

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO
MAESTRIA EN CIENCIAS E INGENIERIA



“EVALUACIÓN DE PLATAFORMAS DE DESARROLLO DE
APLICACIONES MÓVILES JAVA ME Y ANDROID PARA LA
IMPLEMENTACIÓN DE SERVICIOS DE SALUD MÓVIL”

Que para Obtener el grado de
Maestra en Ingeniería

Presenta
Mariana Méndez Flores

Director (a) de Tesis
Dra. Mabel Vázquez Briseño

Ensenada B.C.

Agosto 2012

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO
UNIDAD ENSENADA

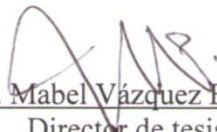
TESIS

**“Evaluación de plataformas de desarrollo de aplicaciones móviles Java ME y
Android para la implementación de servicios de salud móvil”**

Que para obtener el grado de maestría en ingeniería presenta:

Mariana Méndez Flores

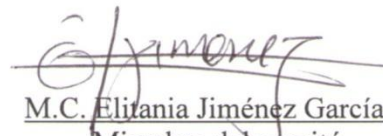
Aprobada por:



Dra. Mabel Vázquez Briseño
Director de tesis



Dr. Juan Iván Nieto Hipólito
Miembro del comité



M.C. Elitania Jiménez García
Miembro del comité

Ensenada Baja California, México. Agosto 2012

RESUMEN

El objetivo de salud móvil es conectar y coordinar a todos los participantes de los procesos ligados a la salud para mejorar la atención médica, incluidos los pacientes, la amplia gama de profesionales de la salud, farmacéuticas, medios y anunciantes y proveedores entre otros. Salud móvil es un avance de Salud Electrónica y gracias a la reducción de costos y la tecnología móvil su presencia se expande en lugares del mundo donde la tecnología tradicional tenía muy difícil acceso.

En esta Tesis se presentan los resultados obtenidos de la evaluación de las plataformas de programación para dispositivos móviles Java ME y Android específicamente para aplicaciones de Salud móvil. Esta evaluación es importante porque se indican los comportamientos de cada plataforma en la comunicación y conexión por Bluetooth y con un Servidor Web, además de presentar las funcionalidades con las que cuentan para el desarrollo de aplicaciones en el área de la salud.

DEDICATORIA

Este trabajo de tesis está dedicado con respeto y amor:

A

MIS PADRES

Juana E. Flores y Gustavo Méndez

**A quienes les debo todo lo que soy ya que han sabido orientarme siempre
con libertad, cariño y confianza.**

AGRADECIMIENTOS

A Dios, por estar a mi lado siempre, haciéndome sentir acompañada cuando más lo necesite, fortaleciéndome para no dejar inconcluso este proyecto que me permitió iniciar.

A mi Esposo Sergio y a mi Hija Sofía, por su apoyo incondicional y por su paciencia para soportar mis cambios ánimos y los días fuertes de trabajo ¡Los AMO!

A mis hermanas Ana María, Roxana, Ana Lilia y Johana porque siempre estuvieron junto a mí cuando las llegue a necesitar, por darme ánimos y hacerme sentir que están orgullosas por este paso más que estoy concluyendo.

A mis inseparables Amigas (ahora Comadres) Edith y Norma, gracias a ellas ingrese a estudiar la Maestría, siempre me apoyaron en todo momento para concluir mi trabajo. ¡Gracias Comadres!

A mi Amiga Dolores, que recorrimos juntas este camino, siempre apoyándonos en todo momento tanto en los estudios como en lo personal. ¡Gracias Amiga!

A mis Amigas Daleth y Eloina, porque sé que siempre me brindan su apoyo, colaboración y cariño sin interés alguno y porque sé que se alegran por este nuevo paso que culmino.

A mis compañeros Víctor, Claudia Rangel y Claudia Fimbres, ahora grandes amigos, por amistad, sus consejos y su apoyo incondicional.

A mi compañero Luis Pérez (que en paz descanse) porque los semestres que me dio oportunidad dios de compartir con él siempre nos estuvimos apoyando mutuamente y fueron de grandes satisfacciones para los dos.

A todos aquellas personas que vivieron conmigo la obtención de este grado de Maestría, por estar ahí en las buenas y en las malas. De corazón les agradezco su Apoyo, Cariño y Amistad.

AGRADECIMIENTOS ESPECIALES

Un gran agradecimiento a la Dra. Mabel Vázquez Briseño por la colaboración, paciencia, apoyo brindados siempre, por la oportunidad de aprender de su capacidad y experiencia científica, por darme la confianza, afecto y amistad necesarios para la creación de este trabajo.

Mi más sincero agradecimiento a mis sinodales Dr. Juan Iván Nieto Hipólito y M.C. Elitania Jiménez Gracia por su disposición y apoyo en la revisión de este trabajo.

A la FIAD por brindarme la posibilidad de seguir estudiando

A CONACyT por el apoyo económico brindado durante mis estudios de posgrados para poder realizar este trabajo que me permite obtener el grado de Maestría.

INDICE GENERAL

CAPÍTULO I.....	13
1. INTRODUCCIÓN	13
1.1 Antecedentes.....	13
1.1 Planteamiento del Problema	15
1.2 Preguntas de Investigación.....	16
1.3 Objetivo General	16
1.4 Objetivos Específicos.....	16
1.5 Metas	17
CAPITULO II	18
2. ESTADO DEL ARTE	18
2.1 Salud Móvil (m-Salud).....	18
2.2 Dispositivos Móviles.....	18
2.2.1 Teléfonos Móviles.....	19
2.2.2 Asistente Digital Personal	19
2.2.3 Computadora Portátil	20
2.3 Sistemas Operativos para Dispositivos Móviles.....	21
2.3.1 Palm OS	22
2.3.2 Symbian	22
2.3.3 BlackBerry OS	23
2.3.4 Windows Mobile.....	23
2.3.5 iOS	23
2.3.6 Android SO.....	24
2.4 Plataformas de programación para Dispositivos Móviles.	24
2.4.1 Plataforma Java Micro Edition.....	24
2.4.2 Plataforma Android.....	26
2.5 ¿Porqué Comparar Java ME y Android, si los dos están basados en Java?	27
2.6 Tecnologías de Comunicación Inalámbrica de corto alcance.....	28
2.7 Resumen.....	30
CAPITULO III.....	31
3. SALUD MÓVIL.....	31
3.1 Introducción	31
3.2 Historia.....	32
3.3 Determinación de los Requerimientos para una Aplicación de Salud Móvil.	32
3.3.1 Aplicaciones de Salud Móvil	34
3.3.2 Establecimiento de Módulos para una Aplicación de Salud Móvil	41
3.4 Resumen.....	42
CAPITULO IV.....	43
4. DISEÑO DE LA APLICACIÓN DEL CASO DE ESTUDIO	43

4.1	<i>Introducción</i>	43
4.2	<i>Diseño de la Aplicación del Caso de Estudio</i>	43
4.3	<i>Métricas de evaluación para las Plataformas Java ME y Android</i>	46
4.4	<i>Resumen</i>	48
CAPITULO V		49
5.	EVALUACIÓN CUANTITATIVA	49
5.1	<i>Evaluación del ESCENARIO 1: conexión y comunicación BT</i>	50
5.2	<i>Evaluación del ESCENARIO 2: conexión y comunicación con el servidor</i>	52
5.3	RESULTADOS de la evaluación del ESCENARIO 1.	54
5.3.1	Tiempos de Conexión Bluetooth	54
5.3.2	Análisis de líneas de código	60
5.4	RESULTADOS de la evaluación del ESCENARIO 2.	62
5.4.1	Tiempos de conexión HTTP	62
5.4.2	Análisis de líneas de código	68
5.5	<i>Resumen</i>	70
CAPITULO VI		71
6.	EVALUACIÓN CUALITATIVA	71
6.1	<i>Evaluación del Escenario 3: aceptación del usuario</i>	71
6.2	RESULTADOS de la evaluación del ESCENARIO 3	72
6.2.1	Aceptación del Usuario	72
6.2.2	Análisis de líneas de Código	77
6.3	<i>Aplicación desarrollada en las plataformas Java ME y Android</i>	77
6.3.1	Interfaz de Usuario en Java	77
6.3.2	Interfaz de Usuario Android	79
6.3.2	Resultados de las Interfaz de Usuarios creadas en las plataformas Java ME y Android.	80
6.4	<i>Resumen</i>	83
CAPITULO VII		84
7.	RECOMENDACIONES	84
7.1	<i>Recomendación del uso de las plataformas Java ME y Android</i>	90
CAPITULO VIII		91
8.	CONCLUSIONES	91
8.1	<i>Trabajo Futuro</i>	92
REFERENCIAS		93
ANEXO 1		98

INDICE DE FIGURAS

Figura 2.1 Dispositivos móviles utilizados en los servicios de salud móvil .	21
Figura 2.2 Arquitectura del entorno de ejecución de la Plataforma Java ME	26
Figura 2.3 Arquitectura del entorno de ejecución de la Plataforma Android	27
Figura 2.4 Clasificación de redes Inalámbricas	29
Figura 4.1 Diseño de una Aplicación de Salud Móvil	44
Figura 4.2 Diagrama General de Iteraciones de la Aplicación	46
Figura 4.3 Diagrama de Métricas de Evaluación	47
Figura 5.1 Escenario 1. Conexión y comunicación Bluetooth entre la Interfaz BT y Emulador/Dispositivo para cada Plataforma	511
Figura 5.2 Escenario 2. Conexión y comunicación del emulador/dispositivo móvil al servidor web.	522
Figura 5.3 Gráfica de los tiempos promedios de conexión entre los emuladores y la interfaz BT	555
Figura 5.4 Gráfica de los tiempos de conexión entre los teléfonos móviles y la interfaz B.	566
Figura 5.5 Gráfica del promedio de tiempo la captura de temperatura a diferentes distancias con los emuladores sin obstáculo	588
Figura 5.6 Gráfica del promedio de tiempo la captura de temperatura a diferentes distancias con los emuladores con obstáculo	588
Figura 5.7 Gráfica del promedio de tiempo la captura de temperatura a diferentes distancias con los teléfonos móviles sin obstáculo	599

Figura 5.8 Gráfica del promedio de tiempo la captura de temperatura a diferentes distancias con los teléfonos móviles con obstáculo.....	60
Figura 5.9 Líneas de código para la conexión BT en Java ME	622
Figura 5.10 Líneas de código para la conexión BT en Android.....	622
Figura 5.11 Gráfica de los tiempos de conexión al servidor con emulador utilizando 2 redes WiFi infinitum.....	633
Figura 5.12 Gráfica de los tiempos de conexión al servidor con los dispositivos móviles utilizando 2 redes WiFi infinitum.....	644
Figura 5.13 Gráfica de los tiempos de lectura al servidor con el emulador utilizando 2 redes WiFi infinitum.....	655
Figura 5.14 Gráfica de los tiempos de lectura al servidor con los teléfonos móviles utilizando 2 redes WiFi infinitum.....	666
Figura 5.15 Gráfica del promedio de tiempos de escritura con el emulador al servidor utilizando 2 redes WiFi infinitum.....	677
Figura 5.16 Gráfica del promedio de tiempos de escritura el teléfono móvil al servidor utilizando 2 redes WiFi infinitum.....	677
Figura 5.17 Líneas de código para la comunicación con el servidor con el paquete <i>javax.mocroedition.io</i> de Java ME	698
Figura 5.18 Líneas de código para la comunicación con el servidor con el paquete <i>org.apache.*</i> de Android.....	699
Figura 5.19 Líneas de código para la comunicación con el servidor con el paquete <i>java.net</i> de Android	699
Figura 6.1 Escenario 3. Aceptación del usuario	71
Figura 6.2 Edad Promedio de los Encuestados	73

Figura 6.3 Escolaridad de encuestados.....	73
Figura 6.4 Tipo de Dispositivo móvil que utilizan.....	74
Figura 6.5 Sistema operativo del dispositivo móvil	74
Figura 6. 6 Estructura de la interfaz de usuario en Java ME.....	78
Figura 6.7 Estructura de la interfaz de usuario en Android.....	80
Figura 6.8 Pantalla Principal en la Plataforma Java ME	81
Figura 6.9 Menús creados en Java ME.....	81
Figura 6.10 Resultados de las opciones del menú.....	81
Figura 6.11 Pantalla principal en la plataforma Android	82
Figura 6.12 Ajuste/ captura de datos del paciente	82
Figura 6.13 Resultados obtenidos del menú.....	83

INDICE DE TABLAS

Tabla 3.1 Aplicaciones de Salud Móvil.....	407
Tabla 5.1 Resultados de promedios de tiempos conexión entre emulador/teléfono móvil y la interfaz BT.....	54
Tabla 5.2 Tiempo promedio de lecturas de temperaturas entre el dispositivo móvil y la interfaz sin obstáculo.	55
Tabla 5.3 Tiempo promedio de lecturas de temperaturas en milisegundos entre el dispositivo móvil y la interfaz con obstáculos.....	57
Tabla 5.4 Promedio de intentos de conexión entre el dispositivo móvil y el sensor médico.....	60
Tabla 5.5 Líneas de código para conexión Bluetooth.....	61
Tabla 5.6 Tiempos promedio de conexión al servidor web.....	63
Tabla 5.7 Tiempo promedio de lectura al servidor Web	65
Tabla 5.8 Tiempo promedio de escritura al servidor Web.	66
Tabla 5.9 Líneas de código para la comunicación al servidor.....	68
Tabla 6.1 Resultados de Encuesta.....	76
Tabla 6.2 Tamaño de la aplicación en kilobytes.....	77
Tabla 7.1 Herramientas para Módulos de una aplicación de Salud Móvil.....	88
Tabla 7.2 Indicadores para el Usuario Programador	89

CAPÍTULO I

1. INTRODUCCIÓN

La reciente evolución de las tecnologías telefónicas móviles (y sus resultados en cuestiones de movilidad en diferentes aplicaciones de telemedicina y tele asistencia) se reflejan claramente en el creciente dominio comercial de los servicios móviles de telemedicina. Los servicios móviles NO eran posibles con la norma convencional de sistemas telefónicos antiguos y la Red Digital de Servicios Integrados. Los ejemplos actuales incluyen: transmisiones ECG (electrocardiograma), imágenes de vídeo y tele radiología; servicios inalámbricos para predecir la morbilidad de emergencia y accidentes cerebrovasculares (derrame cerebral), y otros sistemas de control integrado de telemedicina móvil (Istepanian et al, 2003).

Salud móvil es un nuevo paradigma que reúne a las nuevas tecnologías de comunicaciones inalámbricas y de red con el concepto de “Asistencia Sanitaria Conectados en cualquier Momento y en cualquier Lugar” (Istepanian et al, 2009).

