tecnologías actuales, el internet y las comunicaciones móviles, este estándar está conformado por diversos componentes:

- WAE (WirelessApplicationEnvironment)

El objetivo principal es construir un entorno de aplicación basado en el mismo estándar de la web.

WAE, tiene incluido el navegador wap, este está integrado por el WML (WirelessMarkupLanguage) y el WML Script.

WML es el lenguaje que equivale a HTML utilizando el protocolo WAP, la diferencia está en que WML posee un menor número de tags.

También es un lenguaje para el desarrollo de aplicaciones para dispositivos móviles con capacidades limitadas, pantallas pequeñas y que tienen una conexión a red con un ancho de banda mínimo.

WML Script es un lenguaje script similar a Visual Basic Script o JavaScript el cual se ejecuta en el cliente para su uso con WML y además permite hacer páginas dinámicas.

WBMP (WírelessBitMaP) que es el formato predeterminado de imágenes para WAP, se generan imágenes muy pequeñas cuyas dimensiones son de 96 x 48 pixeles.

- La capa WSP (Wireless sesión protocol) facilita a la aplicación una interfaz compuesta de dos servicios.
  - ✓ Trabaja en la conexión sobre la capa de transacción WTP.
  - ✓ Ésta no está orientada a la conexión que funciona por encima de la capa de transporte (WDP).
- La capa de transacción WTP (Wirelesstransactionprotocol) cuenta con los servicios de trasportación de datos, la seguridad de obtiene a través del uso de identificadores únicos de transacciones.
- WAP también tiene un protocolo perteneciente a la capa de seguridad el cual es



conocido como WTLS (Wireless Transport Layer Security) ya que provee privacidad, integridad de datos y autenticación a aplicaciones de teléfonos celulares y otras terminales inalámbricas.

- Capa de transporte WDP (wirelessdatagramprotocol) permite la comunicación de forma transparente con los protocolos válidos, además facilita una interfaz común a los protocolos de capas superiores.
- Estas capas han sido implementados para operar principalmente en un ancho de banda delimitado.
- Además una de las ventajas permite utilizar en dispositivos con capacidades de memoria y procesamiento muy limitadas.

### 3.5.1.1. Características

#### **WML (Wireless Mark-up language)**

- Definen sintaxis, variables, además se ha ido liberando y estandarizando, WML al igual que HTML usan tags, posee características estandarizadas como el uso de formularios, presentación de textos e imágenes, permite manipular datos y proporcionar interactividad con las páginas mediante un lenguaje de script, WML Script.
- El WML permite adaptarse a las limitaciones de los dispositivos, las dimensiones de las pantallas, la navegación sin teclado y ancho de banda reducida.
- A la hora de desarrollar se debe tener presente los elementos que se van a utilizar ya que son sensibles a las mayúsculas y minúsculas (etiquetas, atributos, identificadores, variables).
- Se define un par de etiquetas que contienen el cuerpo del documento.  
<wml> y </wml>



### **WMLScript**

- Está basado en JavaScript, permite ejecutar funciones de lado del cliente es decir sin tener que procesar información al servidor, logrando optimizar los tiempos de respuesta.
- WMLScript cubre las necesidades del navegador, colabora con el manejo de los datos, la realización de cálculos básicos y diferentes tipos de validación de campos de formularios. Cuando se necesita WMLScript se convoca desde página WML.

### **XHTML (eXtensibleHyperTextMarkupLanguage)**

- Es un lenguaje de programación estándar basado en HTML y tiene sus mismas funcionalidades, pero cumple con especificaciones más estrictas basadas en regla XML.
- Con este lenguaje se logra la integración entre el Wap móvil y acceso tradicional a internet, además ofrecen a los navegadores Wap la utilización de estilo (CSS), reutilización de contenidos y permite acceder desde el móvil a los diferentes servicios que se presenta en la web.

#### **3.5.1.2. Proceso para desarrollar una aplicación para WAP**

Para realizar una aplicación WAP, es lo mismo que hacer una página Web, aunque con variaciones mínimas.

#### **Requerimientos para desarrollar:**

- Editor de páginas WML o XHTML MP
  - ✓ WAPtor
  - ✓ Bloc de notas
  - ✓ Developer toolkit



- Emulador WAP
  - ✓ WAP Proof
  - ✓ WinWAP
- Servidor Web
  - ✓ Apache

### **Desarrollo:**

Después del desarrollo de una aplicación es enviado a un servidor una vez hecho esto, el usuario puede acceder desde un dispositivo móvil con soporte WAP, cada teléfono móvil posee su propia configuración para poder insertar la dirección de la página WAP.

De igual manera que en HTML, WAP necesita una configuración del servidor, a continuación se ha plasmado la siguiente pregunta:

¿Existe algún servidor especial que soporte las páginas WAP?

Simplemente configurar un servidor normal que sirva para páginas de Internet, la diferencia que se debe configurar los MIME Types.

### **MIME Types.**

Al navegar por la web usando un explorador de un PC, el servidor se comunica con el navegador diciéndole que tipo de documento está sirviendo, función que se realiza usando los caracteres MIME. Si no se configura los MIME Types característicos del WAP, el servidor no podrá decir al navegador nada sobre el tipo de documento.

#### ✓ **MIME type Extensión**

Para el código WML text/vnd.wap.wml .wml

Para las imágenes wml (extensión wbmp) image/vnd.wap.wbmp  
.wbmp

Para el WML Script text/vnd.wap.wmlscript .wmls

Para el WML compilado text/vnd.wap.wmlc .wmlc

Para el WML Script compilado text/vnd.wap.wmlscriptc .wmlsc



## 4. CAPITULO IV

### 4.1 PROTOTIPO

#### 4.1.1. REQUERIMIENTOS DE SOFTWARE Y HARDWARE

Las aplicaciones móviles son muy importantes ya que permite disminuir costos y optimizar la eficiencia de los servicios de las empresas u organizaciones, en la actualidad los sistemas empresariales deben acoplarse a nuevas herramientas para llevar a cabo en cualquier lugar operaciones que antes sólo se podían realizar desde la empresa ya que la comunicación móvil se ha convertido en parte fundamental de la competitividad.

Entre los beneficios que aporta la comunicación móvil se encuentran los siguientes puntos:

- Se puede interactuar en tiempo real con la empresa sin la necesidad de estar dentro de ella, para realizar operaciones o transacciones.
- Comunicaciones a través de la red de telefonía móvil.
- Una gestión automatizada más eficiente de la información y de ciertos procesos corporativos.
- Mejorar la calidad del servicio al cliente.
- Facilitar el control de los procesos, el personal y la organización.

De acuerdo a las necesidades de cada empresa podemos implementar aplicaciones móviles que sean compatibles con los sistemas de información existentes.

##### 4.1.1.1. Selección lenguaje de programación

Debido a que existen varios lenguajes de programación para dispositivos móviles debemos tener presente al desarrollar software, ¿qué lenguaje es el mejor?, y no tenemos una respuesta absoluta a ese interrogante, ya que elegir un lenguaje de programación depende de distintos factores como es el tipo de programa que queremos realizar, la plataforma para la cual queremos que sirvan nuestros programas, incluso va por la preferencia de un lenguaje específico.

De acuerdo a este interrogante hemos seleccionado nuestro lenguaje de programación para móviles *Java-J2ME* ya que cumple con nuestros requerimientos.



- Está enfocado a cualquier tipo de dispositivo con capacidad limitada.
- A través de la combinación de sus dos tipos de configuración con los diversos perfiles puede formar una extensa gama de aplicaciones orientadas a diversos dispositivos.
- La tecnología JAVA entro rápidamente como un lenguaje universal y el aprendizaje de la tecnología J2ME no es difícil.
- Portabilidad de código.
- Seguridad, ya que son aplicaciones auto contenidas, no tiene acceso al sistema.
- Las configuraciones ofrece características para hardware limitadas de los celulares, compatible con la mayoría de sistemas operativos móviles.
- Además utiliza API para el envío y recepción de mensajes cortos SMS.

#### 4.1.1.2. Selección de gateway sms

La comunicación SMS no se da solo entre dispositivos móviles también se obtiene un servicio de utilización para comunicar con ordenadores que respondan o envíen mensajes automáticos o manualmente. Durante la investigación se encontró varias opciones de pasarelas o gateway ya sean Open Source o Propietario.

Las opciones o pasarelas encontradas son:

##### **Kannel**

El licenciamiento que tiene esta solución es Software Libre de tipo BSD Kannel es una implementación completa de WAP y SMS compatible además con la mayoría de las centrales de servicios de mensajes cortos (SMSC).

##### **Alamin**

Software libre bajo licencia GPL que permite intercambiar mensajes entre redes GSM e IP, está desarrollado bajo Perl.

##### **Provato.**

Es software propietario. Provato es una pasarela de mensajes cortos (SMS) así como también de mensajes multimedia (MMS) que permite el despliegue y la



conexión de aplicaciones locales y de terceros a la red de mensajería móvil por medio de Web Services, SOAP, XML y JMS

### **Gnokii.**

Es una implementación de Nokia Data Suite para Linux y FreeBSD que requiere teléfono móvil compatible, habitualmente Nokia, aunque recientemente han sido soportados modelos de otros fabricantes, permite tanto envío como recepción de mensajes.

Minuciosamente se seleccionó la “**Pasarela Kannel**” para nuestro proyecto de Tesis, ya que cumple con nuestras expectativas, por las siguientes razones.

- ✓ Es software libre
- ✓ Servicio de mensajería sms (envía y recibe sms)
- ✓ Seguridad
- ✓ La SMSC Virtual puede ser un módem GSM, o teléfono móvil con módem interno, trabajando como una SMSC.
- ✓ Compatibilidad con varias bases de datos.

Posteriormente detallaremos las características, funciones, las conexiones, etc, del Gateway seleccionado.

Esta pasarela trabaja como Push Proxy Gateway (PPG), este último servicio envía datos a los dispositivos móviles.

Kannel presenta una arquitectura cliente/servidor, donde el núcleo es el bearerbox ya que es considerado servidor del gateway, su función principal es realizar conexiones con los SMSC's y realizar el ruteo entre las conexiones y los servicios.

El BearerBox tiene una comunicación directa con los componentes, "SMSBox" y "WAPBox", que se encargan de acceder a los servidores HTTP que tienen el contenido que se desea acceder. En nuestro caso utilizaremos smsbox.

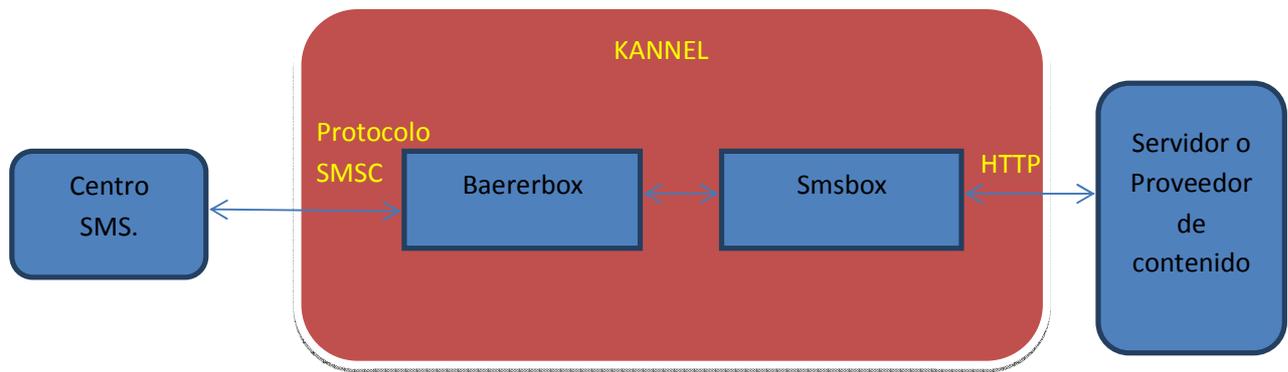


Figura 4.1: Arquitectura del Gateway.

Fuente: El Autor.

Nos enfocaremos principalmente en la caja smsbox, principalmente en los servicios que gestiona, en la cual un dispositivo móvil envía el mensaje y es recibido por el SMSC, enseguida es comunicado con el bearerbox, entonces el mensaje es encaminado al smsbox que accede a una página web y retorna un resultado para ser enviado al teléfono móvil.