1.1 Antecedentes

Una de las aplicaciones más frecuentes de Salud Móvil (m-salud) es el monitoreo para diagnóstico y seguimiento de pacientes. Esta aplicación requiere, habitualmente, el uso de diversos sensores, encargados de la recuperación de datos, obteniendo información del estado de variables vitales para posteriormente transmitirla para su procesamiento, utilizando un dispositivo móvil como medio de comunicación.

Las Redes Inalámbricas de Área Corporal, representan una tecnología emergente para la integración de sistemas de m-salud, que se constituyen de un conjunto de plataformas de comunicación caracterizadas por el uso de sensores diminutos, ágiles en la transmisión de datos y considerablemente económicos. Cada plataforma de comunicación está constituida por uno o más Sensores Fisiológicos (SF). Dichos SF se refieren a: sensores de movimiento, electrocardiógrafos, electroencefalogramas, etc. Los SF pueden colocarse en el cuerpo como pequeños parches inteligentes, integrados en la ropa o implantados bajo la piel o los músculos (Jovanov E. , 2006).

Cuando se utiliza una plataforma de red, para recolección de datos, la información recabada puede ser enviada a dispositivos móviles con el fin de ser analizada y realizar toma de decisiones a partir de las variables capturadas. Para lograr esto, es necesario el uso de plataformas de programación para estos tipos de dispositivos, que permitan la distribución rápida de información, pero sobre todo que su sistema sea lo más ligero posible, debido a que este tipo de dispositivos presentan características limitadas en cuanto a almacenamiento y memoria.

Existen plataformas para el desarrollo de aplicaciones para dispositivos móviles como: *LiMo (Linux Mobile)*, *OpenMoko*, *Mac OSX*, *MOTOMAGX*, *Java ME*, *.net Microsoft*, *Access Linux Platform (ALP)*, *Android*, *ARM Linux Mobile Platform*, *BlackBerry*, *Qtopia*. Cada una de ellas cuenta con diferentes características (Aparicio, 2009).

De las anteriores podemos destacar a: Java ME y Android dentro de las más utilizadas. La primera plataforma al momento de realizar esta tesis es la más utilizada en la actualidad por los teléfonos celulares, según estudios (EDC, 2005) el 56% de los desarrolladores de aplicaciones móviles prefieren su utilización para garantizar compatibilidad entre diversos dispositivos; y más de la mitad de los desarrolladores, 52%, utiliza las herramientas de código abierto y Linux para el desarrollo de este tipo de aplicaciones (Liu, Ming-lu, & Lin, 2008), por lo cual se decidió por Android, además de ser una plataforma de creación reciente y promete ser la competencia de Java ME. Estas plataformas se describen más a detalle en el capítulo 2

1.2 Justificación

A partir del crecimiento en el uso de dispositivos móviles, la salud móvil ha evolucionado a nivel mundial, son cada vez más las aplicaciones disponibles en el área de cuidado de la salud.

El uso de aplicaciones móviles pueden suponer un cierto cambio para el personal sanitario (médicos, enfermeros u otros asistentes), ya que puede facilitar el diagnóstico remoto y la recopilación de datos médicos de pacientes, y disminuir los problemas de conectividad que se generan al no tener redes fijas desplegadas globalmente. Así, el uso de esta tecnología

puede ser eficiente para: a) Acceder a registros de datos, b) Localización de epidemias y casos de emergencia y c) Análisis, diagnóstico y consulta; todo esto empleando aplicaciones diseñadas para introducir, acceder, enviar y recibir los datos de los pacientes en teléfonos móviles, tales como: PDAs, *Smartphone* o computadoras portátiles.

La disponibilidad de información, la velocidad en el procesamiento de datos, el rendimiento de sistemas de cómputo y su convergencia con tecnologías de red inalámbricas acelerará el desarrollo de sistemas de salud móvil, esto es, el desarrollo de sensores y aplicaciones inteligentes enfocados a temáticas de impacto en la salud, permitirá reestructurar los mecanismos y rutas actuales de atención médica, ofertándolos a disponibilidad de todos, independientemente del lugar donde vivan ó trabajen (Istepanian et al, 2004). El reto es lograr la implementación de estos sistemas con servicios realmente aplicables.

Hoy en día se cuenta con tecnologías de red inalámbricas que permiten ofrecer a los profesionistas de la salud acceso a registros de pacientes desde cualquier punto dentro de la Intranet de la Institución, esto permite un mayor soporte a la movilidad, así como rapidez y control en la gestión de la información, es decir, se ofrecen servicios como: Acceso a historial médico del paciente, resultados de laboratorio, datos farmacéuticos, información de seguro, etc. Para el desarrollo de estas aplicaciones existen varias plataformas de programación, sin embargo no existen investigaciones dirigidas al análisis funcional en esta área; actualmente no se cuenta con una comparativa que determine la plataforma de programación ideal para el desarrollo de aplicaciones de salud móvil. Esto ofrece la posibilidad de una investigación en esta área.

1.1 Planteamiento del Problema

Aun cuando ya existen algunas investigaciones sobre las características de las plataformas para el desarrollo de aplicaciones móviles; no se cuenta con un análisis específico de las características y funcionalidades de estas plataformas que permita determinar a partir de métricas (características de soporte, manejo de herramientas, funcionalidad, conectividad, etc.) la elección de una plataforma para el desarrollo de una aplicación de Salud Móvil. En esta investigación se evaluarán particularmente las plataformas de programación Java ME y Android como herramientas para el desarrollo de

aplicaciones orientadas a Salud Móvil; debido a que Java ME es una plataforma que permite desarrollar aplicaciones que se pueden ejecutar sobre diversos sistemas operativos, Android por su parte es una plataforma de reciente creación de la cual se espera en México, que al igual que en Estados Unidos, sea una de las más utilizadas en los dispositivos móviles. Otro factor importante en la elección de estas plataformas es que las dos son de código libre o abierto, lo cual permite desarrollar aplicaciones sin tener que comprar programas o entornos de desarrollo.

1.2 Preguntas de Investigación

- ¿Cuáles son las características que debe contener una aplicación de servicios de salud móvil?
- ¿Cuales son las métricas adecuadas en escenarios de Evaluación de Plataformas Móviles?
- ¿Existe algún (os) prototipo (s) para el desarrollo de aplicaciones de salud móvil?
- ¿Cuáles son las funcionalidades de la plataforma Java ME y Android para el desarrollo de aplicaciones de salud móvil?
- ¿Es posible definir entre las plataformas de programación móvil Java ME y Android cual es la ideal para el desarrollo de aplicaciones de salud móvil?

1.3 Objetivo General

Realizar estudio comparativo para determinar las funcionalidades de las plataformas Java ME y Android específicamente útiles para aplicaciones de Salud Móvil.

1.4 Objetivos Específicos

Con el fin de lograr el objetivo general se plantean los siguientes objetivos específicos:

- Definir los Módulos para los Sistemas de Salud Móvil.
- Definir las funcionalidades específicas de cada plataforma para la implementación de aplicaciones de Salud Móvil.
- Definir las métricas que permitan la evaluación y comparación de las plataformas
- Desarrollar un prototipo para el servicio de Salud Móvil, en ambas plataformas.

- Evaluar los prototipos desarrolladas en ambos plataformas.
- Integrar los lineamientos, limitantes y posibilidades de cada plataforma para el desarrollo de aplicaciones de salud móvil.

1.5 Metas

- Desarrollar una aplicación para servicios de salud móvil en las plataformas Java ME y Android que contenga las características que debe proveer este tipo de aplicaciones.
- Establecer conectividad por medio de Bluetooth con los sensores médicos para capturar los datos para la aplicación en las plataformas Java ME y Android.
- Definir las funcionalidades específicas de cada plataforma para la implementación de aplicaciones de Salud Móvil.

CAPITULO II

2. ESTADO DEL ARTE

2.1 Salud Móvil (m-Salud)

Nacido de una definición técnica “e-med sin cables”, *m-health* o salud móvil (m-salud) ha sido considerado como una convergencia de e-salud y los sistemas de telemedicina. En general m-salud implica la informática móvil, tecnologías de la comunicación y sensores médicos para sistemas de cuidado de la salud.

M-salud recompone los servicios de salud y da nueva forma a los mecanismos subyacentes a través de sensores inteligentes, dispositivos de administración de fármacos, redes de área personal inalámbricas (WPAN), WLAN, WAN, etc. (Liu et al, 2008).

2.2 Dispositivos Móviles

Es significativo el papel que en nuestra sociedad desempeñan los dispositivos móviles hoy en día. La idea de movilidad se ha impregnado en nuestros hábitos cotidianos, tanto los referentes al trabajo como a la vida privada.

Un dispositivo móvil se puede definir como un aparato de tamaño pequeño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones generales (Baz et al, 2010). De acuerdo con esta definición existen multitud de dispositivos móviles, a continuación se mencionan algunos tipos de dispositivos móviles:

- Teléfono Inteligente
- Teléfono Inalámbrico
- Videoconsola portátil
- Reproductor de audio portátil
- PDA (*Personal Digital Assistant*)
- Cámara digital

- Cámara de vídeo
- Computadora portátil
- IPAD

De los dispositivos anteriormente mencionados solo algunos son utilizados en los servicios de Salud Móvil, a continuación se describen:

2.2.1 Teléfonos Móviles

El teléfono móvil es un dispositivo inalámbrico electrónico basado en la tecnología de ondas de radio, que tiene la misma funcionalidad que cualquier teléfono de línea fija. Su principal característica es su portabilidad, ya que la realización de llamadas no es dependiente de ningún terminal fijo y no requiere ningún tipo de cableado para llevar a cabo la conexión a la red telefónica. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado funciones adicionales como mensajería instantánea (SMS), agenda, juegos, cámara fotográfica, acceso a Internet, reproducción de video e incluso GPS y reproductor mp3.

Teléfono Inteligente o Smartphone (En Inglés) es un dispositivo electrónico que funciona como un teléfono celular con características similares a las de una computadora personal. Casi todos los teléfonos inteligentes son móviles que soportan completamente un cliente de correo electrónico con la funcionalidad completa de un organizador personal. Una característica importante de casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero (Baz et al, 2010).

2.2.2 Asistente Digital Personal

Un Asistente Digital Personal mayormente conocido por sus siglas en inglés como PDA (*personal digital assistant*), es una computadora de mano originalmente diseñada como agenda electrónica con un sistema de reconocimiento de escritura. En 2009 estos dispositivos, ya podían realizar muchas de las funciones de una computadora de escritorio pero con la ventaja de ser portátil.

Inicialmente los PDAs incluían aplicaciones estrictamente relacionadas con su función como agenda electrónica, es decir, se reducían a calendario, lista de contactos, bloque de notas y recordatorios. Con el paso de tiempo han ido evolucionando hasta los dispositivos actuales que ofertan un rango mucho más extendido de aplicaciones, como juegos, acceso al correo electrónico o la posibilidad de ver películas, crear documentos, navegar por Internet o reproducir archivos de audio.

Las características del PDA moderno son pantalla sensible al tacto, conexión a una computadora para sincronización, ranura para tarjeta de memoria, y al menos Infrarrojo, Bluetooth o WiFi.

La irrupción de Microsoft Windows CE (2000) y Windows Mobile (2003) en el sector los dotó de mayores capacidades multimedia y conectividad. Las PDAs de hoy en día incluyen multitud de comunicaciones inalámbricas (Bluetooth, WiFi, IrDA, GPS,) que los hace tremendamente atractivos hasta para cosas tan inverosímiles como su uso para domótica o como navegadores GPS (Baz et al, 2010).

2.2.3 Computadora Portátil

Computadora portátil (también ***laptop***, término proveniente del inglés) es una pequeña computadora personal móvil, que pesa normalmente entre 1 y 3 Kg. Las computadoras portátiles son capaces de realizar la mayor parte de las tareas que realizan las computadoras de escritorio, con la ventaja de que son más pequeñas, más livianas y tienen la capacidad de operar sin estar conectadas a la electricidad por un período determinado.



Figura 2.1 Dispositivos móviles utilizados en los servicios de salud móvil (Yoorah, 2008).

Como se puede ver en la figura 2.1 se pueden utilizar varios dispositivos móviles para los servicios de salud móvil, sin embargo los teléfonos inteligentes o teléfonos celulares actualmente son una parte indispensable en nuestras vidas, es el dispositivo móvil que está teniendo mayor éxito en el mercado, debido al precio accesible comparado con las PDA's o Computadoras portátiles. Por tal motivo este trabajo de investigación está centrado con este tipo de dispositivos móviles.

2.3 Sistemas Operativos para Dispositivos Móviles.

Partiendo de la definición de sistema operativo: Capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo.

Existen multitud de opciones, las más extendidas son: Palm OS, Symbian, BlackBerry OS, Windows Mobile, y recientemente iOS y el sistema operativo de Google, Android, además por supuesto de los dispositivos con sistema operativo Linux. Las características básicas de cada uno son las siguientes:

2.3.1 Palm OS

Palm OS es un sistema operativo propietario destinado a dispositivos móviles, más específicamente a PDAs (Personal Digital Assistant). Palm OS comenzó su desarrollo en 1996 y Palm Inc. comenzó a licenciarlo en diciembre de 1997 con sus novedosos aparatos PalmPilot.

Las principales características de este SO son: Hardware altamente integrado con el SO, sistema operativo de 32 bits con arquitectura adaptable en escala, modular y basada en estándares, permite la ejecución de múltiples hilos de ejecución; Usaba un monitor monocromático; preferible antes que implementar los colores de manera pobre; Pocas funciones del SO, se centraba sobretodo en la usabilidad. Estaba diseñado para ser una herramienta práctica, no un sistema orientado a personas con conocimiento informático (Earl, 2008).

2.3.2 Symbian

Este es el sistema operativo para móviles más extendido entre “Smartphones”, y por tanto el que más aplicaciones para su sistema tiene desarrolladas. Actualmente Symbian ocupa más del 65% del mercado de sistemas operativos.

Su principal virtud es la capacidad que tiene el sistema para adaptar e integrar todo tipo de aplicaciones. Admite la integración de aplicaciones y, como sistema operativo, ofrece las rutinas, los protocolos de comunicación, el control de archivos y los servicios para el correcto funcionamiento de estas aplicaciones. La tecnología del sistema operativo Symbian se ha diseñado teniendo en cuenta puntos clave como el poder proporcionar la energía, memoria y gestión de entrada y salida de recursos requeridos específicamente en los dispositivos móviles. También, supone una plataforma abierta, ésta es la clave, que asocia a las telecomunicaciones y los estándares globales de internet.

Los usuarios de Symbian señalan como principal ventaja del sistema el hecho de que exista una amplia selección de aplicaciones disponibles para todo tipo de teléfonos móviles. Destacan también la compatibilidad con los estándares de conectividad y redes como Bluetooth, WiFi, GSM, GPRS, CDMA y WCDMA (Baz et al, 2010).

2.3.3 BlackBerry OS

BlackBerry es un sistema operativo multitarea que está arrasando en la escena empresarial, en especial por sus servicios para correo y teclado QWERTY. Actualmente BlackBerry OS cuenta con un 11% del mercado.

BlackBerry aparece en el mercado justo en el momento en que comenzaba a demandarse un sistema operativo que permitiera utilizar de una forma fácil, cómoda y rápida los servicios de correo electrónico. Hoy en día es también proveedor de servicios de correo electrónico a dispositivos que no son BlackBerry, gracias al programa BlackBerry Connect (Baz et al, 2010)

2.3.4 Windows Mobile

Windows Mobile es un SO de la familia Windows CE, desarrollado por Microsoft. A pesar de llevar el nombre Windows, no es un sistema derivado ni es una versión recortada del mismo, sino que es un nuevo sistema diseñado específicamente para dispositivos móviles.

Este sistema, está estrechamente vinculado a otros productos de la misma marca (servicios Live, Office Mobile, Internet Explorer Mobile, etc.) y cuenta con una interfaz gráfica de muy buena calidad, y muy similar a la de los sistemas operativos Windows.

2.3.5 iOS

iOS comienza conjuntamente con el nacimiento del conocido iPhone, en el 2007. Aunque, esta aseveración es discutible, ya que este sistema operativo que corre en el Iphone es en realidad una versión adaptada del OS X. Por lo cual, en este sentido, este sistema ya tiene años en el mercado y ha sido puesto a prueba.

El sistema ha sido adaptado, removiendo todos los componentes que no son críticos para un dispositivo móvil, y se le adicionan funcionalidades que si están relacionadas con el mundo de la telefonía móvil (Earl, 2008).

2.3.6 Android SO

Android SO es el más reciente de los sistemas operativos para móviles del mercado. Es un sistema operativo para dispositivos móviles como teléfonos inteligentes y tabletas basado en el núcleo Linux. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google.

Alguna de sus características son: posee un navegador integrado, el cual está basado en el motor de código abierto WebKit; Contiene gráficos optimizados, como lo son los gráficos 2D y 3D; almacenamiento de datos estructurados, tecnología GSM, la cual es dependiente del Hardware Además de Wi-Fi, Bluetooth, 3G y EDGE, también dependientes del Hardware. Permite Cámaras, GPS, acelerómetro, brújula. Entre muchas más (Developers, 2010).

2.4 Plataformas de programación para Dispositivos Móviles.

Como se mencionaba en el capítulo anterior existen varias plataformas de programación para dispositivos móviles cada uno de ellos con características particulares; a continuación se describen las plataformas Java Micro Edition (JME) y Android.

2.4.1 Plataforma Java Micro Edition

Esta Plataforma es un subconjunto de la plataforma Java estándar, que proporciona una colección certificada de API's (Interface de Programación de Aplicaciones) Java para el desarrollo de aplicaciones para dispositivos con recursos limitados como teléfonos celulares, PDA, lectores de Blu-Ray, impresoras, entre muchos más (Oracle, 2010).

Características

Java ME se ejecuta sobre una máquina virtual basada en núcleo conocida como KVM (por sus siglas en inglés kilobyte Virtual Machine), es compatible con el desarrollo multiplataforma a través de configuraciones y perfiles.

- Una Configuración define las características mínimas de la máquina virtual de Java y un conjunto de bibliotecas para una familia horizontal de dispositivos, es decir,

dispositivos similares en limitaciones de procesamiento y memoria, los requisitos de interfaz de usuario y capacidades de conexión (Galvez & Ortega, 2007).

- Un archivo *profile* o perfil contiene bibliotecas específicas con las características únicas para una clase particular de dispositivo.

Todos los dispositivos que tienen habilitado Java ME son compatibles específicamente con la siguiente configuración y Perfil:

1. CLDC (Configuración Limitada de Dispositivos Conectados / Connected Limited Device Configuration), es un marco para las aplicaciones Java ME dirigidas a los dispositivos con recursos mínimos. CLDC contiene un subconjunto estricto de librerías de las clases de Java necesarias para el desarrollo de aplicaciones móviles (Gavalas et al, 2011).
2. MIDP (Perfil de Información de Dispositivos Móviles / Mobile Information Device Profile), es una especificación para el uso de Java embebido en los dispositivos, tales como teléfonos móviles y PDA's. MIDP es parte de la plataforma Java ME y se sitúa en la parte superior del CLDC
3. Las Aplicaciones Java ME son desarrolladas sobre CLDC/MIDP, denominados como MIDlets y por lo general son empaquetados en un archivo Java llamado JAR (Oracle, 2010).

En la figura 2.2 se muestra la distribución del esquema de ejecución del entorno completo de la plataforma Java ME

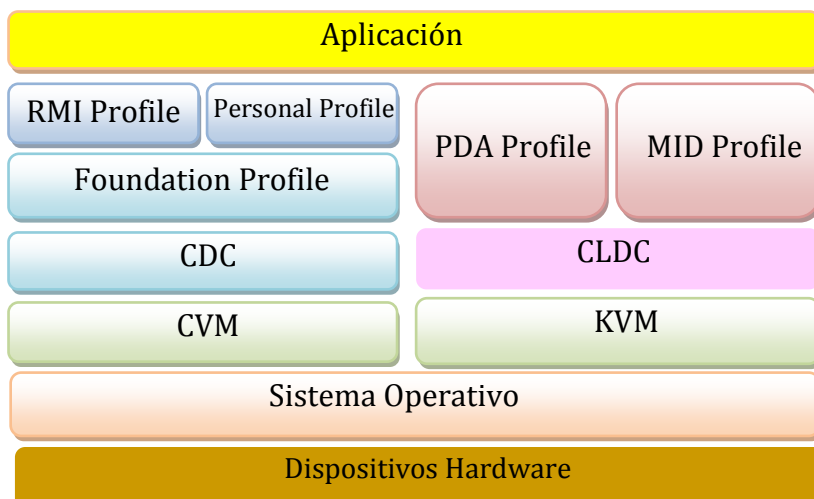


Figura 2. 2 Arquitectura del entorno de ejecución de la Plataforma Java ME (**Gálvez et al, 2007**).

2.4.2 Plataforma Android

La plataforma Android fue lanzada por Google en 2007, para avance en los estándares abiertos para dispositivos móviles. Android es una plataforma con licencia de código abierto para dispositivos móviles basados en Linux. Su pila de software para dispositivos móviles incluye un sistema operativo, un middleware y aplicaciones clave (Google, 2008).

Características

Las aplicaciones Android son principalmente escritas en Java y compiladas en formato ejecutables Dalvik (DEX). Cada aplicación ejecuta su propio proceso, con su propia máquina virtual Dalvik. Dalvik ejecuta archivos DEX, que se convierten en tiempo de compilación de la clase estándar y los archivos JAR. DEX archivos son más compactos y eficientes que los archivos de clase.

Los desarrolladores tienen acceso completo a todos los marcos y las API que las aplicaciones básicas y las librerías de desarrollo de Google. La arquitectura de Android está diseñada para simplificar la reutilización de componentes. Cualquier aplicación puede publicar sus capacidades, y cualquier otra aplicación puede utilizar esas capacidades, sujetas a las restricciones de seguridad impuestas por el marco (Gavalas et al, 2011).

El kit de software de desarrollo de Android (SDK) es compatible con aplicaciones de creación con una gran funcionalidad. Al igual que el iPhone que puede manejar las pantallas táctiles, acelerómetros, gráficos 3D y GPS, así como la colaboración entre las aplicaciones como correo electrónico, mensajería, calendarios, las redes sociales y servicios basados en ubicación.

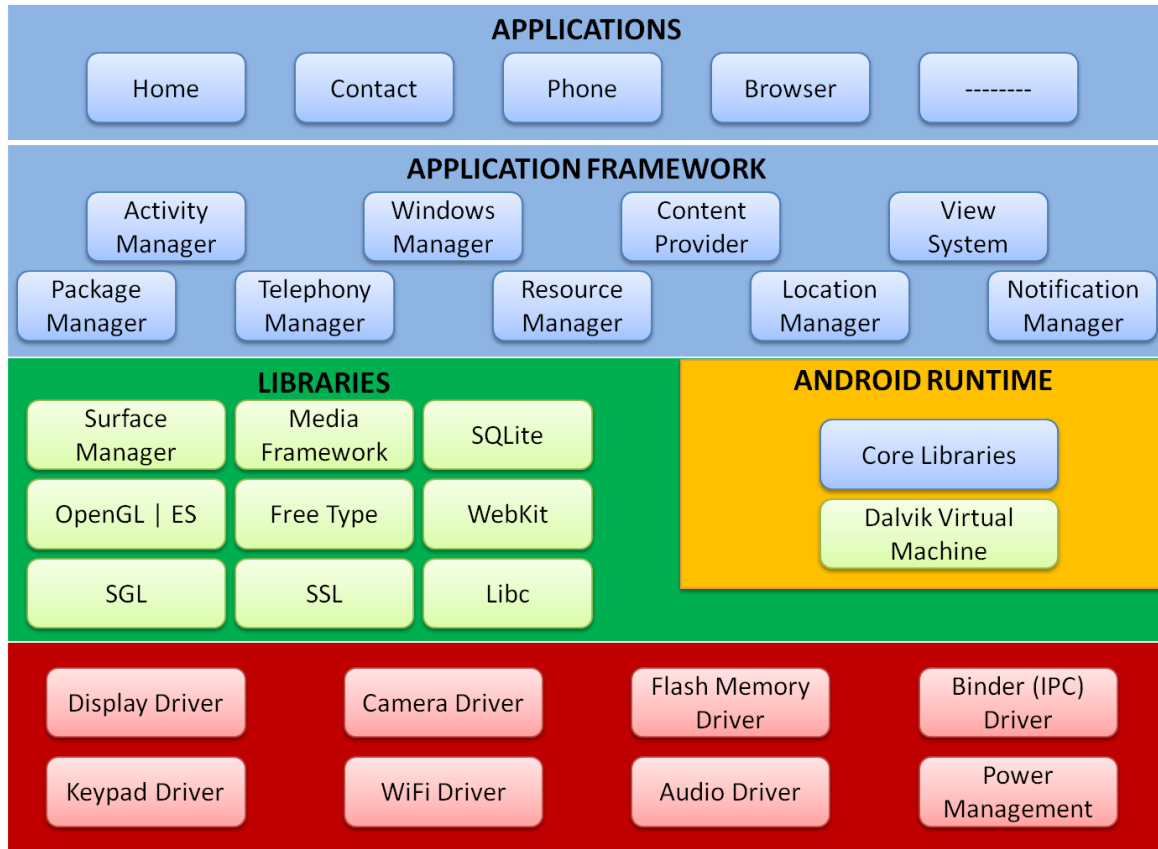


Figura 2. 3 Arquitectura del entorno de ejecución de la Plataforma Android (**Developers, 2010**)

2.5 ¿Porqué Comparar Java ME y Android, si los dos están basados en Java?

El motivo de la comparación de estas plataformas es debido a como se menciona en el capítulo 1, porque Java ME es hasta el momento la plataforma más utilizada en diferentes equipos móviles, sin embargo Android, la cual es una plataforma de muy reciente creación, es un fuerte competidor de la misma, proporcionando un entorno basado en Java, pero con diferencias notables. La plataforma Java ME esta basada en una maquina virtual totalmente Java (figura 2.2) la cual se trata de una implementación reducida y especialmente orientada a dispositivos con bajas capacidades de memoria, mientras que la maquina virtual de Android es la llamada Dalvik (figura 2.3) diseñada específicamente para el sistema operativo Android, la cual a diferencia de java no esta basada en una pila. Esto también hace imposible la ejecución o migración de una aplicación desarrollada en Android en un equipo con solo Java, sin embargo las aplicaciones creadas en Java Me si pueden ser

ejecutadas en un equipo Android siempre y cuando se agregado un soporte J2ME mediante aplicaciones de terceros como el J2ME MIDP Runner (Netmite, 2011).

Además cada plataforma presenta su propio ciclo de vida en sus aplicaciones y manejan librerías y API's propias, que son mencionadas en los siguientes capítulos.

2.6 Tecnologías de Comunicación Inalámbrica de corto alcance

Las redes inalámbricas se basan en un enlace que utiliza ondas electromagnéticas (radio e infrarrojo) en lugar de cableado estándar. Hay muchas tecnologías inalámbricas que se diferencian por la frecuencia, el alcance y la velocidad de transmisión que utilizan.

Las redes inalámbricas permiten que los dispositivos remotos se conecten sin dificultad, ya sea que se encuentren a unos metros de distancia como a varios kilómetros. Asimismo, la instalación de estas redes no requiere de ningún cambio significativo en la infraestructura existente como pasa con las redes cableadas. Tampoco hay necesidad de perforar las paredes para pasar cables ni de instalar portacables o conectores. Esto ha hecho que el uso de esta tecnología se extienda con rapidez (Kioskea.net, 2008).

Los Teléfonos celulares son parte de la gran ola de dispositivos conectados a internet de nueva generación. Lo que ha llevado al diseño de nuevas tecnologías y redes de acceso inalámbricas. La figura 2.4 muestra la clasificación de las redes inalámbricas de acuerdo a su área geográfica desde que el usuario se conecta a la red.

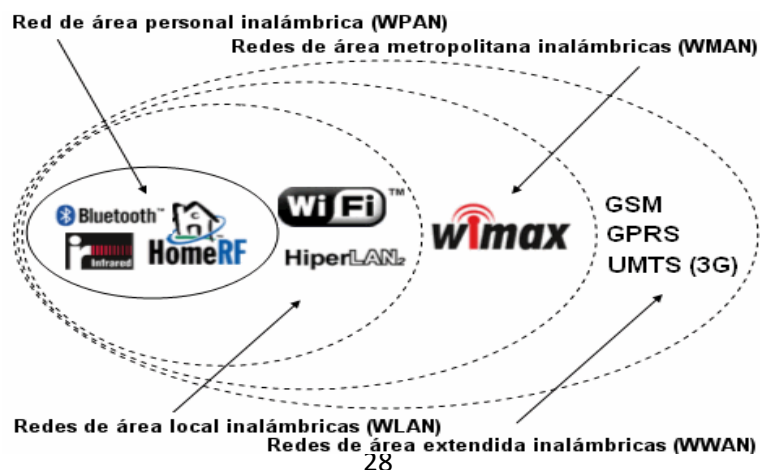


Figura 2. 4 Clasificación de redes Inalámbricas (**Kioskea.net, 2008**)

Dentro de cada tipo de red inalámbrica que muestra la figura 2.4 se encuentra diferentes tecnologías, en esta investigación se trabajara con las tecnologías de las redes WPAN.