Los servicios de SMSC son los servicios de contenidos iniciados por mensaje SMS a determinado número de teléfono, que a su vez responde con el contenido solicitado si está disponible.

La característica de Kannel es abstracta, cada protocolo SMSC a un protocolo bien conocido HTTP simplifica el despliegue de servicios.

Los servicios que se gestionan por el smsbox por el hecho de ser HTTP, tienen la desventaja de no permitir transaccionalidad impidiendo conocer las transacciones que iniciaron y terminaron exitosamente, otra desventaja presente es el no soporte de persistencia por lo que en caso de falla del sistema se pueden llegar a perder mensajes, ya que no implementa un método de recuperación o reenvío.



Funcionalidades y características generales:

- Implementa Push por medio de las distintas configuraciones y servicios que permite integrar.
- Dispone de reportes de las entregas de mensajes.
- Puede conectar varias fuentes de entrada y salida de mensajes (conexiones directas a un SMSC, módems GSM y/o terminales GSM).
- Permite conectar e integrar varios servicios basados en HTTP.

#### **4.1.1.3. Selección de Base de datos**

De acuerdo a los requerimientos hemos seleccionado la base de datos Mysql, esta es compatible con kannel Gateway.

- Almacenamiento de los mensajes enviados y recibidos.
- Sistema de bases de datos relacional multiusuario.
- En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos.
- Conexión directa desde el smsbox para el envío de mensajes.
- Mediante dlr se puede ver si se ha tenido éxito el mensaje.

#### **4.1.1.4. Requerimientos de Hardware**

- Computador para que se ejecute el Kannel, los requerimientos mínimos que se deben tener en cuenta son un PC Pentium II de 400 MHz y con un mínimo de 128 MB de memoria RAM.
- Computador para que se ejecute el servidor web, los requerimientos mínimos que se deben tener en cuenta son: mínimo 2 GB de RAM, procesador Intel Dual Core.



- Una central de servicio de mensajes cortos (SMSC). Si no se tiene acceso a una SMSC, se puede utilizar una SMSC Virtual.
- La SMSC Virtual puede ser un módem GSM, o teléfono móvil con módem interno, trabajando como una SMSC. Para nuestro proyecto utilizaremos el Modem E173 de movistar.
- Teléfonos móviles para instalar aplicación J2ME.

## 4.1.2. INTERFAZ DE USUARIO

### 4.1.2.1. Arquitectura de la pasarela SMS.

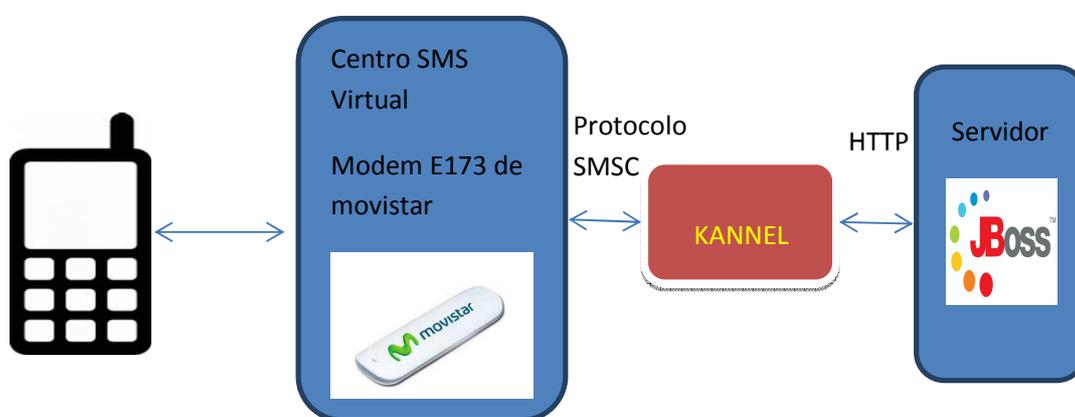


Figura 4.2: Posición lógica de la pasarela SMS entre un teléfono y un servidor de contenido.

Fuente: El Autor.

Cuando se utiliza los servicios de SMS, el cliente (terminal móvil) envía un mensaje SMS a un número determinado, que apunta al centro de mensajes SMS específico responsable de ese número. Este centro de mensajes SMS envía el mensaje a partir de un protocolo de SMS centro específico. Por ejemplo, un Nokia del centro de SMS utiliza el protocolo de CIMD.



El modelo anterior se divide en tres partes:

- Servidor Web, Base de Datos y Aplicación Web
- Gateway SMS.
- Aplicación Móvil.

A continuación detallaremos sus funciones, conexiones, etc.

#### **4.1.2.2. Interfaces, Configuraciones y conexiones.**

##### **4.1.2.2.1. Servidor Web, Modelo de la Base de Datos y Aplicación Web.**

###### **Descripción del Servidor Web.**

Jboss es implementado en java, consiste en un servidor de aplicaciones J2EE de código abierto, además es multiplataforma, fue desarrollado para aplicaciones de e-business de alto rendimiento. Permite a las empresas desarrollar, desplegar, integrar, organizar y presentar aplicaciones y servicios web en una arquitectura orientada a los servicios mediante un servidor de aplicaciones Open Source implementado en java puro.

Para la implementación de nuestro proyecto se ha configurado este servidor de aplicaciones, las ventajas principales de este servidor son las siguientes:

- JBoss Hibernate para el mapeo y persistencia de objetos/datos relacionales.
- JBoss Seam para simplificar el desarrollo de aplicaciones Web 2.0.
- JBoss jBPM para permitir la creación y automatización de procesos de negocio.
- JBoss Rules para el fácil acceso, modificación y gestión de políticas comerciales.
- Amplio servicio técnico.



- Aumento de la productividad de la empresa.
- Disminución de costos total de la propiedad.
- Acceso al producto: Código fuente y binario, documentación.
- Revisiones y acceso a las últimas correcciones de fallas y errores de seguridad.
- Flexibilidad: Las suscripciones no son específicas para una versión, por lo tanto incluyen el acceso a versiones anteriores y el soporte correspondiente.

Para desarrollar la aplicación web utilizaremos esta herramienta ya que nos facilitara la implementación, además utilizaremos persistencias y la conexión directa con la base de datos MYSQL.

En este apartado mostraremos el modelo da la base de datos, principalmente las entidades con sus respectivos atributos y sus relaciones que se utilizara en el sistema de mensajería.



Case Method o Modelo de la Base de Datos.

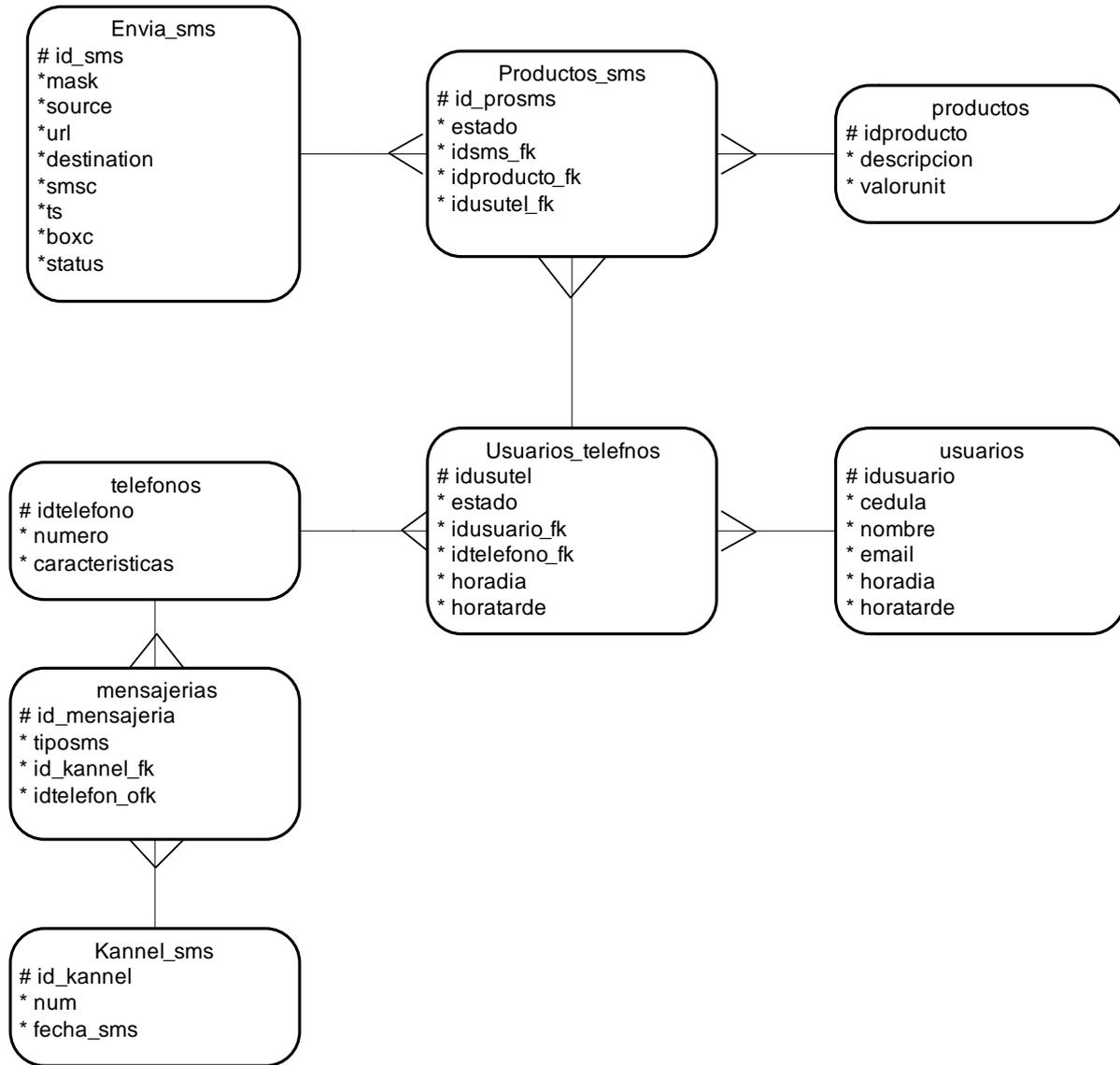


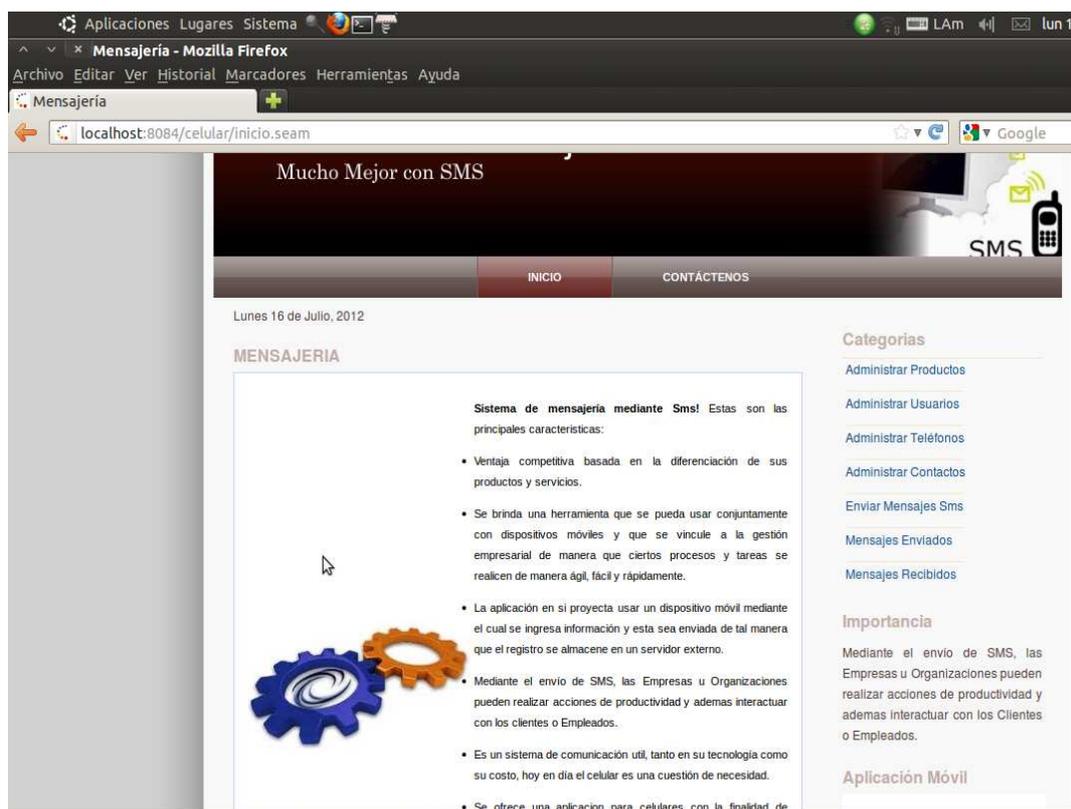
Figura 4.3: Case Method

Fuente: El Autor



## Descripción de la Aplicación web.

En este apartado detallaremos como esta implementado la parte del envío y recepción de mensajes SMS desde nuestra aplicación web, además de los usuarios, teléfonos y productos importantes.



Nuestra aplicación web consiste en un sistema de mensajería empresarial, en la cual tiene como objetivo gestionar el envío y recepción de mensajes cortos SMS, en la página inicio se presenta una breve descripción de las características principales de este sistema.

Esta aplicación se divide en siete categorías:

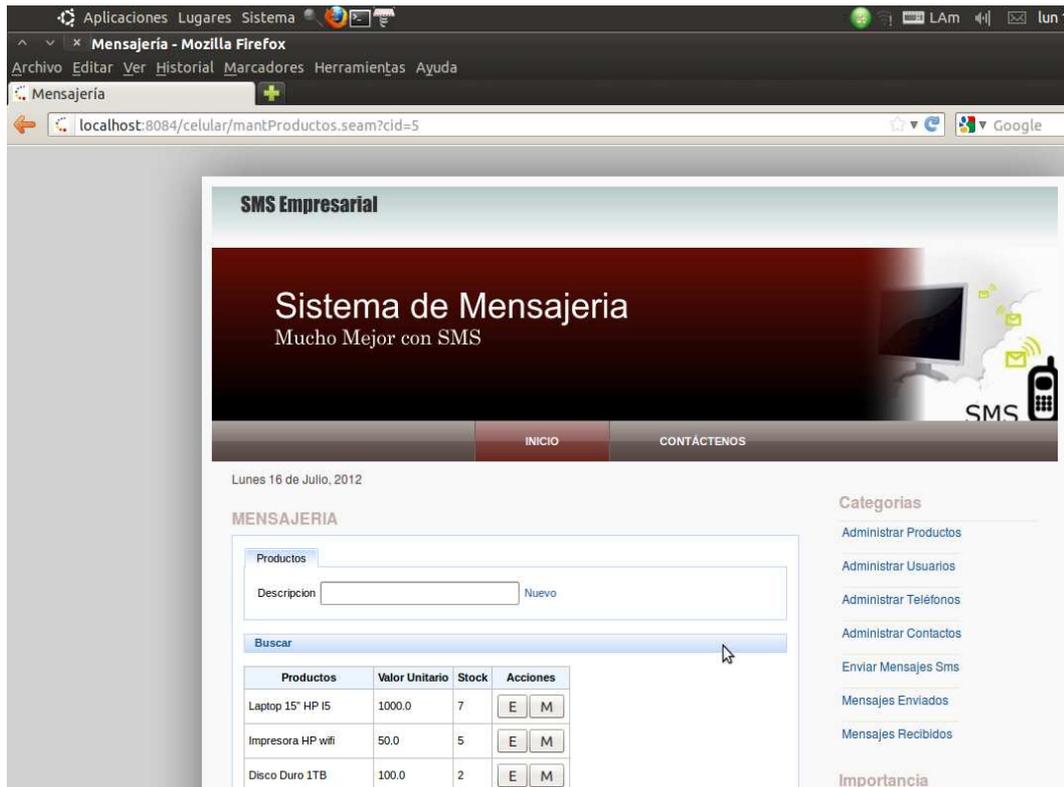
- Administrar Productos.
- Administrar Usuarios.
- Administrar Contactos.
- Enviar Mensajes SMS



- Mensajes Enviados.
- Mensajes Recibidos.

A continuación detallaremos cada categoría.

### **Administrar Productos.**



El usuario puede ingresar los productos en el sistema, esto puede ser migrado desde otra base de datos, los datos importantes para la categoría productos es la descripción, el valor unitario y el stock debido a que necesitamos información relevantes para el envío de mensajes SMS, además el usuario podrá modificar o eliminar dicho producto.

También cuando realiza peticiones de productos desde la aplicación móvil, automáticamente se actualiza el stock de los productos.



## Administrar Usuarios.

Usuarios

Cedula

Nombre

Nuevo Buscar

Cedula	Nombre	Email	Entrada	Salida	Acciones
0104050133	jorge cabrera	jjeo84@hotmail.com	Wed Dec 31 23:00:00 GMT-05:00 1969	Wed Dec 31 20:00:00 GMT-05:00 1969	E M
0103759718	carlita cabrera	carlacab@hotmail.com			E M

Categorias

- Administrar Productos
- Administrar Usuarios
- Administrar Telefonos
- Administrar Contactos
- Enviar Mensajes Sms
- Mensajes Enviados
- Mensajes Recibidos

Importancia

Mediante el envío de SMS, las Empresas u Organizaciones pueden realizar acciones de productividad u

El usuario puede ingresar sus empleados en el sistema, esto puede ser migrado desde otra base de datos, los datos importantes para la categoría usuarios es la cédula, el nombre y el email debido a que necesitamos información relevante para el envío de mensajes SMS.

El usuario o empleado se puede registrar vía Web, cuando el mensaje es enviado desde el móvil, el servidor Gateway SMS obtiene el mensaje y a continuación es insertado en la tabla de mensajes recibidos (mensajería), entonces un disparador o trigger almacenado en la base de datos procesa el mensaje para ver el tipo de SMS, valida e inserta los datos del usuario y automáticamente es almacenado el estado del usuario en la categoría contactos, también el usuario podrá modificar o eliminar dicho usuario.



## Administrar Teléfonos.

The screenshot shows a web browser window displaying the 'Sistema de Mensajería' interface. The page title is 'Sistema de Mensajería' with the tagline 'Mucho Mejor con SMS'. The interface includes a navigation menu with 'INICIO' and 'CONTÁCTENOS'. The main content area is titled 'MENSAJERIA' and features a search form for phone numbers. Below the search form is a table listing phone numbers and their characteristics.

Numero	Características	Acciones
092972518	Platform : SAMSUNG GT-C3510 - MemTotal: 1460 - *	E M
0910192341	Motorola Android	E M
093745812	Nokia 5300	E M
064838218	huawei	E M
093211111	RI AKK REDDY ultra final	E M

On the right side of the interface, there is a 'Categorías' menu with options: Administrar Productos, Administrar Usuarios, Administrar Teléfonos, Administrar Contactos, Enviar Mensajes Sms, Mensajes Enviados, and Mensajes Recibidos. Below this is an 'Importancia' section with the text: 'Mediante el envío de SMS, las Empresas u Organizaciones pueden'.

El usuario puede ingresar su número de teléfono en el sistema, los datos importantes para la categoría teléfono es el número y las características debido a que necesitamos información relevantes para el envío de mensajes SMS.

El usuario o empleado se puede registrar vía Web, cuando el mensaje es enviado desde el móvil, el servidor Gateway SMS obtiene el mensaje y a continuación es insertado en la tabla de mensajes recibidos (mensajería), entonces un disparador o trigger almacenado en la base de datos procesa el mensaje para ver el tipo de SMS, valida e inserta los datos del teléfono.



## Administrador Contactos.

El usuario puede ingresar los contactos en el sistema, los datos importantes para la categoría contactos son los datos del usuario (nombre, cédula), el número de teléfono y estado del usuario (Disponible, No disponible, salí a comer, etc.), debido a que necesitamos información relevantes para el envío de mensajes SMS.

El usuario o empleado se puede registrar vía Web, cuando el mensaje es enviado desde el móvil, el servidor Gateway SMS obtiene el mensaje y a continuación es insertado en la base de datos en la tabla de mensajes recibidos (mensajería), entonces un disparador o trigger almacenado en la base de datos procesa el mensaje para ver el tipo de SMS, valida el número del móvil, la cédula del usuario y actualiza los datos en la categoría contactos con su estado de acuerdo al usuario que ha enviado el SMS .



## Enviar Mensajes SMS.

En esta categoría consiste en el envío del mensaje para esto necesitamos los productos y los contactos. Aquí debemos seleccionar los productos necesarios y los contactos para posteriormente enviar el mensaje.

Productos    Contactos

Descripcion

Buscar

Productos	Valor Unitario	Stock	Sms
Laptop 15" HP i5	1000.0	7	<input checked="" type="checkbox"/>
Impresora HP wifi	50.0	5	<input type="checkbox"/>
Disco Duro 1TB	100.0	2	<input checked="" type="checkbox"/>
Pen Drive samsung 8GB	15.0	3	<input type="checkbox"/>
Memoria Ram 4GB	25.0	3	<input checked="" type="checkbox"/>

EnviarSms    Nuevo Sms    Stop

Caracteres: 81  
Contactos Seleccionados: 0, debe seleccionar al menos 1 contacto....

Mensaje

**Categorías**

- Administrar Productos
- Administrar Usuarios
- Administrar Teléfonos
- Administrar Contactos
- Enviar Mensajes Sms
- Mensajes Enviados
- Mensajes Recibidos

**Importancia**

Mediante el envío de SMS, las Empresas u Organizaciones pueden realizar acciones de productividad y además interactuar con los Clientes o Empleados.

**Aplicación Móvil**

Seleccionamos los productos.

Productos    Contactos

Nombre

Numero

Buscar

Telefono	Nombre	Estado	Agregar
092972518	jorge cabrera	Sali a Comer	<input checked="" type="checkbox"/>
064838218	gaby sol	Sali a Comer	<input type="checkbox"/>
083211111	carlita cabrera	Disponible	<input type="checkbox"/>
099887765	pepe	Hora No laborable	<input type="checkbox"/>

EnviarSms    Nuevo Sms    Stop

**Administración**

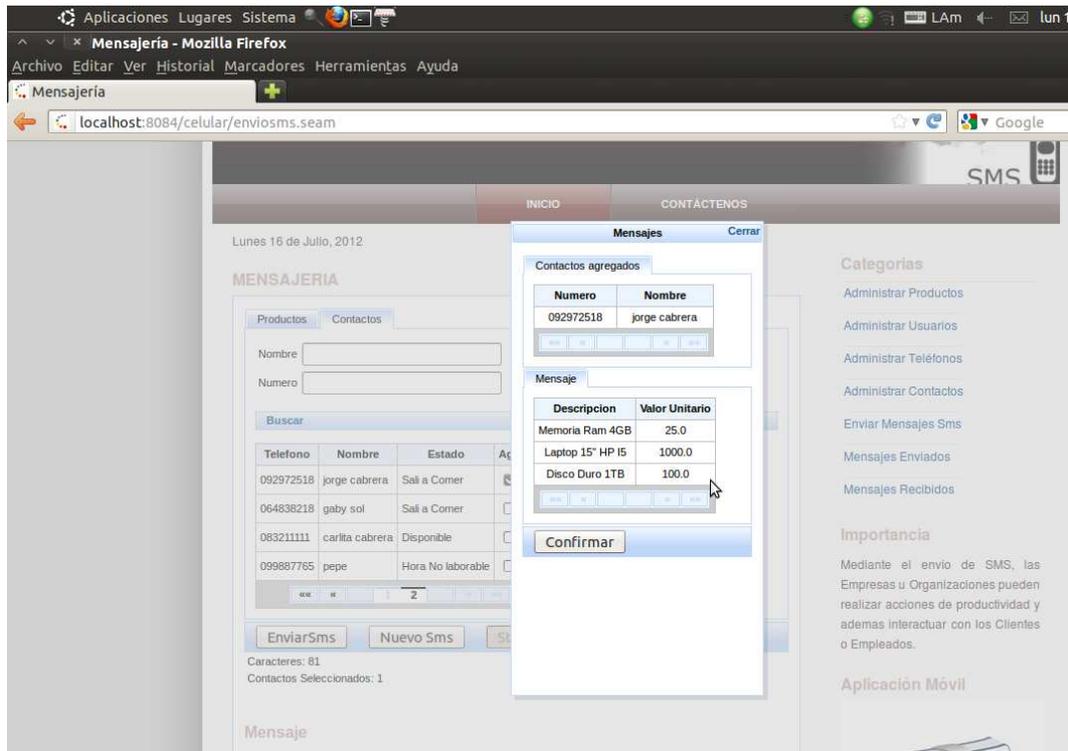
- Administrar Usuarios
- Administrar Teléfonos
- Administrar Contactos
- Enviar Mensajes Sms
- Mensajes Enviados
- Mensajes Recibidos

**Importancia**

Mediante el envío de SMS, las Empresas u Organizaciones pueden realizar acciones de productividad y además interactuar con los Clientes o Empleados.