WPAN o Red de Área Personal, son redes que enfocan sus sistemas de comunicación a una área típica de 10 metros a la redonda que envuelve a una persona o un dispositivo ya sea que este en movimiento o no. A diferencia de una WLAN, una conexión hecha a través de una red WPAN, involucra muy poca o nula infraestructura o conexiones directas hacia el mundo exterior. Este tipo de tecnología también procura hacer un uso eficiente de los recursos, por lo que se han diseñado protocolos simples y lo más óptimo para cada necesidad de comunicación y aplicación (Candolfi Arballo, 2008).

Entre las tecnologías WPAN más utilizadas y disponibles en el mercado se encuentra el estándar IEEE 802.15.1 Bluetooth, el estándar IEEE 802.15.4 ZigBee y HomeRF (Home Radio Frequency).

La tecnología inalámbrica Bluetooth (IEEE 802.15.1) es un sistema de comunicaciones de corto alcance con velocidad de transmisión de 1 a 3 Mbps, cuyo objetivo es eliminar los cables entre dispositivos electrónicos, tanto portátiles como fijos, manteniendo cierto nivel de seguridad. Esta tecnología permite a los usuarios conexiones instantáneas. Otro objetivo de BT es crear conexiones bajo la primicia de fiabilidad, bajo consumo y mínimo costo (Bluetooth SIG, 2011).

ZigBee (ZB) es un estándar inalámbrico emergente para la transmisión de menor tráfico de datos en la red y aplicaciones de muy bajo consumo de energía; transmitiendo a 250 kbps (Istepanian, Jovanov, & Zhang, 2004). Potencialmente tiene aplicaciones en Domótica, Control Industrial y Cuidados Sanitarios Personales (Fernández et al, 2009).

HomeRF (*Home Radio Frequency*), lanzada en 1998 por HomeRF Working Group (que incluye a los fabricantes Compaq, HP, Intel, Siemens, Motorola y Microsoft, entre otros) ofrece una velocidad máxima de 10 Mbps con un alcance de 50 a 100 metros sin

amplificador. A pesar de estar respaldado por Intel, el estándar HomeRF se abandonó en enero de 2003, en gran medida porque los fabricantes de procesadores empezaron a usar la tecnología Wi-Fi en placa (por medio de la tecnología *Centrino*, que incluía un microprocesador y un adaptador Wi-Fi en un solo componente).

2.7 Resumen

En este capítulo se definieron las bases sobre lo que se debe conocer para el desarrollo de aplicaciones móviles en el sector salud, como principio se describe el concepto de salud móvil, además se describe qué es un dispositivo móvil y los tipos de dispositivos existentes. De igual modo se presentan los sistemas operativos que existen para estos dispositivos, así como las plataformas para el desarrollo de aplicaciones móviles. Por último se describen las tecnologías de comunicación inalámbricas de corto alcance. En el siguiente capítulo se establecen los requerimientos de una aplicación para servicios de salud móvil, el diseño de la aplicación móvil que se utilizara como caso de estudio, así como las métricas de evaluación que nos servirán para evaluar las plataformas Java ME y Android.

CAPITULO III

3. SALUD MÓVIL

3.1 Introducción

Salud Móvil es un término de reciente creación comúnmente definido como la práctica de la salud a través del uso de dispositivos móviles (Anta, Shireen, & Guiffida, 2009), si juntamos esta definición con lo se veía en el capítulo 2 podemos decir entonces que un sistema de salud móvil “nos proporcionan servicios médicos a distancia desde cualquier lugar y en cualquier momento sin la necesidad de permanecer en un lugar específico”¹, ya que la salud móvil hace uso de la informática móvil, sensores médicos y tecnologías de la comunicación.

Si bien es sencilla la definición anteriormente mencionada, es importante citar la definición reconocida de Istepanian, primer científico que acuño esta frase, como “Comunicaciones móviles emergente y tecnologías de red para la asistencia sanitaria”².

La salud móvil tiene varias facetas distintas: la posibilidad de la comunicación casi en tiempo real, el intercambio bidireccional de información, la posibilidad de acceso remoto, que salta la barrera de la distancia entre el paciente y el médico, y la ampliación del sector salud con la aparición de nuevos roles y actores no tradicionales, tales como los operadores de telefonía celular, que pueden llegar a participar de diferentes formas innovadoras en la cadena de valor para desarrollar o extender los servicios de salud (Anta et al, 2009). En la actualidad, las aplicaciones de salud móvil incluyen el uso de PDA para la recopilación de datos de salud, el uso del teléfono celular para transmitir información de salud a los médicos y para el monitoreo en tiempo real, tanto de pacientes como no pacientes (Anta et al, 2009).

¹ Pindter, J., González, V., & Tovar, C. (2009). Proposal for an m-Health System. *CERMA '09*, (pp. 55-59). 1 de Diciembre 2009.

² Istepanian, R. S., Laxminarayan, S., & Pattichis, C. S. (2005). *Topic in Biomedical Engineering International Books Series: M-Health: Emerging Mobile Systems*. Piscataway New Jersey: Springer.

3.2 Historia

Como se describió en el capítulo 1, Salud móvil (m-salud) surge dentro del área de telemedicina con la finalidad de aprovechar los avances de la tecnología.

El mundo está siendo testigo de una de las transformaciones en salud más radicales de la historia. En los últimos años, los temas de salud dejaron de ser dominio exclusivo de los expertos. De hecho, la salud ha pasado a ocupar un lugar central en los ámbitos más relevantes de la agenda global: el desarrollo económico, la seguridad nacional, la democracia y los derechos humanos. Debido a sus vastas implicaciones, los objetivos en materia de salud figuran de manera tan prominente en los Objetivos de Desarrollo del Milenio (Frenk, 2011).

El campo de salud móvil tiene una aplicación para los países industrializados, sin embargo en los últimos años, se ha convertido en una solicitud de los países en desarrollo, derivadas del rápido aumento en la penetración de la telefonía móvil en los países de bajos ingresos. El campo surge en gran medida como un medio de proporcionar mayor acceso a segmentos más amplios de la población en los países en desarrollo, así como la mejora de la capacidad de los sistemas médicos de calidad (Céspedes, 2011).

3.3 Determinación de los Requerimientos para una Aplicación de Salud Móvil.

La tecnología móvil es una oportunidad para el incremento en la atención de la salud en todo el mundo y, sobre todo, para los países del tercer mundo (en vías de desarrollos según la denominación actual). Las Naciones Unidas y la Fundación de Vodafone trabajaron en un proyecto donde se recogieron más de 50 iniciativas para la utilización de la tecnología móvil en aplicaciones de salud en 26 países, entre ellos, la India con 11 iniciativas o Sudáfrica y Uganda con seis cada uno de ellos (Consulting, 2009). En México también se está trabajando en el desarrollo de aplicaciones de este tipo de las cuales se pueden resaltar 3 aplicaciones: VIDANET, CARDIONET y DIABEDIARIO, las cuales están desarrolladas por Voxiva (Compañía privada proveedor global de soluciones móviles de información y una de las primeras empresas en utilizar el término de salud móvil en aplicaciones para teléfono celular (VOXIVA, 2011) en asociación con el Instituto de Salud Carso. En base a

estas iniciativas, se realizaron la clasificación de las aplicaciones de m-salud de la siguiente manera:

- **Educación y Conciencia:** Las aplicaciones de educación y conciencia, son aplicaciones donde los mensajes SMS se envían directamente a los teléfonos de los usuarios para ofrecer información sobre los métodos de prueba y tratamientos, la disponibilidad de servicios de salud y el manejo de la enfermedad (Istepanian et al, 2005).
- **Recolección de Datos Remotos:** La recolección de datos es otro componente fundamental de los programas de salud pública. En el mundo en desarrollo, la recolección de información de campo es particularmente importante, ya que muchos segmentos de la población rara vez pueden visitar un hospital, incluso en el caso de una enfermedad grave. La recopilación de datos de pacientes es vital, y la información debería ser idealmente actualizada y accesible sobre una base de datos en tiempo real. El proceso de recopilación de datos es más eficiente y confiable si se realiza a través de los teléfonos inteligentes, PDAs, o teléfonos móviles en lugar de las encuestas en papel que deben presentarse en persona y manualmente capturar la información en la base de datos central de salud (Consulting, 2009).
- **Monitorización Remota:** Una de las zonas idóneas para crecer a la par con la tecnología móvil es la monitorización remota de pacientes. El control remoto abre nuevas posibilidades para el tratamiento de los pacientes en forma ambulatoria, una capacidad fundamental en los países en desarrollo, donde se limita el acceso a camas de hospital y clínicas. Este grupo de aplicaciones consiste en la comunicación de uno o de dos vías para supervisar las condiciones de salud, mantener citas con el médico, o garantizar la adherencia al régimen de medicación. Algunas aplicaciones también pueden incluir pacientes internados y ambulatorios con sensores para la vigilancia (Istepanian et al, 2005)
- **Apoyo a Diagnostico, Tratamiento, Comunicación y Formación de Trabajadores del Sector Salud:** Están diseñados específicamente para proporcionar a los trabajadores de la salud, en zonas remotas, asesoramiento sobre el diagnostico y el tratamiento de los pacientes. Mientras algunos proyectos pueden

proporcionar aplicaciones a teléfonos móviles, como un paso a paso de los sistemas médicos de árbol de decisión para ayudar a los trabajadores de la salud en sus diagnósticos directamente de los propios pacientes, hasta en donde los pacientes pueden tomarse una fotografía de la herida o enfermedad y permitir un diagnóstico y tratamiento. Los proyectos o aplicaciones en el sector de salud móvil implican conectar a un subconjunto de trabajadores de salud, con las fuentes de información médica.

- **Enfermedad y Seguimiento de Brote Epidémico:** Se refiere a aplicaciones con despliegue en dispositivos móviles, con capacidad para capturar de forma rápida y transmitir datos sobre la incidencia de la enfermedad, puede ser decisivo en la prevención y contención de los brotes. Este tipo de aplicaciones ya se están utilizando en el Perú, Ruanda y la India como un sistema de alerta temprana, permitiendo a los funcionarios de salud pública que puedan controlar la propagación de enfermedades infecciosas.
- **Diagnóstico y Tratamiento de apoyo:** Las aplicaciones m-salud en esta área están diseñadas para proporcionar el diagnóstico y el asesoramiento sobre el tratamiento a los trabajadores de salud a distancia a través del acceso inalámbrico a bases de datos de información médica o personal médico. Con diagnóstico m-salud habilitado y apoyo al tratamiento, los pacientes pueden recibir tratamiento en sus aldeas y hogares, evitando la necesidad de costosas visitas al hospital, que están más allá del alcance de muchos (Istepanian et al, 2005).

3.3.1 Aplicaciones de Salud Móvil

M-salud es un área amplia que trasciende múltiples disciplinas y la utilización de una extensa gama de tecnologías. Hay una variedad de aplicaciones, dispositivos y tecnologías de comunicación emergentes en este ámbito, que se pueden combinar para crear sistemas de m-salud (Olla et al, 2008). A continuación se describen las características de aplicaciones m-salud que ya están trabajando.

- En el trabajo (Jovanov et al, 2000) se propone una Red de Área Personal (*PAN, Personal Area Network*) inalámbrica, que integre la información de sensores

inteligentes. En este caso, la interfaz inalámbrica se implementa mediante un transceptor RF a 916 MHz de *Link Technologies*, proporcionando capacidad de comunicación bidireccional a 33.6 Kbps con una cobertura de algo más de 50 metros. El siguiente paso es dotar a este sistema del uso de tecnologías 3G.

- En la misma línea se encuentra el trabajo (Hu et al, 2003), que presenta una red de sensores *ad-hoc* a gran escala, ASNs (*Ad hoc Sensor Networks*), para permitir a los especialistas médicos monitorizar a pacientes situados en cualquier lugar, utilizando para ello tecnologías 3G. El mencionado trabajo se centra en la seguridad de la red, para garantizar la privacidad, que tal y como se indica en (Jovanov et al, 2000) es un factor clave en la monitorización continua durante el desarrollo de la actividad cotidiana del individuo.
- En (Consulting, 2009) se presenta una aplicación denominada “*EpiSurveyor*” desarrollada por *DataDyne*, es una herramienta gratuita de código abierto que permite recopilar datos de un teléfono móvil y mandarlos a un servidor para su posterior análisis. WHO, DataDyne.org y la Technology Partnership pusieron en prueba *EpiSurveyor* en Kenia y Zambia. El éxito en su implementación ha mejorado la puntualidad y disponibilidad de los datos de salud, haciendo más fácil el fortalecimiento de los programas de asistencia sanitaria, con vacunas contra la malaria y otros programas preventivos encaminados a mejorar la salud pública.
- En el país Ruanda, se estima que 0.5 millones de personas son afectadas por el VIH, una solución innovadora se ha encontrado y aplicado con éxito. **TRACnet** (Prajakta et al, 2007) fue establecido y ejecutado por el CRED (Investigación y Tratamiento del Centro del SIDA), una institución del Ministerio de Salud de Rwanda en 2005. Es un sistema dinámico de la tecnología de información que permita recoger, almacenar, recuperar, mostrar y difundir información de los programas críticos, así como para administrar la distribución de medicamentos y la información de los pacientes relacionados con el cuidado y tratamiento del VIH / SIDA. Este sistema permite a los profesionales que participan en anti-retrovirales (ARV), los programas de tratamiento para presentar informes por vía electrónica y tener acceso oportuno a la información vital. Los empleados de los centros de salud pueden entrar a un sitio

Web bilingüe (Inglés y Francés), para enviar o recibir los resultados del programa sobre el VIH / SIDA tan pronto como se procesan (Voxiva, 2009).

En la tabla 3.1 se muestra una lista de algunos proyectos para la salud móvil que se desarrollaron o están en pruebas en diferentes países y se definen características que se tienen y/o manejan en común.