Seleccionamos los contactos y obtenemos los números de los dispositivos móviles, damos click en enviar SMS y luego nos pedirá la confirmación del mensaje en un panel. El botón “EnviaSms” se habilita si tiene seleccionado los contactos y productos siempre y cuando no sobrepase de los 160 caracteres.



Para el envío del mensaje hemos utilizado un API Kannel de Java para realizar esta transacción.

```
sm=smsManager.sendSMS(host, port, username, password, from, to, text, udh, null, null, null, null, dlrmask, dlrurl, null, null, ...)
```

**Dónde:**

- Local host es el servidor.
- Host=13034: este puerto está configurado en el smsbox para el envío del mensaje.
- Username=user, password=pass: es el usuario y el password



respectivamente configurado en el smsbox.

- To: es el número del teléfono que va hacer enviadoel mensaje corto.
- Text: el mensaje para ser enviado.
- Udh: cabecera del mensaje, consiste un numero en hexadecimal e incluye el puerto a de la aplicación móvil.
- Dlrurl: es un url donde recibe el estado del mensaje.

### Mensajes Recibidos.

Tipo	De	Contenido del mensaje	Fecha
Peticion de Productos	092972518	Memoria Ram 4GB: \$25.0(1)	01/07/2012 09:16:19 AM
Peticion de Productos	092972518	Memoria Ram 4GB: \$25.0(5) Laptop 15" HP i5: \$1000.0(6) . Es como tener mis SMS es un email, SMS+	30/06/2012 08:07:29 PM
Peticion de Productos	092972518	Disco Duro 1TB: \$100.0(2) . Es como tener mis SMS es un email, SMS+	30/06/2012 07:56:26 PM
Estado Usuarios	092972518	0104050133 Salí a Comer . Es como tener mis SMS es un email, SMS+	30/06/2012 07:52:25 PM
Telefonos	092972518	Platform : SAMSUNG GT-C3510 - MemTotal: 1460 -. Es como tener mis SMS es un email, SMS+	30/06/2012 07:47:12 PM
		0104050133 jorge cabrera.jeo84@hotmail.com Wed Dec 31	30/06/2012

En el servidor Gateway SMS Kannel obtenemos una URL con tres parámetros en el grupo sms-service.

get-url =

"http://localhost:8084/celular/receivesms.jsp?sender=%p&text=%b&hora=%t"

- El número del móvil quien envía el mensaje.
- El mensaje en sí.



- La hora y fecha del envío del SMS.

En la url obtiene tres parámetros y es enviado al script para posteriormente insertar en la tabla kannel\_sms en la cual corresponde los mensajes recibidos.

### receivesms.jsp

```

<%String num=request.getParameter("sender");
String sms=request.getParameter("text");
String fecha = request.getParameter("hora");
        /*conexion sera nuestra conexion a la bd*/
try{
Class.forName(driver);

conexion = DriverManager.getConnection(url, usuario, clave);
String inserto = "insert into kannel_sms (message,num, fecha_sms) values
('"+sms+"', '"+num+"', '"+fecha+"')";
Statement s1 = conexion.createStatement();
int value = s1.executeUpdate(inserto);
if(sms.substring(0,1).equals("1"))
{
mensaje="Por favor espere un momento para realizar la actualizaciÃ³n.";
}
if(sms.substring(0,1).equals("2"))
{
mensaje="Su peticion de productos se realizo satisfactoriamente.";
}
if(sms.substring(0,1).equals("3")||sms.substring(0,1).equals("4")||sms.substring(0,1).equals("5"
))
{
mensaje="Su notificaciÃ³n se realizÃ³ satisfactoriamente.";
}
if(sms.substring(0,1).equals("6"))
{
String ced=sms.substring(1,11);
boolean v=s1.execute("SELECT * from usuarios where cedula ='"+ced);
        if(v)
        {
                mensaje="Su actualizaci3n de Usuario se realiz3 satisfactoriamente.";
        }
        else
        {
                mensaje="No se realiz3 actualizacion, no existe cedula";
        }
}
if(sms.substring(0,1).equals("7"))
        {mensaje="Su actualizacion de Telefono se realizo satisfactoriamente."; }
if(sms.substring(0,1).equals("8"))
{mensaje="Su actualizacion de Estado se realizo satisfactoriamente.";}
conexion.close();
session.setAttribute("conexion",conexion);
} catch (Exception ex){
mensaje=ex.toString();
mensaje="Error en mensaje recibido";
}
        %>
        <h3><%out.println(mensaje);%></h3>

```



Luego es enviado o notificado el resultado como respuesta al dispositivo móvil debido a la configuración del kannel.

Al insertar en la tabla kannel\_sms un disparador o trigger procesa el mensaje para ver qué tipo de SMS corresponde y lo envía a otra tabla de la base de datos procesada el mensaje, la cual almacena el tipo de mensaje recibido.

Los tipos de mensajes pueden ser:

- Usuarios
- Teléfonos
- Estado de usuarios
- Notificaciones
- Peticiones

Descripción del trigger.

```
delimiter //  
CREATE TRIGGER smsrecibe AFTER INSERT ON kannel_sms  
FOR EACH ROW BEGIN  
CALL mensajes(new. id_kannel, new. num, new. message);  
END //  
delimiter ;
```

En el disparador almacenamos un procedimiento en la cual este trigger es llamado cuando un SMS es recibido y es procesado el mensaje por este procedimiento.

En el plan de pruebas verificaremos el procedimiento correspondiente al trigger.



#### 4.1.2.2.2. Configuración del Gateway kannel

##### Instalación

```
1. wget http://www.kannel.org/download/1.4.3...y-1.4.3.tar.gz
2. tar -xvzf gateway-1.4.3.tar.gz
3. cd gateway-1.4.3
4. ./configure --with-mysql --with-mysql-dir=/var/lib/mysql --enable-start-stop-
daemon
```

##### Configuración

De acuerdo a nuestras necesidades la configuración de la pasarela Kannel se divide en grupos.

- Grupo BEARERBOX(Núcleo o core)
- Grupo Conexiones SMSC
- Grupo MÓDEMS
- Grupo SMSBOX
- Grupo SENDSMS
- Grupo de Servicios

**Grupo Bearerbox:** Es el núcleo, consiste en la conexión de un SMSC a un Teléfono, además es el que se encarga de enviar y recibir los mensajes SMS a un nivel bajo.

##### Configuración:

- Asignar un password.
- Comunicación con el servidor Kannel desde el mismo equipo (127.0.0.1).



- El puerto es una variable obligatoria, el número de puerto en el que el bearerbox escucha HTTP.
- Esto NO es lo mismo que el puerto HTTP del servidor local HTTP, simplemente se puede asignar cualquier puerto.

```
group = core
admin-port = 13025
smsbox-port = 13026
admin-password = cabrera
admin-deny-ip = "*.*.*.*"
admin-allow-ip = "127.0.0.1"
#dlr-storage = internal
dlr-storage = mysql
log-file = "/var/log/kannel/bearerbox.log"
log-level = 0
box-deny-ip = "*.*.*.*"
box-allow-ip = "127.0.0.1"
access-log = "/var/log/kannel/access.log"
access-log-clean =true
store-file = "kannel.store"
```

**Grupo Smsbox:** La configuración de este grupo provee un nivel más alto de manejo de mensajes SMS después de que se han recibido de los centros de SMS por bearerbox, o antes de que se transmitan a bearerbox para la entrega.

- Puerto.
- Indicar en qué host está el bearebox (bearerbox-host).
- Configurar el ámbito global-remitente (global-sender): el número del Modem GSM, por lo tanto es el mismo del campo my-number del grupo SMSC.
- Configurar un registro y el nivel de información a obtener: similares al grupo Bearerbox.



```
# SMS box setup
group = smsbox
smsbox-id = dlrbox
bearerbox-host = 127.0.0.1
sendsms-port = 13034
sendsms-chars = "0123456789"
global-sender = +59384996806
log-file = "/var/log/kannel/smsbox.log"
log-level = 0
access-log = "/var/log/kannel/acces.log"
```

**Grupo SmsC:** La configuración de este grupo se crea las conexiones SMSC, para que el núcleo o bearebox pueda conectarse a los centros SMS, con el fin de enviar y recibir los mensajes SMS.

Para la configuración son necesarios los siguientes parámetros:

- Permitir conexiones desde el servidor local.
- Configurar los parámetros de conexión con el Modem GSM, como el dispositivo (device) y la velocidad (speed). Para conocer los parámetros correctos, es necesario tener instalado el controlador USB para módems GSM y se necesita el paquete de wvdial, junto con su utilidad de configuración wvdialconf.

Pasos.

- Para instalar wvdial: apt-get install wvdial wvdialconf.
- Para generar la configuración, tenemos que ejecutar la utilidad Wvdialconf: wvdialconf /etc/wvdial.conf.
- Al ejecutar wvdial, los parámetros necesarios son:
- Device= /dev/ttyUSB0.
- Speed = 9600 baudios.
- También se puede ejecutar la herramienta Dmesg para conocer el puerto o dispositivo al que está conectado el Modem GSM. El resultado



de ejecutar Dmesg se ilustra en la Figura.

```
Found a modem on /dev/ttyUSB0.  
Modem configuration written to /etc/wvdial.conf.  
ttyUSB0<Info>: Speed 9600; init "ATQ0 V1 E1 S0=0 &C1 &D2"  
ttyUSB2<Info>: Speed 9600; init "ATQ0 V1 E1 S0=0 &C1 &D2"  
root@usuarioUbu:/home/cabrera#
```

Configurar los parámetros de la red GSM: número del SIM (my-number) y el número del centro SMS (sms-center).

```
##### SMSC / GSM modem setup  
group = smsc  
smsc = at  
smsc-id = smscabrera  
modemtype = auto  
port = 10000  
receive-port = 10000  
device = /dev/ttyUSB0  
my-number = +59384996806  
log-level = 0  
sim-buffering = true  
connect-allow-ip = 127.0.0.1
```

**Grupo Módems:** En este grupo se configura los parámetros del modem que va a ser utilizado para la comunicación con el Gateway, además para conocer los parámetros necesitamos una herramienta instalada en nuestro sistema operativo, consiste en obtener el puerto, la información del modem y para iniciar el modem.

Esta herramienta es Minicom.

Ejecutamos: `minicom -o /dev/ttyUSB0`

El resultado es: "ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0", por lo tanto esta es la cadena para iniciar (init-string) el Módem.

En este proyecto se ha optado por un módem HUAWEI E173 de movistar, que permite la comunicación, para poder cumplir con los objetivos de enviar y



recibir mensajes SMS.

```
group = modems
id = huawei_e173
name = "Huawei E173"
detect-string = "huawei"
init-string = "AT+CMEE=1" #línea para obtener toda la información del Módem.
init-string = "ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0" #línea para iniciar el
#Módem
message-storage = sm
need-sleep =true
speed = 9600
```

**Grupo sendsms-user:** La configuración de este grupo permite configurar o crear usuarios que son utilizados cuando es empleada para enviar mensajes SMS PUSH, es decir mediante http, con una URL.