Aplicaciones en el campo de la Salud Móvil			
País	Nombre del proyecto	Descripción	Características en Común
Educación y Conciencia			
México	VidaNet	Vidanet da a las personas que viven con VIH (PVVIH) la posibilidad de registrarse para recibir mensajes para ayudar a mejorar su adherencia a su tratamiento específico. El objetivo es desarrollar un modelo estratégico de comunicación educativa.	Este tipo de Aplicaciones presentan en común el envío y/o recepción de mensajes SMS a través de Teléfonos móviles. Por lo cual se puede definir que estas aplicaciones necesitan presentar interfaces de usuario sencillas, intuitivas, con menús sencillos de manejar, sin necesidad de que el usuario tenga que navegar mucho dentro de la aplicación.
México	CardioNet	Es una solución de auto-cuidado de la salud, las personas llenan un cuestionario de preguntas tales como sexo, edad, peso, talla, problemas de salud, etc. Con base en esto, el individuo es evaluado de acuerdo a la Organización Mundial de la Salud (OMS). A partir de esta evaluación el individuo comienza a recibir mensajes educativos fomentándolo para hacer ejercicio y comer sano.	
Ruanda/ Uganda	ResultsSMS.org	ResultsSMS es una plataforma de código abierto compatible con OpenMRS diseñado para entregar resultados de alguna prueba médica de los pacientes, la educación del paciente, las citas de seguimiento y recordatorios a los pacientes a través de SMS.	
India	Libertad del VIH/SIDA	Se compone de cuatro juegos móviles dirigidos a diferentes mentalidades y la psicología de los usuarios móviles. Los juegos se instalan desde dispositivos de bajo nivel (blanco/Negro) hasta dispositivos sofisticados de alto nivel (color).	
Apoyo Diagnostico, tratamiento, comunicación y la formación del personal sanitario			
India	Tele-Doc	TeleDoc para dispositivos de telefonía móvil de los trabajadores de salud de los poblados en la India, que les permite comunicarse con los médicos que utilizan una aplicación web para ayudar a diagnosticar y prescribir a los pacientes.	Para este tipo de aplicaciones además de presentar IU sencillas, se tiene que poner mucha énfasis en la

Perú	Nacer	Nacer es un teléfono- y WEB-, basado en un sistema de información y comunicación para la salud materna e infantil que permite a los profesionales de la salud en lugares remotos de comunicación el intercambio de información de salud crítica entre los expertos médicos y los hospitales regionales. Todos los datos reportados se registran en una base de datos central, y está disponible para los funcionarios de salud en tiempo real para el análisis y la toma de decisiones.	conexión a Servidores WEB y Bases de Datos ya que este tipo de aplicaciones recolectan la información la cual es almacenada y procesada en tiempo real.
Rwanda	TRACnet	Diseñado para recoger, almacenar, recuperar y difundir el programa crítico de medicamentos y la información de pacientes relacionados con el tratamiento de VIH/SIDA. Bajo el liderazgo del Ministerio de Salud y la Investigación de Tratamiento y Centro del SIDA (TRAC), TRACnet se está implementando para aumentar la eficiencia en el servicio del VIH/SIDA, y mejorar la calidad de la atención al paciente.	
La vigilancia de enfermedades, la recopilación de datos a distancia, y el seguimiento de brote epidémico			
Kenia y el Mundo	EpiSurveror.org	EpiSurveyor es un sistema en línea desarrollado por DataDyne.org que permite el rápido desarrollo de las formas de recolección de datos que pueden ser descargados en los teléfonos móviles: el usuario puede utilizar en horas todo el funcionamiento del almacenamiento de datos.	En este tipo de proyectos existen varios aplicaciones que dan servicio en diferentes maneras al sector de la salud, las cuales requieren de sistemas m-salud de fácil y rápido aprendizaje, que permitan la captura de información, y de esta manera se apoya al problema que existe con la inasistencia al médico de gran parte del sector de la población por diversas situaciones la más común por recursos económicos limitados. La información Capturada debe ser
	Voxiva Health Watch	Voxiva HealthWatch es una plataforma integrada de vigilancia utilizados por los organismos de salud pública en todo el mundo para apoyar la vigilancia integrada de las enfermedades, la vigilancia de síndromes, y la respuesta coordinada.	
India	AESSIMS	AESSIMS está diseñado para aumentar la capacidad de salud a nivel de campo al permitir trabajadores de primera línea de salud que informe la incidencia de enfermedades a través de una innovadora combinación de teléfono y la tecnología basada en web que aprovecha la infraestructura disponible	

			almacenada en el mismo dispositivo o de manera remota a una BD para después poder ser consultada por el paciente o el trabajador del sector salud.
El Tratamiento de Apoyo y cumplimiento de la medicación para pacientes, incluida la gestión de enfermedades Crónicas.			
Kenia	Weltel	A pacientes con VIH-positivos les fueron enviados mensajes de texto por semana preguntando por su bienestar. Los pacientes respondieron el mensaje diciendo que todo estaba bien, o que tenían un problema. Si hay un problema, un trabajador de salud los llama para ayudarles.	Para este tipo de aplicaciones son muy similares las características que se mencionaron en las aplicaciones anteriormente mencionadas: IU sencillas, Captura de datos en tiempo real, almacenamiento de la información en el dispositivo y/o BD.
México	Diabediario	Es una solución para el cambio de estilos de vida de los diabéticos y para controlar y mejorar su adherencia a su tratamiento de la diabetes. Cualquier persona con diabetes, que tiene un teléfono celular Telcel, pueden participar en el programa. Diabediario utiliza las telecomunicaciones para generar cambios de actitud hacia la prevención de riesgos y el cumplimiento de los tratamientos prescritos.	
Perú	Cell-Preven	Es un sistema utilizado en los teléfonos móviles para la recolección de datos obtenidos de entrevistas a trabajadoras sexuales, estos datos obtenidos permiten ayudar a las trabajadoras sexuales con el tratamiento de vaginosis a través de mensajes SMS.	
Estados Unidos	mCare	Departamento Médico del Ejército de USA utiliza la aplicación en su teléfono móvil para mensajería en el caso de reintegrar a los soldados heridos. Mensajes SMS basados en consejos de salud, recordatorios de citas para soldados Estadounidenses que regresan del servicio.	
Indiana	DIMA-Aplicación de Monitoreo en el consumo de	Es una aplicación para PDA llama DIMA (Solicitud de revisión de la ingesta dietética) para pacientes con diálisis de bajo nivel de alfabetización y limitada experiencia con tecnología para supervisar ingesta de líquidos y sodio. Los pacientes pueden	

	alimentario.	seleccionar los iconos de alimentos en la pantalla de un PDA, esto permite el seguimiento de los niveles de líquidos y sodio, que pueden ayudar a los pacientes a controlar su ingesta de alimentos y mejorar su calidad de vida.	
--	--------------	---	--

Tabla 3.1 Aplicaciones de Salud Móvil.

En Latinoamérica las aplicaciones piloto de salud móvil se han centrado en el monitoreo y en el tratamiento de enfermedades basado en centros de atención telefónica. Pero ya es hora de pasar a aplicaciones que aprovechen completamente la tecnología que ya está disponible en la región para la salud. Sin embargo, algunas aplicaciones móviles podrían implementarse fácilmente con inversiones iniciales bajas y demostrarían su valor en un corto plazo, logrando un sistema de salud mejor y más accesible para la mayoría (Anta et al, 2009).

3.3.2 Establecimiento de Módulos para una Aplicación de Salud Móvil

En base a la información presentada en la columna “Características en común” de la tabla 3.1 que describen las particularidades en común que presentan las diferentes aplicaciones de salud móvil, llegamos a la conclusión de que existe la necesidad de establecer los módulos mínimos que requiere el desarrollo de una aplicación de salud móvil. Por lo cual se establecen los módulos para una aplicación del tipo de monitorización remota, que será el tipo de aplicación que desarrollaremos para el caso de estudio. Los módulos son los siguientes:

1. **Comunicación con Sensores Médicos / Adquisición de Datos de los sensores médicos.** Los datos son generalmente transferidos a intervalos periódicos y analizados por el proveedor de servicios. Para los Sistemas de Salud Móvil es necesario que los datos sean capturados y analizados en tiempo real. Una comunicación eficaz y procesamiento de datos en tiempo real es esencial.

Las tecnologías ampliamente utilizadas comercialmente incluyen Bluetooth y ZigBee. En este estudio nos enfocaremos a la comunicación con Bluetooth debido a que los dispositivos móviles ya tienen integrada esta tecnología, además de que BT presenta una mayor velocidad de transmisión, como se menciona en el capítulo 2.

2. **Almacenamiento de la Captura de Datos.** En la mayoría de las aplicaciones de Salud Móvil con sensores, los datos fisiológicos del paciente son transmitidos a un servidor central de salud, donde se tenga disponible y capacidad de almacenamiento. Estos datos son generalmente transferidos a intervalos periódicos y analizados por el proveedor de servicio. Para un sistema con sensores la frecuencia de recolección de datos es mucho mayor que en muchos estudios ambientales. Dado que las emergencias y riesgos para la salud pueden ocurrir en cualquier momento, es necesario que los datos se capturen y analicen en tiempo real. Una comunicación eficaz y el procesamiento en tiempo real de los datos son esenciales (Prajakta et al, 2007).
3. **Interfaz de usuario.** Como se describió en el capítulo 2 los teléfonos móviles presentan limitaciones en sus capacidades y una de ellas es su pantalla. Por lo cual es importante tener en cuenta este punto al momento de desarrollar las aplicaciones

de salud móvil. Para que las redes de sensores que se manejen sean con IU sencillas que el usuario no tenga la necesidad de navegar por varias pantallas, además de contar con menús y/o opciones fáciles de manejar.

4. **Presentación de Información de manera Gráfica.** Como se menciona en el modulo 3 las interfaces de usuario deben ser sencillas, pero entendibles para el sector salud. Es necesario presentar la información capturada de una forma grafica para su comprensión rápida, esto a su vez lo hace sencillo para el usuario.

3.4 Resumen

En este capítulo se describió de manera más extensa lo que es Salud Móvil, además se describe la clasificación de estos y se proporcionan algunos ejemplos de aplicaciones de este tipo. Tomando como base la bibliografía consultada se definieron los requerimientos o módulos mínimos que se necesitan para el desarrollo de una aplicación de salud móvil para la monitorización remota. Son 4 los módulos mínimos que deben tener: 1. Captura de Datos de un sensor lo cual permite la captura de la información fisiológica que se requiera, estos datos pueden ser capturados utilizando diferentes tipos de tecnologías de comunicación. 2. Almacenamiento de Datos, es importante que los datos que se capturen desde los sensores médicos sean guardados para su posterior análisis, el almacenamiento puede ser de manera remota a un servidor Web o directamente en el mismo dispositivo móvil. 3. Interfaz de Usuario, el diseño de las interfaces de usuario son parte importante de aplicaciones de dispositivos móviles pequeños sobre todo en teléfonos celulares, debido a sus capacidades tan limitadas. 4. Presentación de información de manera grafica, la interpretación de los datos que se capturan desde los sensores médicos, es importante presentárselas a los servidores del sector salud de manera grafica ya que esto les facilita su interpretación.

CAPITULO IV

4. DISEÑO DE LA APLICACIÓN DEL CASO DE ESTUDIO

4.1 Introducción

Como se veía en el capítulo 3, las aplicaciones de salud móvil se clasifican de acuerdo a los servicios que proporciona cada una de ellas. Para el Caso de estudio de este trabajo, con el análisis de la información anteriormente presentada se decidió realizar una aplicación que se asocia en la clasificación de los sistemas de salud móvil para el Monitoreo de Paciente. La cual nos ayudará a determinar las métricas con las cuales se evaluarán las plataformas de programación Java ME y Android.

4.2 Diseño de la Aplicación del Caso de Estudio

El diseño de la Aplicación del Caso de Estudio fue determinado en base a los módulos que se establecieron en el capítulo 3 para una aplicación de salud móvil. En la figura 4.1 se ilustra este diseño.

El caso de estudio utilizado en este trabajo se realizó en dos partes: primero se desarrollo una aplicación que sólo permite capturar la temperatura ambiente registrada por un sensor de temperatura y guardarlas en una BD y directamente en el teléfono móvil. Esta aplicación fue desarrollada en Android y Java ME para evaluar los módulos 1 y 2 descritos en el capítulo 3. Después, se tomó como referencia la aplicación FCM (Frecuencia Cardiaca Móvil) en Android de (Alaníz Sida, 2011) para desarrollarla en la plataforma Java ME y realizar la evaluación de los módulos 3 y 4 del capítulo 3.

A continuación se describe la aplicación FCM, debido a que es la que se desarrolló completamente en ambas plataformas cubriendo los 4 módulos que se definieron para el desarrollo de las aplicaciones móviles (capítulo 3).

- El paciente con algún tipo de problema que requiera seguimiento continuo tendrá el sensor médico, el cual podrá comunicarse con el dispositivo móvil utilizando la tecnología Bluetooth; dicho sensor trabaja sobre la BAN (Body Area Network, por sus siglas en inglés) (Chan et al, 2008) y la aplicación capturará los datos enviados

por dicho sensor; con esto cumplimos el módulo “Comunicación y Adquisición de Datos del Sensor Médico”.

- Estos datos son recibidos por un dispositivo móvil donde la aplicación será capaz de llevar un monitoreo de la frecuencia cardiaca del paciente, guardando una bitácora de su comportamiento, la información es almacenada directamente en el teléfono móvil y pueden ser enviados a una BD; con este punto cubrimos el módulo “Almacenamiento de Captura de Datos”
- El paciente podrá enviar la bitácora al personal profesional de la salud para que este pueda llevar a cabo un análisis de la información. Dicha información se enviará por medio de correo electrónico y contendrá información general sobre el paciente, así como los datos registrados por el sensor y una liga hacia una gráfica generada a partir de los datos recolectados; con lo cual abarcamos el módulo de “Presentación de la información de manera Gráfica”.

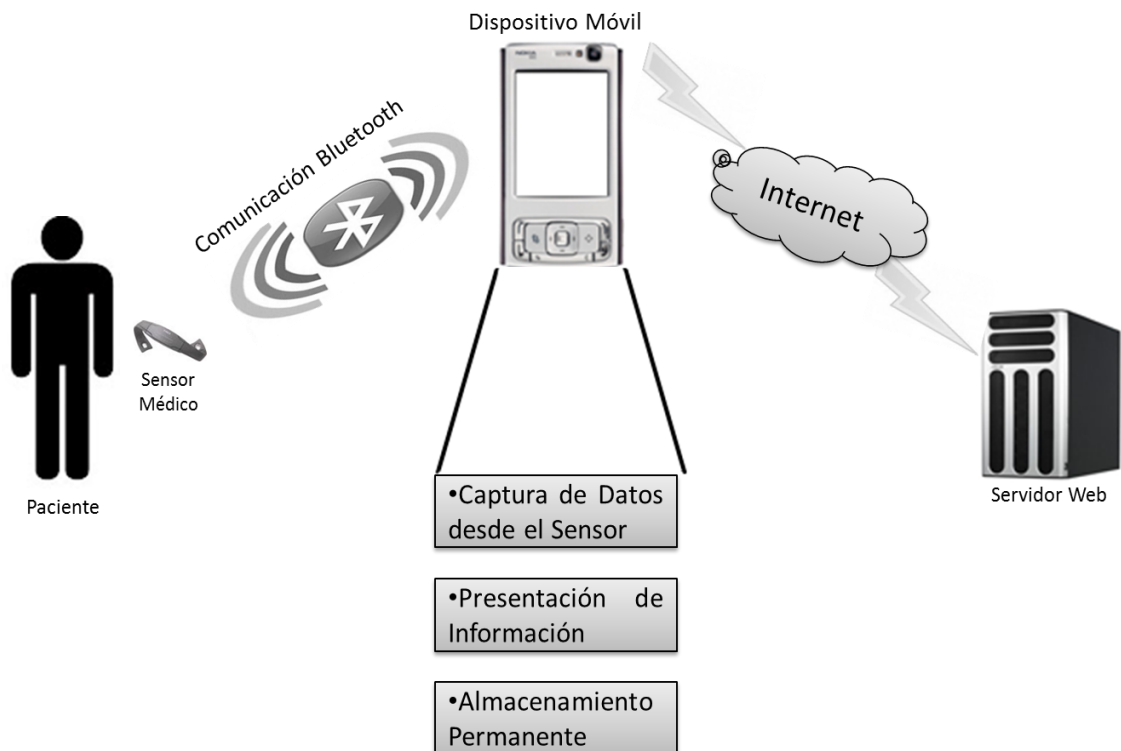


Figura 4.1 Diseño de una Aplicación de Salud Móvil.