Este grupo permite que las aplicaciones web puedan enviar mensajes SMS usando Kannel. En este grupo se configuran las solicitudes simples vía HTTP, y básicamente se requiere un usuario y contraseña.

Eje:

<http://localhost:13034/cgi-bin/sendsms?username=user&password=pass&to=+092972518&text=enviando SMS de prueba>

```
group = sendsms-user
username = user
password = pass
max-messages = 10
concatenation = true
```

**Grupo sms-service:** En esta configuración el grupo permite dar respuesta a los mensajes SMS que son enviados por los teléfonos móviles, la respuesta o resultado se obtiene en el contenido que devuelve el script en este caso el archivo jsp, es decir da respuesta al mensaje del celular. Para este proyecto se ha definido el servicio que se identifica con su keyword.



Valor Default (Sin valor en Keyword): Utilizado en la etapa de desarrollo y depuración, permitiendo almacenar todos los mensajes recibidos en una base de datos, con el fin de hacer pruebas de recepción de mensajes.

```
# SMS services definition, what to do when an SMS is recieved
group = sms-service
keyword = default
get-url = "http://localhost:8084/celular/receivesms.jsp?sender=%p&text=%b&hora=%t"
accept-x-kannel-headers = true
max-messages = 3
concatenation = true
catch-all = true
```

#### **4.1.2.2.3. Aplicación móvil**

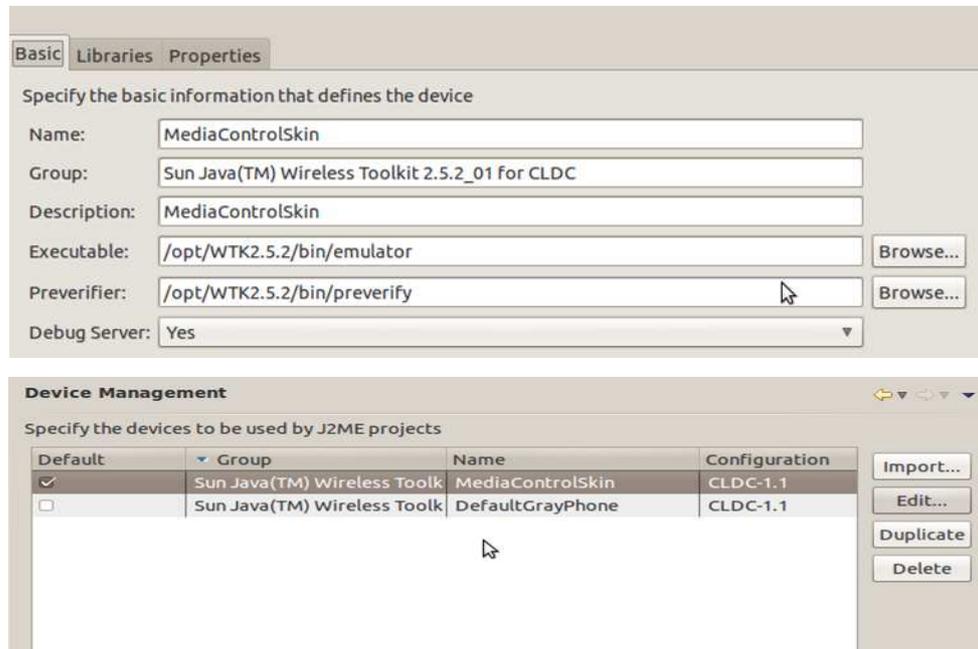
La aplicación móvil consiste en un programa de gestión empresarial para la automatización de procesos dentro de una organización.

Anteriormente se ha seleccionado el lenguaje J2ME para el desarrollo de la aplicación, el entorno de desarrollo es de libre distribución como lo es Eclipse y con el emulador para probar la aplicación.

J2ME Wireless Toolkit posee su propio emulador, implementa una API para el envío de mensajes SMS, además tiene su propia máquina virtual, sus perfiles y configuraciones para el desarrollo de aplicaciones móviles.



Este API es configurado en nuestro entorno de desarrollo Eclipse para su posterior implementación.



A continuación detallaremos la aplicación móvil para la gestión empresarial.

Al ejecutar la aplicación se pedirá autenticación de usuario ya que necesita la clave para iniciar el programa móvil y poder utilizar todas las opciones.





El menú principal se divide en:

- Configuración
- Control Usuario
- Productos
- Petición
- Notificación

Posteriormente especificaremos cada uno de las opciones.

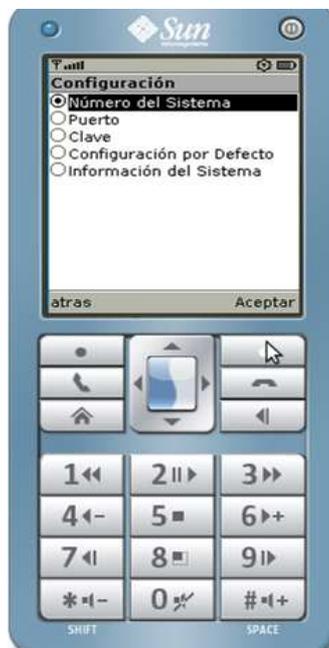




## Menú Configuración

Al entrar en configuración obtendremos una lista de opciones.

- Numero de sistema. - En este caso es el número del servidor SMS.
- Puerto.- Se puede configurar el puerto por donde va a salir el SMS.
- Clave.- Se puede cambiar la clave de la aplicación.
- Configuración por defecto.
- Información del sistema.





En la opción número del sistema, el usuario podrá cambiar el numero ya que puede existir algún cambio dentro del servidor, en este caso el número del Gateway, para realizar el cambio tendrá que ingresar la clave de la aplicación y confirmar.



En puerto, el usuario podrá modificar el número de puerto, esta opción es utilizada para el envío del SMS y de igual manera de ingresar la clave y la confirmación.





En esta opción se podrá visualizar la información de la aplicación, la cual presenta la configuración, el perfil, capacidad de memoria, etc.

El usuario al dar clic en aceptar tendrá la opción de enviar un SMS al Gateway para registrar las características del teléfono y almacenar en la base de datos.



### **Menú Control Usuario**

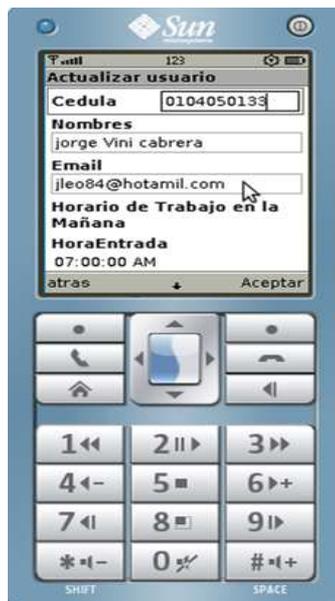
En este menú tenemos una lista de opciones:

- Usuario.
- Estado del Usuario.
- Eliminar Usuario.





La aplicación permite registrar los datos personales del usuario, la cedula, nombres, email y los horarios del trabajo, estos datos se almacena en el celular, además puede enviar un SMS al servidor kannel el cual recibe el mensaje y lo procesa para actualizar la información del usuario.



Confirmación del envío del SMS con los datos del usuario.





De igual manera el usuario dispone un estado que es almacenado en el celular y es enviado al servidor para notificar su estado actual y así gestionar o controlar al usuario.

- El estado del usuario es:
- Disponible
- No Disponible
- Ausente
- Salí a Comer
- Hora no Laborable





## Menú productos

En el menú productos existe una lista con las siguientes opciones:

- Visualizar.
- Actualizar Productos.
- Eliminar productos.



Al entrar en la opción “Visualizar” se podrá encontrar los productos disponibles con sus precios respectivos recibidos desde el servidor kannel.





En la opción “Actualizar Productos” podrá enviar un SMS al Gateway para realizar una petición y poder recibir productos actuales.



### **Menú Peticiones**

El usuario o empleado podrá realizar peticiones de acuerdo a las necesidades del cliente, el usuario podrá seleccionar él o los productos que el cliente requiera de acuerdo a su costo y a la cantidad que se encuentra en stock y obtiene al instante la respuesta de su petición.



Al seleccionar los productos el usuario ingresara la cantidad del producto para



realizar su petición, inmediatamente se envía un SMS al Gateway y obtenga al instante la respuesta de su petición.



Nota: Lo que está entre paréntesis corresponde la cantidad en stock de cada producto

### **Menú Notificaciones**

El usuario o empleado podrá realizar notificaciones de los productos que se encuentra en bodega.

Las notificaciones que podrá realizar es:

Productos defectuosos.- Son los productos que tiene algún tipo de problema o daños físicos.

Productos Fuera de stock.- Son los productos que ya no existe en bodega.

Productos embodegados.- Son los productos que han estado embodegados o almacenados en bodega y son notificados de su existencia.



Al seleccionar el tipo de notificación la siguiente opción es elegir el producto e ingresar la cantidad e inmediatamente se envía un SMS al Gateway y obtenga al instante la respuesta de su notificación.



Método para el envío del mensaje a través de la aplicación móvil.

```

MessageConnection conn = null;
TextMessage msg = null;
String r="";
Conn = (MessageConnection)Connector.open("sms://"+getTelefono()+":"+pt);
// metodo para la conexion y ademas obtenemos el numero del gateway y el puerto que esta
// almacenado en los RecordStore
msg = (TextMessage) conn.newMessage(MessageConnection.TEXT_MESSAGE);
// recibimos un vector de la opciones seleccionadas y concatenamos para realizar un solo texto
// para poder enviar el mensaje

String sms="";
txtsms[0]=id+txtsms[0];
for(int i=0;i<txtsms.length;i++)
{
if(txtsms[i]!=null)
{
sms=sms+txtsms[i]+"-";
}
}

msg.setPayloadText(sms);

// Enviamos el SMS
conn.send(msg);

```



Método para recibir el mensaje en la aplicación móvil para eso debemos obtener el puerto desde el record store.

```
MessageConnection mc = (MessageConnection)
Connector.open("sms://:"+pt);
Message msg = mc.receive();
```

Un record store es un conjunto de registros, y un registro es un byte array de datos de tamaño variable. Un record store está representado por un objeto de la clase RecordStore.



### 4.1.3. PLAN DE PRUEBAS

#### SISTEMA DE MENSAJERÍA Y DE LA APLICACIÓN MÓVIL

##### **Propósito**

Este documento describe el plan de pruebas para EL SISTEMA DE MENSAJERIA Y LA APLICACION MOVIL. Para este fin se definen los siguientes objetivos:

- Identificar los elementos que se van a probar.
- Detallar la estrategia de pruebas.
- Especificar los recursos necesarios para llevar a cabo las pruebas.

##### **Ámbito**

De acuerdo a los requerimientos, debemos realizar las pruebas para verificar si cumple con todo lo desarrollado.

Este Plan de Pruebas describe las pruebas de integración, además el conocimiento de la lógica interna del código del sistema.

El objetivo es probar el sistema de mensajería y la aplicación móvil desarrollada para detectar si no existe ningún tipo de error.

##### **Requerimientos de las pruebas**

La lista proporcionada en esta sección identifica los elementos o las características que tiene el proyecto de mensajería, incluyendo la aplicación móvil ya que conforma un sólo sistema.

Se ha definido los siguientes tipos de prueba que se explicaran a continuación:

- **Pruebas de caja blanca o estructural:**
  - ✓ Verificar el conocimiento de la lógica interna del código del sistema y la aplicación móvil.
- **Pruebas de integración:**
  - ✓ Verificar el acceso al sistema de mensajería empresarial.
  - ✓ Verificar la recuperación correcta de las modificaciones realizadas en la base



de datos tanto en el sistema de mensajería como en la aplicación móvil.

- ✓ Verificar accesos simultáneos de lectura de datos.

- **Pruebas de funcionalidad:**

Servidor Gateway

- ✓ Verificar la comunicación entre el baarebox y smsbox del Gateway kannel.
- ✓ Verificar la ejecución de los servicios del Gateway (baarebox y smsbox)

Sistema de mensajería.

- ✓ Verificar el ingreso de los productos, usuarios, teléfonos en el sistema de mensajería (aplicación web).
- ✓ Verificar el envío del SMS a través del sistema de mensajería, incluyendo el Gateway kannel.
- ✓ Verificar los mensajes recibidos en el sistema de mensajería.