En la Figura 4.2 se muestra el diagrama a detalle del proceso que realiza la aplicación que se utilizará en el caso de estudio, la cual fue desarrollada en las plataformas Java ME y Android. El funcionamiento del sistema de salud móvil consiste en:

- El paciente deberá seleccionar su *Modo de Estado*, es decir, si el paciente se encuentra en estado NORMAL (realizando actividades que normalmente realiza diario), realizando algún DEPORTE o DURMIENDO (descansando), con la finalidad de medir los resultados dependiendo de la actividad que se esté realizando y mostrar una alerta en caso necesario. Si el paciente no indica su estado e inicia la captura de datos del sensor, automáticamente se tomará como estado “Normal”.
- El paciente inicia la captura de datos del sensor médico conectándose a dicho sensor. La aplicación mostrará en la pantalla del dispositivo cada 30 segundos automáticamente lo que este leyendo del sensor que muestra al usuario si se presenta alguna anomalía en sus intervalos de frecuencia; para ello se utilizan los colores verde (normal), amarillo (precaución) y rojo (crítico) dependiendo de la frecuencia cardíaca.
- La aplicación permite al usuario introducir alguna nota para reportar alguna anomalía relacionada con su padecimiento, la cual es guardada en el teléfono móvil.
- Además permitirá enviar de manera automática un correo electrónico al médico especialista los datos capturados en la aplicación, así como la (s) nota (s) que redactó el usuario.
- Para mostrar los datos capturados del paciente sobre el comportamiento de la frecuencia cardíaca de manera gráfica se envían los datos en un URL al servicio que proporciona Google para mostrar gráficas llamado “Google Chart Tools”, este servicio se describen en el capítulo 5 a detalle.

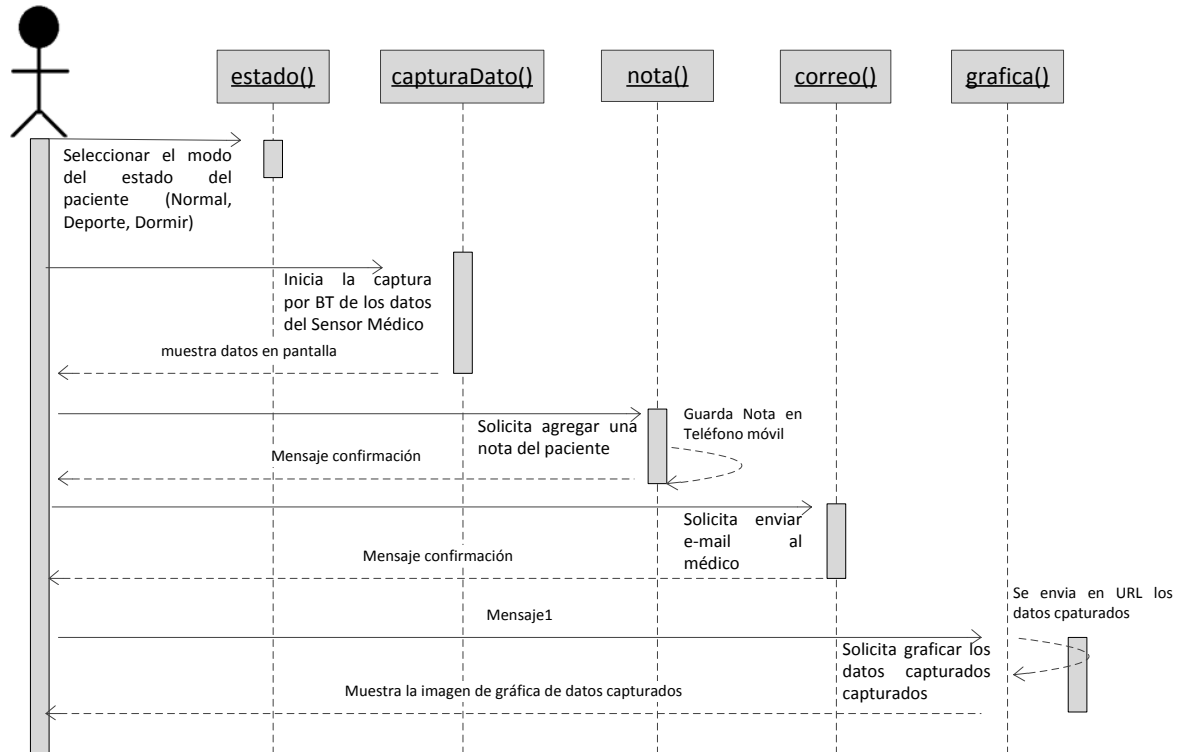


Figura 4. 2 Diagrama general de iteraciones de la aplicación

4.3 Métricas de evaluación para las Plataformas Java ME y Android

Con la determinación de los módulos de una aplicación m-salud (ver capítulo 3) y el diseño del caso de estudio que se define en este capítulo, nos auxiliaron a definir las métricas que nos permitan evaluar las plataformas de programación Java ME y Android específicamente para aplicaciones de esta área.

Las métricas se dividieron de la siguiente manera:

- **Conexión y Comunicación BT con Sensor Médico:** para evaluar esta parte de la aplicación será necesario conocer las funcionalidades que presenta cada plataforma (Java ME y Android) para realizar la conexión y comunicación BT, así como el tiempo promedio de conexión y lectura de los datos entre el dispositivo móvil y el sensor médico.
- **Comunicación Remota con el Servidor:** para este segmento se presentarán las funcionalidades que contiene cada plataforma para realizar la comunicación entre el

dispositivo móvil y el servidor, así como los tiempos de conexión, lectura y escritura.

- **Almacenamiento Persistente:** Se presentarán las funcionalidades de cada plataforma para realizar el almacenamiento directamente en el teléfono móvil.
- **Interfaz de Usuario:** Para esto se presentarán las funcionalidades que cada plataforma proporciona para el desarrollo de las interfaces de usuario, además se realizarán encuestas a usuarios que utilicen la aplicación en ambas plataformas para obtener información de sus opiniones acerca de las interfaces programadas en dichas plataformas.

Para todos los casos anteriormente mencionados se presentará el número de líneas de Código utilizados por cada plataforma y una métrica general será el peso en Kilobytes de cada aplicación desarrollada. En la Figura 4.3 se muestra el diagrama de las métricas anteriormente descritas.

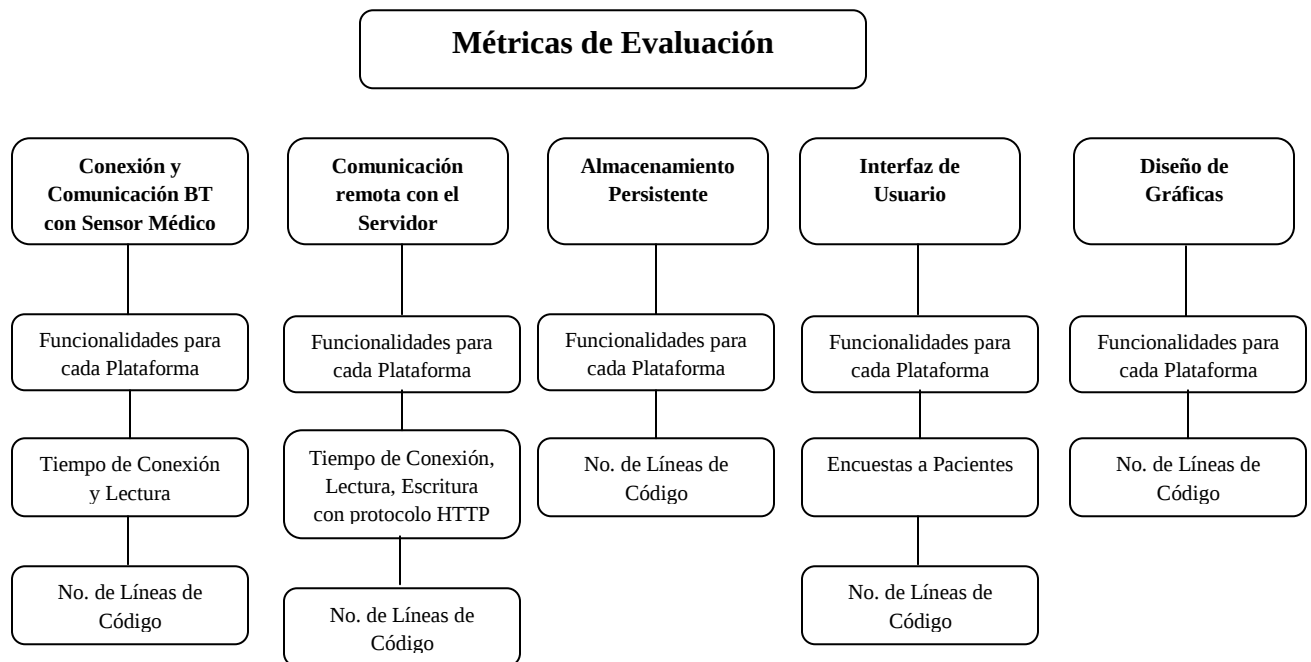


Figura 4. 3 Diagrama de Métricas de Evaluación

4.4 Resumen

En este capítulo se muestra el Diseño de la Aplicación utilizando los requerimientos definidos en el capítulo 3. También se describe el funcionamiento de la aplicación utilizada para el caso de estudio y se definen las métricas que se tomaran en cuenta para la evaluación de las plataformas en los capítulos 5 y 6.

CAPITULO V

5. EVALUACIÓN CUANTITATIVA

Como se mencionó en el capítulo 4, para poder realizar la evaluación comparativa entre las plataformas de programación móviles fue necesario determinar métricas que nos proporcionen información del comportamiento de cada plataforma. Los resultados de estas métricas se obtienen en 3 diferentes escenarios de uso de la aplicación m-salud, lo cual permite obtener las mediciones de una manera específica en el uso real de la aplicación. En las figuras 5.1, 5.2 de este capítulo y en la figura 6.1 del capítulo 6 se ilustran los escenarios que se diseñaron para realizar las pruebas que permiten la evaluación de las plataformas.

Para el desarrollo de la evaluación se requirió el siguiente equipo:

1. Dispositivos Móviles
2. Interfaz Bluetooth y Sensor de Temperatura
3. HxM™ Bluetooth™
4. Servidor Web
5. Lenguaje de Programación: Java SEE, Java ME y XML.
6. Software de Soporte: NetBeans IDE 6.9.1 y Eclipse Classics 3.7.1

Los dispositivos móviles que se utilizaron fueron distintos debido a que cada uno soporta las diferentes plataformas de programación **Java ME** y **Android**. El dispositivo #1 es un equipo de la serie Nokia modelo N95, es un teléfono inteligente, con sistema operativo: Symbian OS ver. 9.2, interfaz del usuario: S60 3a edición paquete de funciones (S60 3.1), Java: MIDP 2.0, SDKs C++ y Java, conectividad: WLAN - IEEE802.11 g/b y Bluetooth 2.0 (Nokia). El dispositivo #2 es un teléfono de la serie de SAMSUNG modelo GALAXY Ace (GT-S5830), basado en el nuevo sistema operativo Android™ 2.2, interfaz de usuario Pantalla completamente táctil, Java SEE y Java MIDP emulador, conectividad: Bluetooth 2.0, WIFI (802.11 b/g/n) (Samsung). Además de los teléfonos móviles, se utilizó una computadora portátil con procesador Intel (R) core(TM) Duo CPU T5550@1.83GHz, memoria RAM 3.0GB y sistema operativo windows de 32 Bits.

La interfaz BT que se utiliza es el llamado BlueSentry RN-800-CB, es un módulo simple Bluetooth I/O que permite la comunicación inalámbrica para clientes PC y PDA con perfil Bluetooth de puerto serie (SPP). Se conecta fácilmente a las salidas del sensor analógico y automáticamente toma mediciones continuas. Los datos se transfieren de forma inalámbrica a los clientes con SPP. Las aplicaciones de software a través de Bluetooth pueden controlar y adquirir datos como si estuvieran conectados a un puerto serie local (BlueSentry). El sensor de temperatura es conectado fácilmente a las salidas analógicas que tiene dicha interfaz; el sensor que se utilizó para la evaluación fue el LM35 el cual captura la temperatura ambiente con una precisión calibrada de 1°C y un rango que abarca desde -55° a +150°C (LM35). Además de esta interfaz BT y el sensor de temperatura, se utilizó el dispositivo HxMTM BluetoothTM, que es un cinturón que al colocarlo en el tórax comienza a tomar el ritmo cardíaco, y la información la envía en tiempo real por Bluetooth (Zephyr). Conjuntamente del equipo físico que se requirió para esta evaluación, fue necesario tener los conocimientos básicos de los lenguajes de programación Java SEE, Java ME y XML para el desarrollo de las aplicaciones que nos permitieron obtener los resultados deseados. Para dicha programación se instalaron en una PC el software que me permitió desarrollar las aplicaciones; para la programación de la aplicación en Java ME fue necesaria la instalación de NetBeans IDE 6.9.1 y para la aplicación en Android se instaló el software IDE Eclipse Classics 3.7.1.

5.1 Evaluación del ESCENARIO 1: conexión y comunicación BT

La evaluación del escenario 1 fue diseñado con el objetivo de cubrir las métricas establecidas en el capítulo 4 para el módulo “conexión y comunicación BT con el sensor médico” definido en el capítulo 3 de este mismo trabajo de investigación.

Las métricas que se cubrirán con esta evaluación son: *tiempo de conexión BT (TCB)*, *tiempo de captura de datos (TCD)*, *funcionalidades para comunicación BT* y *el No. de líneas de código*. Las métricas TC y TCD nos ayudarán a obtener los tiempos promedios de conexión y comunicación Bluetooth de cada plataforma y de esta manera ver el comportamiento de cada una de ellas. La métrica de funcionalidades y de no. líneas de código nos brindaran información sobre las herramientas con las que cuenta cada plataforma para poder llevar a cabo esa comunicación BT, así como el total de líneas de

códigos que cada una de ellas genera para poder lograrlo. La figura 5.1 muestra de manera gráfica el escenario que se utilizó para la evaluación del escenario 1.