Aplicación móvil.

- ✓ Verificar la instalación correcta de la aplicación en el móvil.
- ✓ Verificar los productos recibidos mediante un SMS desde el servidor.
- ✓ Verificar el ingreso de la información de los usuarios, teléfonos peticiones, notificaciones en la aplicación móvil.
- ✓ Verificar el envío del SMS a través de la aplicación móvil.

### **Estrategia de prueba**

En este apartado se presenta el enfoque que vamos a utilizar para probar el sistema de mensajería empresarial y la aplicación móvil. Se define cómo se realizaran las pruebas.

### **Tipos de pruebas y técnicas**

- Pruebas de caja blanca o estructural
  - ✓ También llamadas pruebas estructurales, utilizaremos el código fuente del programa y especialmente su estructura de control, para realizar los casos de prueba.



- ✓ Realizaremos varias estrategias que permita obtener casos de prueba a partir del código fuente.
- ✓ Vamos a elegir una determinada propiedad o característica del código tome el mayor número posible de valores.
- ✓ Prueba de bucles permitiendo como se basan la ejecución de esta estructura.

- Pruebas de integración.

Objetivos de la prueba	Comprobar que funcionen correctamente los procedimientos y métodos de acceso a la base de datos. Y descubrir los errores que se puedan producir en la interacción entre la aplicación móvil y el sistema de mensajería.
Técnicas	Invocar los procedimientos y métodos de acceso a la base de datos con datos válidos e inválidos.
Criterios de finalización	Los procedimientos y métodos de acceso funcionan como se diseñaron y sin errores.
Consideraciones	Las pruebas pueden necesitar un entorno de desarrollo DBMS para recuperar o modificar datos directamente.

- ✓ Caso de prueba 1

Objetivo: Probar la acción del procedimiento para procesar el SMS recibido.

- ✓ Condiciones de prueba.

En el servidor Gateway SMS Kannel se obtendrá una URL con tres



parámetros en el grupo sms-service.

get-url =

```
"http://localhost:8084/celular/receivesms.jsp?sender=%p&text=%b&hora=%t"
```

Condiciones:

- El número del móvil quien envía el mensaje.
- El mensaje en sí.
- La hora y fecha del envío del SMS.

Luego es enviado o notificado el resultado como respuesta al dispositivo móvil debido a la configuración del kannel.

Al insertar en la tabla kannel\_sms un disparador o trigger procesa el mensaje para ver qué tipo de SMS corresponde y lo envía a otra tabla de la base de datos procesada el mensaje, la cual almacena el tipo de mensaje recibido.

Los tipos de mensajes pueden ser:

- ✓ Usuarios
- ✓ Teléfonos
- ✓ Estado de usuarios
- ✓ Notificaciones
- ✓ Peticiones

Al procesar el SMS, verifica el tipo de mensaje y a continuación realiza su respectiva función.

- Pruebas de funcionalidad.

Las pruebas de funcionalidad se centran en los requisitos que presentan en las interfaces de usuarios y reglas de negocio. El objetivo de estas pruebas es verificar la aceptación, procesamiento y recuperación de datos y la adecuada implementación de las reglas de negocio. Este tipo de pruebas están basadas en



técnicas de caja negra, verifica la aplicación interactuando a través de las interfaces.

Objetivos de la prueba	Acreditar la comunicación correcta de la aplicación, la entrada de datos, su procesamiento y recuperación.  Considerando al sistema de mensajería y la aplicación móvil.
Técnicas	Ejecutar cada proceso con datos válidos e inválidos para verificar lo siguiente:  Cuando se utilizan datos correctos se obtienen los resultados esperados.  Cuando se utilizan datos incorrectos se obtienen los mensajes de error o advertencias adecuadas.
Criterios de finalización	Todas las pruebas planificadas se han ejecutado.  Todos los defectos identificados se han considerado.
Consideraciones	Ninguna.

### **Herramientas**

Las herramientas que se utilizaran para llevar a cabo el proceso de prueba son:

<b>Tipo de Prueba</b>	<b>Herramienta</b>
Herramienta DBMS	Mysql
Interfaz de usuario	Sistema de mensajería y la aplicación móvil.
Funcionales	Servidor Web, gateway kannel, SQLyog, JDK. 1.6.
Rendimiento	Suite eTest de Empirix



## Recursos

A continuación describiremos los recursos necesarios para realizar el proceso de prueba.

- Recursos hardware

Recurso	Cantidad	Descripción
PC-1	1	Diseño de las pruebas y Ejecución de las pruebas
Dispositivo Móvil	1	Diseño de las pruebas y Ejecución de las pruebas

- Recursos software

Nombre del elemento software	Tipo y otras notas
Mysql	Herramienta DBMS
Sistema de mensajería y la aplicación móvil.	Interfaz de usuario
Servidor Web, gateway kannel, SQLyog, JDK. 1.6.	Funcionales
Suite eTest de Empirix	Rendimiento

- Recursos humanos

RECURSOS HUMANOS		
Rol	Recursos	Responsabilidades
Gestor de prueba	1	Facilitar dirección técnica. Obtener los recursos apropiados.
Diseñador de prueba	1	Organizar el Plan de pruebas. Diseñar los Casos de prueba.



Probador	1	Ejecutar pruebas. Recuperar los errores. Documentar defectos.
----------	---	---



#### 4.1.4. EJECUCIÓN DE PLAN DE PRUEBAS

- **Pruebas de Cajas Blancas o Estructural**

En este tipo de pruebas verificaremos la lógica interna del sistema de mensajería y la aplicación del móvil. Para el envío del mensaje hemos utilizado un API Kannel de Java para realizar esta transacción.

A continuación se presenta un ejemplo para verificar el envío de SMS desde el sistema de mensajería.

```
public String smsenviaThread(String sms,String numero) {  
    SMSManager smsManager = SMSManager.getInstance();  
        smsManager.setMessagesPrefetchSize(10);  
  
smsManager.setMessagesSendRate(50);  
  
    String host="localhost";  
  
    String port="13034";  
  
    String username="user";  
  
    String password="pass";  
  
    String from="084996806";  
  
    String to="";  
  
    String text=sms;  
  
    String udh="%06%05%04%c3%50%00%00";  
  
    String dlrurl = "http://127.0.0.1/mensajes/dlr.php?type=%d&sm-sc-  
id=%i&answer=%A&service=%n";
```



```
try {
if(numero.length()>0 && (!sms.equals(null)|| !sms.trim().equals(""))) {
if(numero!=null){
to=numero;
sm=smsManager.sendSMS(host, port, username, password, from, to, text, udh,
null, null, null, null, dlrmask, dlurl, null, null, null);
mensa = "Mensaje Enviado exitosamente....."+sm;
smsManager.sendSMS
smsManager.stopSMSManagerWorker();
conecta = true;
}
} else
{
mensa = "Mensaje No Enviado, posiblemente no ha seleccionado contactos o
productos.....";
conecta = false;
}
return mensa;
} catch (Exception ex) {
mensa=ex+"....."+sm;
conecta = false;
ex.printStackTrace();
return mensa;
```



**Dónde:**

- Host=Local host es el servidor.
- port=13034: este puerto está configurado en el smsbox para el envío del mensaje.
- Username=user, password=pass: es el usuario y el password respectivamente configurado en el smsbox.
- To: es el número del teléfono que va hacer enviadoel mensaje corto.
- Text: el mensaje para ser enviado.
- Udh: cabecera del mensaje, consiste un numero en hexadecimal e incluye el puerto a de la aplicación móvil.
- Dlrurl: es un url donde recibe el estado del mensaje.

**Caso de prueba 1**

✓ Objetivo

Obtener el resultado de acuerdo a lo requerido por el usuario, verificar que no exista ningún error. Consiste en el envío de SMS desde el sistema de mensajería.

✓ Condiciones de prueba

Se considera la condición principal en la ejecución del Gateway Kannel, esto quiere decir que deben estar levantados los servicios de dicho Gateway.



La condición previa es que los contactos sean seleccionados más de 1 y los productos de igual manera ya que este consiste el mensaje en texto.

```
if(numero.length()>0 && (!sms.equals(null)|| !sms.trim().equals(""))) {  
  
if(numero!=null){  
  
to=numero;  
  
sm=smsManager.sendSMS(host, port, username, password, from, to, text, udh, null, null, null,  
null, dlrmask, dlrurl, null, null, null);  
  
mensa = "Mensaje Enviado exitosamente....."+sm;  
  
smsManager.sendSMS  
  
smsManager.stopSMSManagerWorker();  
  
conecta = true;  
  
}  
  
}
```

En la prueba de bucle consiste en verificar esta estructura de control que permite repetir una o más sentencias múltiples veces. En este caso el envío de SMS se va a repetir de acuerdo a los números de celulares almacenados en un array, ya que es llamado por un Thread y es convocado varias veces la clase Sms para ser enviado el mensaje corto.

```
public void processUpdates() {  
  
if(i<usutel.size())  
  
{  
  
Sms sm = new Sms();  
  
setMensa(sm.smsenviaThread(sms, usutel.get(i).getTelefono().getNumero()));  
  
}  
  
}
```



✓ Salidas.

Las salidas son los envíos de mensajes cortos a través del API Kannel de acuerdo a las condiciones, en este caso el resultado de los mensajes serán las salidas de verificación. El resultado final se guarda en una variable para posteriormente ser invocado para presentar en la interfaz de usuario.

- mensa = "Mensaje Enviado exitosamente.....";
- mensa = "Mensaje No Enviado, posiblemente no ha seleccionado contactos o productos.....";
- mensa = "Mensaje no enviado "+ex+".....";

Este último puede ser que no se haya levantado el servicio del Gateway.

Como resultado final hemos obtenido el mensaje satisfactorio de acuerdo a las condiciones.



Método para el envío del mensaje a través de la aplicación móvil.

A continuación se presenta un ejemplo para verificar el envío y recepción de mensajes SMS desde la aplicación del móvil.

```
public void run() {
    String address = "sms://"+getTelefono();
    MessageConnection smsconn = null;
    try {
        smsconn = (MessageConnection)Connector.open(address);
        TextMessage txtmessage=
        (TextMessage)smsconn.newMessage(MessageConnection.TEXT_MESSAGE);
        txtmessage.setAddress(address);
        txtmessage.setPayloadText(sendTexto());
        smsconn.send(txtmessage);
        band=0;
    }
    catch (Throwable t1) {
        band=1;
        String statusMessage = "Mensaje no Enviado, se almacena la informacion.....";
        sendingMessageAlert.setString(statusMessage);
        this.midlet.d.setCurrent(sendingMessageAlert);
        t1.printStackTrace();
    }
}
public String sendTexto() {
    String sms="";
    txtsms[0]=id+txtsms[0];
    for(int i=0;i<txtsms.length;i++)
    {
        if(txtsms[i]!=null)
        {
            sms=sms+txtsms[i]+"*";
        }
    }
    return sms;
}
```



## **Caso de prueba 2**

### ✓ Objetivo

Obtener el resultado de acuerdo a lo requerido por el usuario, verificar que no exista ningún error. Consiste en el envío de SMS desde la aplicación del móvil, así mismo al recibir un mensaje en dicha aplicación.

### ✓ Condiciones de prueba

Tomaremos como ejemplo el envío de mensajes de peticiones, de acuerdo a esto la condición previa es seleccionar los productos que el usuario necesita y las cantidades requeridas, de igual manera ya que en esto consiste el mensaje en texto.

Este método de la clase peticiones, consiste en seleccionar los productos y la condición es que seleccione máximo 5 productos, además de eso no sobrepase de la cantidad actual en stock de cada producto, y de igual manera valida si no ha seleccionado ningún producto, al ser convocado este método facilita las validaciones o condiciones para que posteriormente sea enviado el SMS.



```
if(aux<5)
{
if(menu.isSelected(aux))
{
smsAux[aux2]= menu.getString(aux);
aux2++;
}
}
else
{
Alert a = new Alert("", "Se permite seleccionar maximo 5 productos "+this.p
,null,AlertType.ERROR);
this.midlet.d.setCurrent(a); // Muestra un mensaje en pantalla
band=1;
break;
}
aux++;
}
if(band==0)
{
if(aux2==0)
{
Alert a = new Alert("", "No ha seleccionado ningun producto "+this.p ,null,AlertType.ERROR);
this.midlet.d.setCurrent(a); // Muestra un mensaje en pantalla
}
else
{
prod=smsAux;
fp= new FproductosCantidad(this.midlet,this);
fp.crea(p,op);
}
}
```



```

if (arg0 == aceptar)
{
if(op==2)//envio de peticiones y actualizacion productos
{
if(validacantidad()==true)
{
cantidadsms();
enviasms();
t=new Thread(this);
t.start();  }
else      {
Alert a1 = new Alert("Cantidad Incorrecta","Por favor revise cantidad de los productos "
,null,AlertType.ERROR);
this.midlet.d.setCurrent(a1);
}} }

```

Después que realice estas condiciones enseguida procede al envío de mensaje de texto.

La prueba de bucle consiste en verificar esta estructura de control que permite repetir una o más sentencias múltiples veces. En este caso recibimos un vector de productos seleccionados de igual manera concatenamos todo el vector con un asterisco, es decir, forma el mensaje de texto en sí, y nos va a servir en el sistema de mensajería para que procese el SMS.

```

public String sendTexto() {
String sms="";
txtsms[0]=id+txtsms[0];
for(int i=0;i<txtsms.length;i++) {
    if(txtsms[i]!=null){
        sms=sms+txtsms[i]+"*";
    }
}
return sms;
}

```



✓ Salidas.

A continuación enviamos el mensaje

```
msg.setPayloadText(sms); // este método corresponde a que el mensaje es de tipo de texto.
// Paso : Enviamos el SMS
conn.send(msg);
r="Mensaje Enviado";
```

Como resultado final hemos obtenido el mensaje satisfactorio de acuerdo a las condiciones del envío SMS de nuestra aplicación móvil.

- **Pruebas de integración**

A continuación presentamos el procedimiento para procesar el SMS. Este procedimiento recibe como parámetro el número del teléfono y el mensaje de texto. El mensaje recibido posee un número al inicio en la cual corresponde el tipo del SMS. La variable “tipo” obtiene el número y asigna el tipo de mensaje.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `mensajes`(IN idkannel INT,IN num
VARCHAR(15),IN sms VARCHAR(170))
BEGIN

SELECT SUBSTRING(trim(sms),-length(trim(sms)),1) into tipo;
SELECT SUBSTRING(trim(sms),2,length(trim(sms))) into smspro;
SET longitud = length(trim(smspro));
CASE tipo
when '1' THEN SET tiposms='Actualizar Productos';
when '2' THEN SET tiposms='Petición de Productos';
when '3' THEN SET tiposms='Notificación de productos Defectuosos';
when '4' THEN SET tiposms='Notificación de productos por Terminar';
when '5' THEN SET tiposms='Notificación de productos otro';
when '6' THEN SET tiposms='Usuarios';
when '7' THEN SET tiposms='Telefonos';
when '8' THEN SET tiposms='Estado Usuarios';
else SET tiposms='otros';
END CASE;
SELECT REPLACE(num, '+593', '0') into newnum;
SELECT idtelefono into idtel from telefonos where numero = newnum;
IF idtel>0 THEN
IF tiposms <> 'otros' THEN
INSERT INTO mensajerias (tiposms,id_kannel_fk,idtelefono_fk)
values(tiposms,idkannel,idtel);
END IF;

```



```

IF tipo like '2' THEN
/*      --aquí debo recibir el número de teléfono, estado y cedula de usuario*/

      WHILE cont1<longitud DO

          SET cont = cont + 1;
          SELECT SUBSTRING(trim(smspro),-length(trim(smspro))+cont1,1) into comp;
          IF comp = '*' THEN
              set tam=cont;

                  SELECT SUBSTRING(trim(smspro),i,tam-1) into sub;
              set descrip = sub;
                  SET longitud2 = length(trim(descrip));
              set cont2=0;
                  while cont2<longitud2 DO
                      SELECT SUBSTRING(trim(descrip),-
length(trim(descrip))+cont2,1) into comp;
                      IF comp = ':' THEN
                          set tami=cont2;
                          SELECT SUBSTRING(trim(descrip),1,tami) into descrip2;
                          END IF;
                          IF comp = '(' THEN
                              set tami=cont2+2;
                              set tamf= length(trim(descrip))-tami;
                              SELECT CONVERT(SUBSTRING(trim(descrip),tami,tamf)
,UNSIGNED INTEGER) into prod;
                              END IF;
                              SET cont2 = cont2 + 1;
                              END WHILE;

                                  SELECT stock into prodaux from productos where descripcion
like descrip2 ;
                                  set upprod=prodaux-prod;
                                  UPDATE productos set stock = upprod where descripcion like descrip2;

                                      set i = i+tam;
set cont=0;
          END IF;

          SET cont1 = cont1 + 1;
      END WHILE;
  END IF;

```



```

IF tipo like '6' THEN
/*--aqui debo recibir los atributos del usuario cedula nombre email y tambien inserat los horarios
---comparamos con cedula si existe o no para inserta o actiulizar y por valor por defecto ingresar el valor
de estado en usuariotelefonos
*/
    WHILE cont1<longitud DO

        SET cont = cont + 1;
        SELECT SUBSTRING(trim(smspro),-length(trim(smspro))+cont1,1) into comp;
        IF comp = '*' THEN
            set tam=cont;
            SELECT SUBSTRING(trim(smspro),i,tam-1) into sub;
            IF band = 0 THEN
                set cedul = sub;
            END IF;
            IF band = 1 THEN
                set nombr = sub;
            END IF;
            IF band = 2 THEN
                set emai = sub;
            END IF;
            IF band = 3 THEN
                set horadi = sub;
            END IF;
            IF band = 4 THEN
                set horatard = sub;
            END IF;

set i = i+tam;
set band = band+1;
        set cont=0;
        END IF;
        SET cont1 = cont1 + 1;
    END WHILE;

        SELECT idusuario into iduser from usuarios where cedula = cedul;
        IF iduser>0 THEN
            UPDATE usuarios set email=emai,horadia=horadi,horatarde=horatard where cedula =
cedul;
        END IF;
    END IF;

```



```

IF tipo like '7' THEN
    update telefonos set características = smspro where numero like newnum;
END IF;

IF tipo like '8' THEN
/* --en este tipo de sms se recibe el numero de telefono, estado y cedula de usuario para
actualizar el estado del usuario*/

        WHILE cont1<longitud DO

            SET cont = cont + 1;
            SELECT SUBSTRING(trim(smspro),-length(trim(smspro))+cont1,1) into comp;
            IF comp = '*' THEN
                set tam=cont;
                SELECT SUBSTRING(trim(smspro),i,tam-1) into sub;
                IF band = 0 THEN
                    set cedul = sub;
                END IF;
                IF band = 1 THEN
                    set estad = sub;
                END IF;
                set i = i+tam;
            set band = band+1;
            set cont=0;
            END IF;
            SET cont1 = cont1 + 1;
        END WHILE;
        SELECT idusuario into iduser from usuarios where cedula = cedul;
        UPDATE usuarios_telefonos set estado = estad where idusuario_fk = iduser and
idtelefono_fk=idtel;
        /* aqui en la aplicacion del movil debe estar validado, primero debe estar
configurado el usuario pata enviar la opcion 8 del estado*/
END IF;

/*update kannel_sms set message = smspro where id_kannel = idkannel;*/

END$$

DELIMITER ;

```



### Caso de prueba 1

✓ Objetivo

Probar la acción de procesar el mensaje recibido, al insertar un SMS en la tabla `kannel_sms` un trigger es llamado, éste posee un procedimiento almacenado en la cual procesa el mensaje para ver qué tipo de SMS corresponde y lo envía a otra tabla de la base de datos, procesa el mensaje y almacena el tipo de mensaje recibido.

Los tipos de mensajes pueden ser:

- Usuarios
- Teléfonos
- Estado de usuarios
- Notificaciones
- Peticiones

✓ Condiciones de prueba.

En el servidor Gateway SMS Kannel obtendrá una URL con tres parámetros en el grupo `sms-service`, la cual recibe el mensaje y activa el trigger, (Mediante el archivo `receivesms.jsp`).

Las condiciones son:

- El número del móvil quien envía el mensaje.
- El mensaje en sí.

El mensaje de texto tiene un número al inicio, este número corresponde el tipo de mensaje, al entrar al procedimiento verifica que tipo de mensaje es y procede a realizar sus condiciones.

Como por ejemplo el dispositivo móvil envía un SMS, es recibido y procesa el mensaje.

`60104838218*SANTIAGO SANCHEZ*santiago@hotmail.com*10:00:30-22:12:20*`



El procedimiento obtiene el primer valor, en este caso el número 6 entra en el flujo de control y obtenemos el tipo de mensaje, en este caso corresponde a Usuarios, luego divide en subcadenas debido al asterisco (\*).

En la división de esta cadena obtenemos: cédula, nombre, email, horario de trabajo, este mensaje corresponde a la actualización de un usuario en nuestro sistema de mensajería.

Además el número de teléfono reemplaza +593 por 0 ya que nos facilita la validación y consultas para verificar si existe o no este número de teléfono.

✓ Entrada.

+59364838218, 60104838218\*SANTIAGO SANCHEZ\*santiago@hotmail.com\*10:00:30\*22:12:20\*

✓ Salida deseada

En este caso se realizó una actualización ya que existe el usuario debido a una consulta de la cédula.

La actualización es satisfactoria.

Durante este caso de prueba se puede considerar que no existe ninguna anomalía o error durante la ejecución de este procedimiento, además los procedimientos y métodos de acceso funcionan como se diseñaron y sin errores.



- Pruebas de funcionalidad

### Servidor Gateway

#### Caso de prueba 1.

- ✓ Objetivo

En este caso de prueba se verifica los servicios de bearerbox y smsbox que inicie correctamente, para levantar los servicios abrimos un terminal y digitamos lo siguiente.

```
/usr/sbin/bearerbox -v 1 /etc/kannel/kannel.conf
```

```

root@usuarioUbu: /home/cabrera
Archivo Editar Ver Buscar Terminal Ayuda
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: <-- OK
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: --> AT+CSMS=?^M
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: <-- +CSMS: (0,1)
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: <-- OK
2012-06-14 23:20:06 [3463] [6] INFO: AT2[smscabrera]: Phase 2+ is supported
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: --> AT+CSMS=1^M
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: <-- +CSMS: 1,1,0
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: <-- OK
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: --> "ATQ0 V1 E1 S0=0 &C1
&D2 +FCLASS=0" #línea para iniciar el #Módem^M
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: <-- +CME ERROR: INVALID C
HARACTERS IN TEXT STRING
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: --> ATH0^M
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: <-- OK
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: --> AT+CPMS="sm"^M
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: <-- +CPMS: 0,35,0,35,0,35
2012-06-14 23:20:06 [3463] [6] DEBUG: AT2[smscabrera]: <-- OK
2012-06-14 23:20:06 [3463] [6] INFO: AT2[smscabrera]: AT SMSC successfully opene
d.
2012-06-14 23:20:08 [3463] [6] DEBUG: AT2[smscabrera]: --> AT+CPMS?^M
2012-06-14 23:20:08 [3463] [6] DEBUG: AT2[smscabrera]: <-- +CPMS: "SM",0,35,"SM"
,0,35,"SM",0,35
2012-06-14 23:20:08 [3463] [6] DEBUG: AT2[smscabrera]: <-- OK

```

```
/usr/sbin/smsbox -v 1 /etc/kannel/kannel.conf
```

```

root@usuarioUbu: /home/cabrera
Archivo Editar Ver Buscar Terminal Ayuda
cabrera@usuarioUbu:~$ su
Contraseña:
root@usuarioUbu:/home/cabrera# /usr/sbin/smsbox -v 1 /etc/kannel/kannel.conf
2012-06-14 23:22:31 [3626] [0] INFO: Debug_lvl = 1, log_file = <none>, log_lvl =
0
2012-06-14 23:22:31 [3626] [0] INFO: Starting to log to file /var/log/kannel/sms
box.log level 0
2012-06-14 23:22:31 [3626] [0] INFO: Added logfile /var/log/kannel/smsbox.log'
with level '0'.
2012-06-14 23:22:31 [3626] [0] INFO: Service global sender set as '+59384996806'
2012-06-14 23:22:31 [3626] [0] INFO: Logging accesses to '/var/log/kannel/acces.
log'.
2012-06-14 23:22:31 [3626] [0] INFO: Started access logfile /var/log/kannel/acc
es.log'.
2012-06-14 23:22:31 [3626] [0] INFO: HTTP: Opening server at port 13034.
2012-06-14 23:22:31 [3626] [0] INFO: Set up send sms service at port 13034
2012-06-14 23:22:31 [3626] [0] INFO: Connected to bearerbox at 127.0.0.1 port 13
026.

```



El valor 1 corresponde la información detallada de la conexión.

✓ Salida deseada

Hemos obtenido conexión satisfactoria, lo que nos indica que se comunica correctamente con el modem y el smsbox.

**Sistema de mensajería.**

**Caso de prueba 1.**

✓ Objetivo

En este caso de prueba se ha verificado que al ingresar valores reales en nuestro sistema de mensajería no presenta ningún problema ya que las condiciones y validaciones cumple con lo previsto.

Además en este caso de prueba se verificó que al enviar un SMS desde la aplicación web, cumple con las condiciones anteriores y no presenta ningún tipo de error ya que la comunicación con el kannel se encuentra correctamente configurada y ejecutada.



✓ Salida deseada

A continuación la siguiente figura representa al envío del SMS desde la aplicación web con su respectiva respuesta.



Este es el mensaje después del envío del SMS donde señala el número de caracteres y a cuantos contactos ha enviado.





En la siguiente figura representa el resultado final del envío del SMS como informe desde nuestra pasarela kannel.