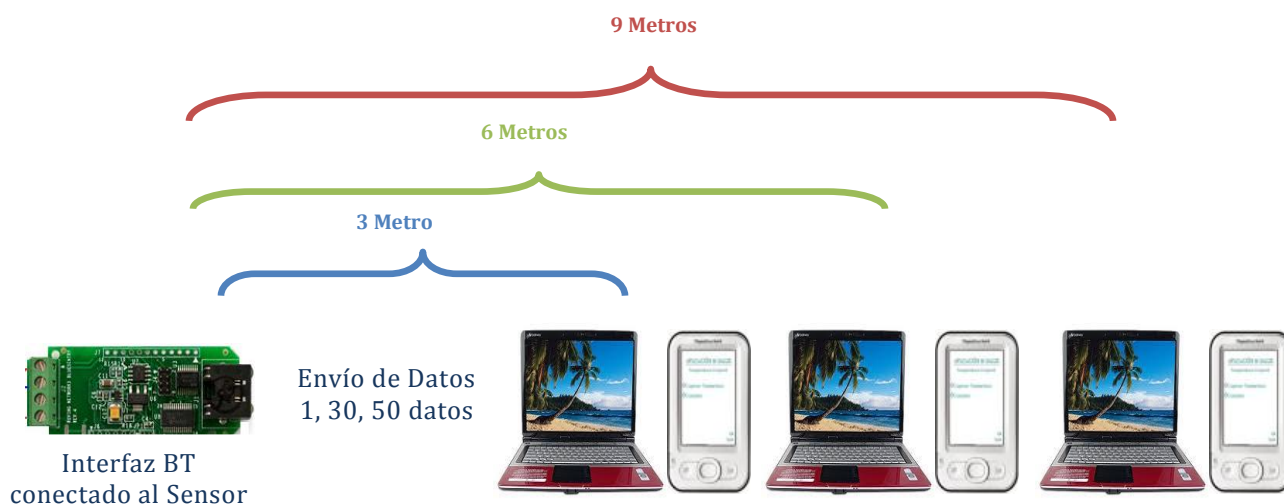


Figura 5.1 Escenario 1. Conexión y comunicación Bluetooth entre la Interfaz BT y Emulador/Dispositivo para cada Plataforma

La Figura 5.1 nos muestra que la evaluación se realizó de 2 maneras, primero la conexión y captura de datos utilizando la interfaz BT y el emulador y después entre la interfaz BT y el dispositivo móvil.

Como sabemos BT tiene un alcance de comunicación de hasta 10 metros, por tal motivo se decidió realizar la conexión y comunicación a diferentes distancias y de esta manera ver el comportamiento de BT en las 2 diferentes plataformas. En base a la bibliografía consultada se decidió establecer las distancias de 3, 6 y 9 metros, esta última como la distancia máxima, ya que después de los 10 metros hay una gran pérdida de información. A esta misma distancia se tomaron lecturas de 1, 30 y 50 temperaturas, estas lecturas se tomaron en 2 situaciones, la primera haciendo la lectura sin ningún obstáculo entre la interfaz BT y emulador/dispositivo móvil; y la segunda utilizando un obstáculo (pared falsa) entre ellos.

El parámetro TC y TCD se grafican respecto a la distancia entre la interfaz BT y el dispositivo móvil (computadora portátil y teléfonos celulares). Los resultados son el promedio de 15 iteraciones en cada medición. Es decir, conectarse el dispositivo móvil con

interfaz BT y leer la temperatura una sola vez 15 veces a una distancia de 3, 6 y 9 metros y lo mismo se hizo para los 30 y 50 datos.

5.2 Evaluación del ESCENARIO 2: conexión y comunicación con el servidor

La evaluación del escenario 2 (ver figura 5.2) fue diseñado con la finalidad de cubrir las métricas para el módulo “Comunicación Remota con el Servidor” definido en el capítulo 3.

Esta evaluación nos permite realiza la comunicación entre los dispositivos móviles y el Servidor Web, con lo que obtenemos los resultados para las métricas: *tiempo de conexión al servidor (TCS)*, *tiempo de escritura al servidor (TES)*, *tiempo de lectura al servidor (TLS)*, *funcionalidades de cada plataforma y no. de líneas de código*.

Para poder obtener estos resultados lo que se realizó en esta parte de la evaluación fue la creación de un cliente HTTP que permitiera la comunicación (conexión, lectura y escritura) con un servidor HTTP Apache.

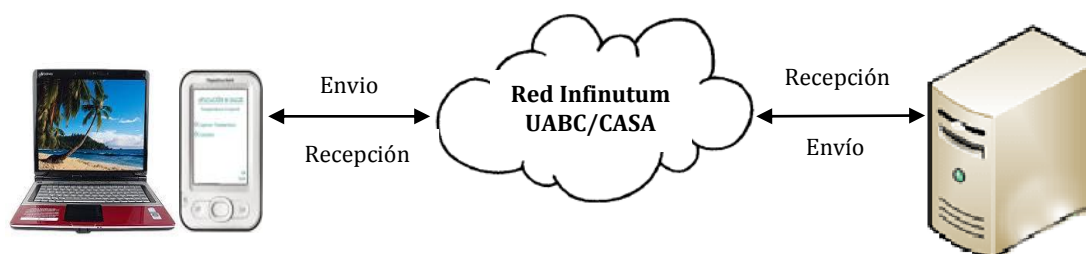


Figura 5.2 Escenario 2. Conexión y comunicación del emulador/dispositivo móvil al servidor web.

El desarrollo del cliente HTTP tuvo que realizarse en ambas plataformas para poder realizar la comparación de tiempos entre ellas.

En la aplicación MIDP de Java ME, se utilizaron conjuntamente los paquetes *javax.microedition.io* y *java.io* de la siguiente manera; el primer paquete contiene numerosas clases que nos permitirán crear y manejar diferentes conexiones de red. Estas conexiones podrán usar diferentes formas de comunicación: HTTP, datagramas, sockets, etc. aquí solo se utilizó la conexión HTTP. El paquete *java.io* se encargó de proporcionarnos las clases necesarias para leer y escribir en estas conexiones. Estas clases orientadas a la conexión en red y comunicaciones reciben el nombre de *Generic Connection Framework* (Galvez & Ortega, 2007).

La plataforma Android nos provee de numerosos paquetes para implementar conexiones en red; en esta aplicación se utilizaron solo 2 de estos paquetes *java.net* y *apacher.Org.**.

Los resultados obtenidos son el promedio de 30 iteraciones de cada medición, es decir, se realizó la conexión desde el emulador y dispositivo móvil al servidor apache 30 veces, utilizando los diferentes paquetes anteriormente mencionados y así sucesivamente con las otras mediciones. Estas mediciones fueron realizadas conectándose a 2 diferentes redes inalámbricas, a) RED 1: red infinitum de CASA con conectividad IPv4 y con una velocidad hasta de 2 Mbps, b) RED 2: red infinitum de UABC con conectividad IPv4 y con una velocidad máxima de 10 Mbps. Como se sabe todas las redes inalámbricas presentan cierta inestabilidad en la velocidad de transmisión, esto debido a que las conexiones WiFi trabajan en la banda de 2.4 GHz, que es la misma banda utilizada por una gran cantidad de dispositivos de comunicación (por ejemplo, los teléfonos inalámbricos, sobre todo si son algo antiguos, Bluetooth y otros) o incluso recibe interferencias de una gran cantidad de electrodomésticos (por ejemplo, un horno microondas). La diferencia que se intento tener entre las 2 redes fue que en la red 1 tuviera el menor tráfico posible de información sobre la banda de comunicación de 2.4GHz, mientras que en la red 2 se tiene una mayor cantidad de tráfico que no es posible controlar. Esto con la finalidad de conocer el comportamiento de las plataformas Java ME y Android al realizar las comunicaciones inalámbricas en los dos casos (sin tráfico y con tráfico).

5.3 RESULTADOS de la evaluación del ESCENARIO 1.

5.3.1 Tiempos de Conexión Bluetooth

A partir del escenario 1 que se utilizó para la evaluación que se describió en la sección 5.1, se obtuvieron los resultados que a continuación se presentan.

En la tabla 5.1 se muestran los promedios de tiempos de conexión Bluetooth (TCB) obtenidos entre a) Emulador e interfaz BT y b) Teléfono móvil e interfaz BT, los promedios obtenidos están en la escala de tiempo milisegundos.

Emulador		
Distancia en metros	JAVA ME	ANDROID
3	158	157.46
6	163.66	162.73
9	173.33	176.46
Teléfono móvil		
Distancia en metros	JAVA ME	ANDROID
3	183.93	178.26
6	191.93	178.66
9	194.8	181.46

Tabla 5.1 Tiempo (ms) promedio de conexión BT entre emulador/teléfono móvil y la interfaz BT.

Los resultados de la tabla anterior se presentan de manera gráfica en las figuras 5.3 y 5.4, en estas gráficas podemos observar claramente el comportamiento que se presenta en la conexión BT entre los emuladores y teléfonos móviles para cada plataforma con la interfaz BT. Puede Observarse en la figura 5.3 que utilizando el emulador el comportamiento de los tiempos entre las plataformas fue muy similar, sin embargo en la figura 5.4 en la cual se muestran los resultados utilizando los 2 teléfonos móviles se puede apreciar una mayor diferencia entre ellas, donde la que mantiene más estabilidad en los tiempos de conexión Bluetooth es la plataforma Android.

Las mediciones de las conexiones y lecturas de temperaturas se realizaron en dos circunstancias, primero se realizó la prueba en la computadora portátil utilizando los emuladores para ambas plataformas y de esta manera tener exactamente las mismas características y condiciones; después se realizó la misma prueba en los 2 teléfonos móviles correspondiente a cada plataforma.

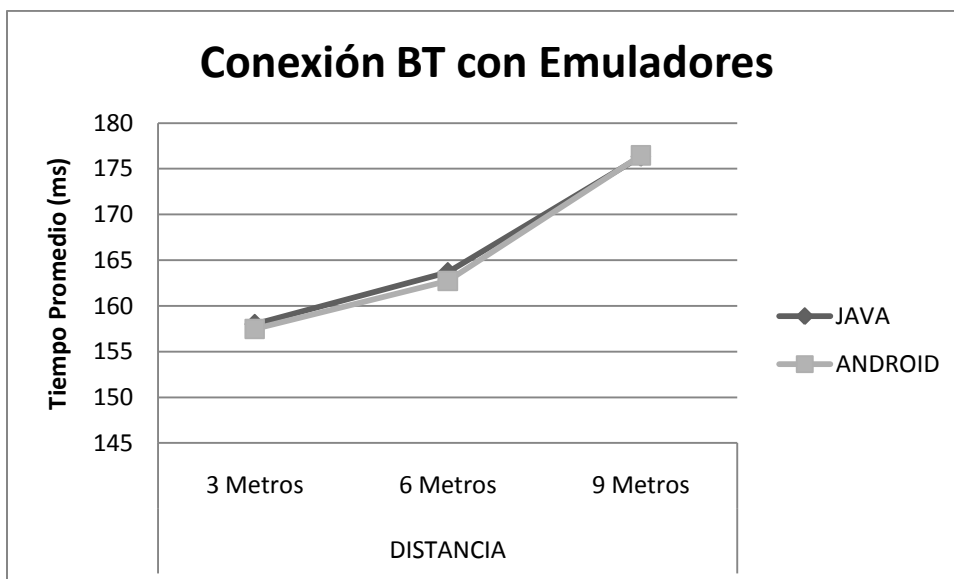


Figura 5.3 Gráfica de tiempos (ms) promedios de conexión BT entre los emuladores y la interfaz BT

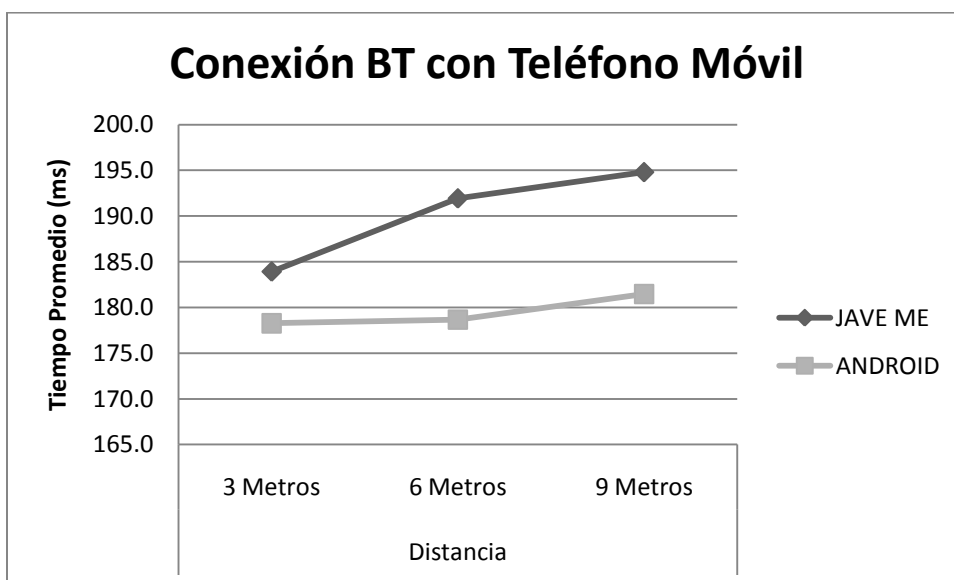


Figura 5.4 Gráfica de tiempos (ms) promedios de conexión BT entre los teléfonos móviles y la interfaz BT

En la tabla 5.2 se muestran los resultados de los tiempos (ms) de la captura de datos (TCD) con Bluetooth que realiza el dispositivo móvil desde el sensor médico. En la primera columna se tienen las diferentes distancias (metros) de las cuales se realizaron las capturas de los datos; las siguientes 3 columnas nos indican los promedios de TCD de las diferentes cantidades de datos (1,30 y 50 datos). Estas capturas de temperaturas se realizaron entre a) emulador e interfaz BT y b) teléfonos móviles e interfaz BT **sin** tener **obstáculo** alguno entre ellos. En las figuras 5.5 y 5.7 se muestran de manera gráfica estos resultados.

Emulador						
Distancia en metros	1		30		50	
	JavaME	Android	JavaME	Android	JavaME	Android
3	162.93	157.40	4975.49	4617.96	8445.78	7494.63
6	168.4	162.53	5359.73	4783.67	9142.64	8346.18
9	194.13	178.73	6086.50	5572.78	9982.17	9153.37
Teléfono móvil						
Distancia en metros	1		30		50	
	JavaME	Android	JavaME	Android	JavaME	Android
3	185.26	173.53	5778.37	5174.26	9159.40	8498.73
6	199.93	179.86	6143.18	5256.18	9869.86	8874.23
9	216.33	197.26	6689.93	5867.96	10916.50	9647.40

Tabla 5.2 Tiempo (ms) promedio de lecturas de temperaturas entre el dispositivo móvil y la interfaz sin obstáculo.