```
root@usuarioUbu: /home/cabrera
Archivo Editar Ver Buscar Terminal Ayuda
2012-06-14 23:22:31 [3626] [0] INFO: Debug_lvl = 1, log_file = <none>, log_lvl =
0
2012-06-14 23:22:31 [3626] [0] INFO: Starting to log to file /var/log/kannel/sms
box.log level 0
2012-06-14 23:22:31 [3626] [0] INFO: Added logfile '/var/log/kannel/smsbox.log'
with level '0'.
2012-06-14 23:22:31 [3626] [0] INFO: Service global sender set as '+59384996806'
2012-06-14 23:22:31 [3626] [0] INFO: Logging accesses to '/var/log/kannel/acces.
log'.
2012-06-14 23:22:31 [3626] [0] INFO: Started access logfile '/var/log/kannel/acc
es.log'.
2012-06-14 23:22:31 [3626] [0] INFO: HTTP: Opening server at port 13034.
2012-06-14 23:22:31 [3626] [0] INFO: Set up send sms service at port 13034
2012-06-14 23:22:31 [3626] [0] INFO: Connected to bearerbox at 127.0.0.1 port 13
026.
2012-06-14 23:45:09 [3626] [3] INFO: smsbox: Got HTTP request </cgi-bin/sendsms>
from <127.0.0.1>
2012-06-14 23:45:09 [3626] [3] INFO: sendsms used by <user>
2012-06-14 23:45:09 [3626] [3] INFO: sendsms sender:<user:+092972518> (127.0.0.1
) to:< 092972518> msg:<Impresora HP: $50.0-Laptop 15" HP I5: $1000.0->
2012-06-14 23:45:09 [3626] [3] INFO: smsbox: Got HTTP request </cgi-bin/sendsms>
from <127.0.0.1>
```

## Caso de prueba 2.

Los mensajes recibidos son obtenidos correctamente por kannel mediante la url (recievesms.jsp), en la cual el SMS es almacenado sin problemas en la base de datos.

## Aplicación móvil.

### Caso de prueba 1.

Como primer paso debemos instalar la aplicación móvil en un celular, para esto obtenemos archivos de instalación.

- JAD. Donde se almacena las configuraciones y perfiles de nuestra aplicación.
- JAR. Donde se encuentra la aplicación en sí.

El resultado final de este caso de prueba se ha verificado que la instalación fue correctamente implementada en el celular sin presentar ningún tipo de anomalías.



### **Caso de prueba 2**

En este caso de prueba se verifico que al enviar un SMS desde la aplicación móvil, cumpliendo con las condiciones anteriores, no presenta ningún tipo de error ya que la comunicación con el Api de java se encuentra correctamente configurada y ejecutada.

### **Caso de prueba 3**

La aplicación móvil es activada por un mensaje SMS, luego es procesado el mensaje dividiendo en productos actualesy es almacenada en un record store, como resultado este caso de prueba se ha realizado satisfactoriamente.



## CONCLUSIONES

Al finalizar el proyecto concluimos diciendo que el estudio de los Sistemas Operativos y los lenguajes de programación móvil son bases fundamentales para el desarrollo de una aplicación móvil, se debe tener presente sus ventajas y características para tomar decisiones. La ventaja principal del sistema operativo es facilitar al usuario la utilización de aplicaciones sin ninguna restricción, sin embargo este estudio verificó cómo está estructurado cada sistema operativo y para qué tipo de celulares son implementados, además si los servicios de comunicación cumple con el requerimiento de cada proyecto, éste se refiere a la comunicación de red vía SMS, este estudio es relevante debido a que es la base y el punto de partida para el desarrollo de una aplicación móvil.

Para desarrollar una aplicación móvil se debe realizar un estudio detallado de los lenguajes de programación, el objetivo principal de desarrollar aplicaciones de usuario final, interfaz gráfica, redes de comunicación, etc., hay que enfocarse en las limitaciones de hardware o capacidades limitadas e interfaces muy reducidas de los dispositivos móviles, además que sea compatible con la mayoría de los sistemas operativos y principalmente que tenga licencia gratuita, si ese es una de las limitaciones.

En el desarrollo de la aplicación móvil, aplicación web y Gateway se ha optado por elegir herramientas gratuitas, además de utilizar requerimientos mínimos de hardware tanto para el desarrollo como para la ejecución, ya que nos brindan alternativas y ahorros de costo para el desarrollo del sistema de mensajería. En la aplicación móvil utiliza un API de java de mensajes cortos de texto o SMS con la finalidad de realizar operaciones empresariales, ya que la ventaja principal del uso de este servicio es su cobertura amplia y su bajo costo en la transacción de mensajes.

Para optimizar los servicios de la empresa, el empleado puede utilizar la aplicación móvil en varios lugares de la ciudad o país, realizando peticiones de los productos de acuerdo a las necesidades del cliente y obteniendo resultados instantáneos. Esto quiere decir que la empresa va donde el cliente, existiendo movilidad de empleados, dando así lugar a una ventaja competitiva.



De acuerdo a las configuraciones, el Gateway Kannel permite una lista de números accesibles al sistema de mensajería e interacción con la base de datos MYSQL, al configurar el Gateway se encontró más características por estudiar y agregar al servicio. En el transcurso de la vida de una aplicación surgen otros requerimientos que deben ser configurados e implementados, al finalizar el proyecto verificamos los objetivos planteados en la denuncia, cubriendo con todos los requerimientos y contando finalmente con un sistema de mensajería empresarial.



## RECOMENDACIONES

Antes de implementar el sistema de mensajería en una organización se debe conocer previamente la importancia de los tipos de celulares existentes, ya que debido a las configuraciones y perfiles que poseen se debe realizar un estudio de esto para posteriormente entregar la aplicación del móvil de acuerdo a las características con el fin de no presentar ninguna anomalía o restricción al instalar la aplicación.

En el caso del servidor, bajo las condiciones y recursos que tenga la empresa deberá decidir qué tipo de medio desea enviar o recibir sus SMS, estos pueden ser módems GSM o enlaces directos al SMSC del operador, seleccionando la operadora que más le beneficie a la organización.

Como sabemos un mensaje corto o SMS tiene un estándar de 160 caracteres, la cual permite que el mensaje enviado por el servidor considere y valide esta longitud, permitiendo un mensaje limitado, entonces recomendamos si la empresa necesita ofertar varios productos y obtener mayor productividad, los productos en sí, se codifique de alguna manera (menos caracteres) y de este modo la aplicación del móvil obtenga sus productos actuales.

En la actualidad las aplicaciones móviles son un eje importante en las empresas debido a que las tecnologías de información y comunicación presentan ventajas importantes para la organización, considerando este punto la carrera de Ingeniería de Sistemas debe enfocarse a la programación en dispositivos móviles, mediante cursos o incluyendo en el pensum de la malla curricular.

En la programación móvil existe una amplia gama de servicios, ya que se pueden ampliar o evolucionar la aplicación desarrollada por ejemplo enviar archivos multimedia, imágenes y porque no los GPS, dependiendo de las necesidades de la empresa. Es importante el crecimiento del sistema de mensajería por lo tanto recomendamos un estudio minucioso del Gateway Kannel ya que cuenta con varias características que pueden ser implementados en el sistema desarrollado, estas características adicionales pueden ser como el envío de archivos multimedia, imágenes, etc.



## BIBLIOGRAFIA

### **Sistemas Operativos**

- <http://www.muymac.com/2011/06/06/ios-5-caracteristicas-detalles-especificaciones>
- [http://es.wikipedia.org/wiki/Symbian\\_OS](http://es.wikipedia.org/wiki/Symbian_OS)
- [http://docs.blackberry.com/es-es/admin/deliverables/12272/BlackBerry\\_Enterprise\\_Server-T323212-832026-1023093329-005-5.0.1-ES.pdf](http://docs.blackberry.com/es-es/admin/deliverables/12272/BlackBerry_Enterprise_Server-T323212-832026-1023093329-005-5.0.1-ES.pdf)
- <http://www.blogdemoviles.com.ar/android-4-0-caracteristicas/>
- <http://es.scribd.com/doc/21782041/Sistemas-Operativos-de-los-celulares>
- <http://es.scribd.com/doc/24158503/Sistemas-Operativos-Para-Moviles>
- <http://es.scribd.com/doc/18308200/ANALISIS-TECNICO-DE-LOS-SISTEMAS-OPERATIVOS-SYMBIAN-Y-WINDOWS-MOBILE>
- <http://es.scribd.com/doc/18308077/Analisis-Parte-2-Nokia-3250-Sistema-Operativo-Symbian>
- <http://es.scribd.com/doc/15059840/Linux-Plus-Magazine-030Marzo-2007>

### **Lenguajes de programación**

- <http://es.scribd.com/doc/7136584/J2ME#>
- <http://perso.wanadoo.es/tutoriales/cursos/wap/index.htm>
- <http://www.slideshare.net/pythons60/ppt-python-s60-presentation>
- [http://www.slideshare.net/pys60/python-para-s60-presentation?src=related\\_normal&rel=876187](http://www.slideshare.net/pys60/python-para-s60-presentation?src=related_normal&rel=876187)
- <http://es.scribd.com/doc/23989488/j2me-y>
- <http://www.slideshare.net/vivian7894/j2me-presentacion>
- <http://www.slideshare.net/javiergs/200710-java-para-dispositivos-mviles>
- <http://www.youblisher.com/p/37009-Please-Add-a-Title/ microsoft>



- <http://es.scribd.com/doc/4294669/conceptos-basicos-para-el-desarrollo-sobre-dispositivos-moviles>
- <http://msdn.microsoft.com/es-es/library/6x6bk1f4%28v=VS.80%29.aspx>
- <http://www.slideboom.com/presentations/151355/Exposicion-Dispositivos-Moviles>
- <http://www.google.com.ec/url?sa=t&source=web&cd=4&ved=0CCsQFjAD&url=http%3A%2F%2Fwww.di.uniovi.es%2F~paule%2Fmaterial%2FDispositivosMoviles%2FNETCompactFramework.pdf&ei=LGQaTv-KLM3dgQf9wvUu&usg=AFQjCNH0C1Dae3oNp5JSxovXqj4etM0U4A>

### **Implementación Aplicación Móvil.**

- <http://es.scribd.com/doc/12891188/Tutorial-J2ME-mobile->
- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=wma>
- <http://www.java-samples.com/showtutorial.php?tutorialid=1198>
- <http://onshoresystems.com/smsapi/JavaOne05-BOF9941.pdf>