En la tabla 5.2 se muestran los resultados de los tiempos (ms) de la captura de datos (TCD) con Bluetooth que realiza el dispositivo móvil desde el sensor médico. En la primera columna se tienen las diferentes distancias (metros) de las cuales se realizaron las capturas de los datos; las siguientes 3 columnas nos indican los promedios de TCD de las diferentes cantidades de datos (1,30 y 50 datos). Estas capturas de temperaturas se realizaron entre a) emulador e interfaz BT y b) teléfonos móviles e interfaz BT **con obstáculos** entre ellos. En las figuras 5.6 y 5.8 se muestran de manera gráfica estos resultados.

Emulador						
Distancia en metros	1		30		50	
	JavaME	Android	JavaME	Android	JavaME	Android
3	174.33	168.67	5108.73	4789.49	8598.46	7635.33
6	189.55	176.89	5521.26	4918.96	9306.13	8582.78
9	247.93	216.5	6297.39	5746.83	10122.95	9328.46
Teléfono Móvil						
Distancia en metros	1		30		50	
	JavaME	Android	JavaME	Android	JavaME	Android
3	238.62	193.97	6263.46	5389.76	9765.33	8726.76
6	272.36	219.23	6598.13	5621.99	10598.16	9163.4
9	321.98	284.63	7273.27	6156.33	11237.86	11052.23

Tabla 5.3 Tiempo promedio de lecturas de temperaturas en milisegundos entre el dispositivo móvil y la interfaz con obstáculos

Con estos resultados obtenidos podríamos decir que al realizar la conexión en la plataforma Android se realiza con mayor rapidez, sin embargo no podemos afirmar que siempre pueda llegar hacer de esta manera, ya que eso depende del modulo de interfaz BT que contenga el equipo celular y no del sistema operativo o de la plataforma en la que se pueda programar al teléfono celular, si no del fabricante de dicho equipo. Por lo que en lo personal la métrica del tiempo de conexión BT no me parece relevante en esta evaluación, pero lo que si me parece relevante son las API's que contiene cada plataforma que se describen mas adelante en el punto 5.3.2.

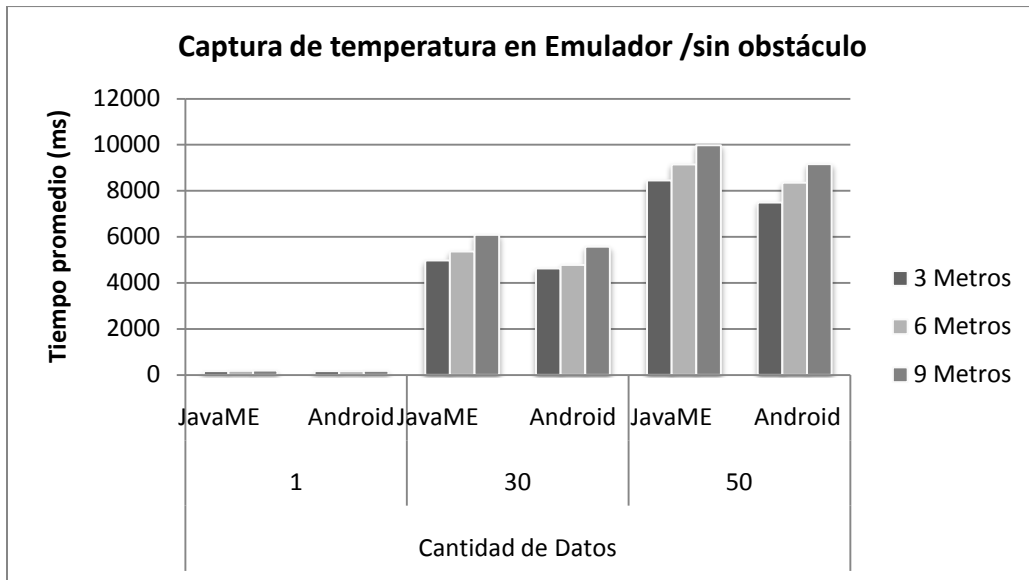


Figura 5. 5 Gráfica de promedios de TCD (ms) en diferentes distancias con los emuladores **sin obstáculo**

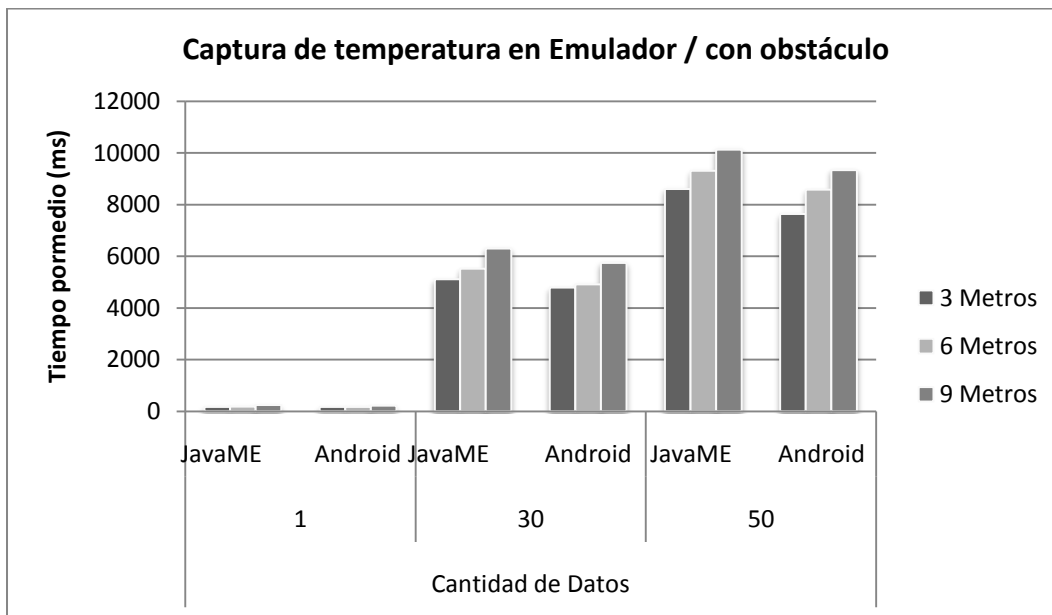


Figura 5. 6 Gráfica del promedio de tiempo la captura de temperatura a diferentes distancias con los emuladores **con obstáculo**

En las figuras 5.5 y 5.6 se puede apreciar que no hay una gran diferencia en los tiempos promedios poniendo obstáculos en entre la interfaz BT y el emulador. Por otra parte tenemos que en los teléfonos móviles (figura 5.7 y 5.8) si se encuentra una diferencia al tener algún obstáculo entre ellos, pero se puede observar que en el dispositivo con la

plataforma Android presenta una mayor estabilidad al capturar la temperatura sin obstáculo y con obstáculo.

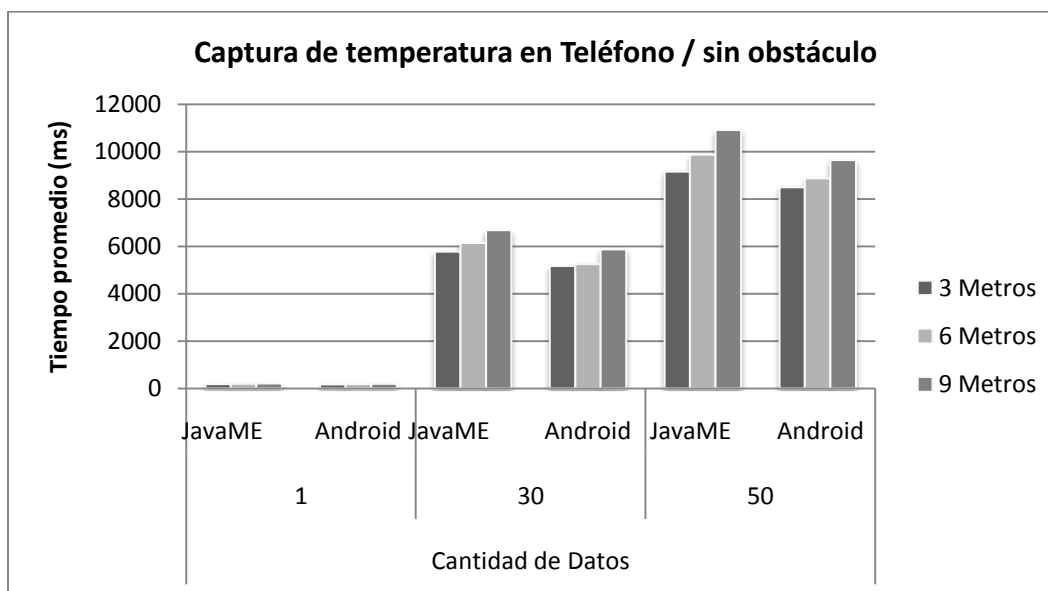


Figura 5. 7 Gráfica del promedio de tiempo la captura de temperatura a diferentes distancias con los teléfonos móviles **sin obstáculo**

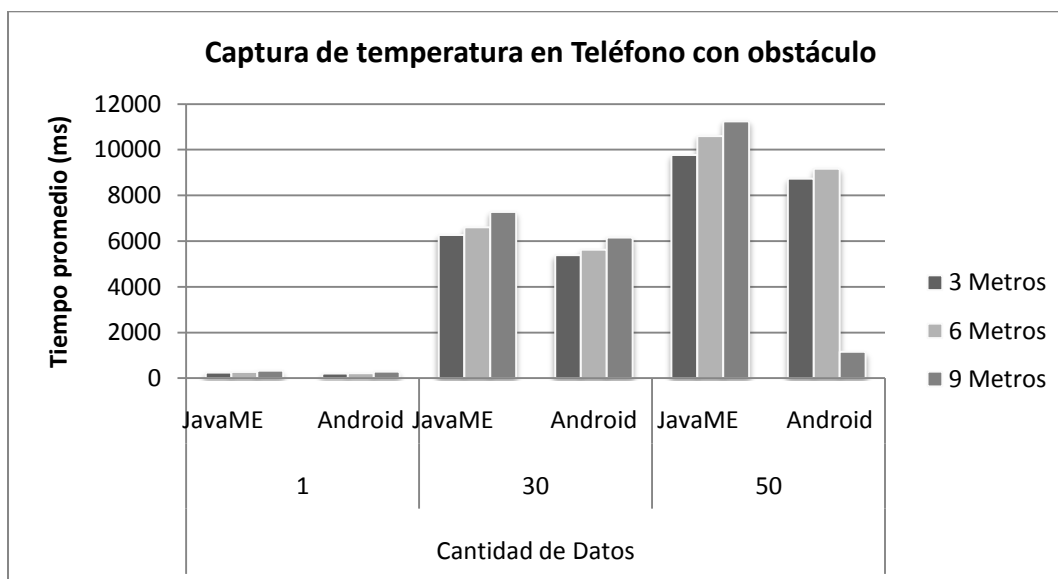


Figura 5. 8 Gráfica del promedio de tiempo la captura de temperatura a diferentes distancias con los teléfonos móviles **con obstáculo**

Para complementar estos resultados se presentan los promedios de intentos de conexión se obtuvieron tomando las temperaturas a diferentes distancias, estos resultados se presentan en la tabla 5.4, donde podemos ver que los intentos van incrementando conforme se aumenta la distancia entre el dispositivo móvil y el sensor médico y cuando se presenta algún obstáculo. Sin embargo, como mencionaba anteriormente esto puede suceder debido al modulo de interfaz BT que presenta cada equipo celular.

INTENTOS DE CONEXIÓN BT entre dispositivo móvil y sensor médico						
Sin obstáculo						
Cantidad Datos	1		30		50	
Distancia	JavaME	Android	JavaME	Android	JavaME	Android
3 Metros	0.13	0.06	0.13	0.06	0.13	0.06
6 Metros	0.20	0.13	0.20	0.133	0.20	0.13
9 Metros	0.33	0.26	0.33	0.26	0.33	0.26
Con obstáculo						
Cantidad Datos	1		30		50	
Distancia	JavaME	Android	JavaME	Android	JavaME	Android
3 Metros	0.20	0.13	0.20	0.13	0.20	0.13
6 Metros	0.33	0.26	0.33	0.26	0.33	0.26
9 Metros	0.53	0.40	0.53	0.40	0.53	0.40

Tabla 5.4 Promedio de intentos de conexión entre el dispositivo móvil y el sensor médico.

5.3.2 *Análisis de líneas de código*

En una comunicación Bluetooth existe un dispositivo que ofrece un servicio (servidor) y otros dispositivos acceden a él (clientes). Dependiendo de que parte de la comunicación debamos programar deberemos realizar una serie de acciones diferentes (Galvez & Ortega, 2007).

Un cliente Bluetooth debe realizar lo siguiente:

- Búsqueda de dispositivos.
- Búsqueda de servicios
- Establecimiento de la conexión
- Comunicación

Para un servidor Bluetooth debe realizar las siguientes operaciones:

- Crear una conexión servidora

- Especificar los atributos de servicio
- Abrir las conexiones clientes

En este caso de estudio sólo fue necesario programar las operaciones del servidor, ya que la conexión Bluetooth se realiza directamente utilizando la dirección MAC del sensor, esta conexión BT produce para cada plataforma un numero de líneas diferentes, estos resultados se presentan en la tabla 5.5.

Para Java SE ME utilizó la API *JSR 82*, la cual surgió para estandarizar el desarrollo de aplicaciones Bluetooth utilizando Java. El objetivo de ésta especificación fue definir un API estándar abierto, no propietario que pudiera ser usado en todos los dispositivos que implementen Java ME. Por consiguiente fue diseñado usando los API's Java ME y el entorno de trabajo CLDC/MIDP (Froupe Quintas et al, 2006)

La API Java para Bluetooth definen dos paquetes que dependen del paquete CLDC `javax.microedition.io`, que fueron utilizadas en el desarrollo de la aplicación:

- `javax.bluetooth`
- `javax.obex`

En el desarrollo de la aplicación en la plataforma Android se tiene la API para Bluetooth esta disponible en el paquete `Android.bluetooth`, dentro del cual se tienen las clases e interfaces para establecer la comunicación Bluetooth. De las cuales se utilizaron: `BluetoothAdapter`, `BluetoothDevice`, `BluetoothHealth`, `BluetoothHealthCallback`, `BluetoothProfile.ServiceListener`.

Y en las figuras 5.9 y 5.10 se presentan las líneas de códigos utilizadas para dicha conexión.

Plataforma Móvil	No. de Líneas de Código
Java ME	9
Android	12

Tabla 5.5 Líneas de código para conexión Bluetooth

```

public void conectaDispositivo() throws IOException{
    try {
        streamConnection = (StreamConnection)Connector.open("btspp://00066602D100:1;master=false;authenticate=false;encrypt=false");
        in = streamConnection.openDataInputStream(); // to read from the bluesentry
        out = streamConnection.openDataOutputStream(); // to write to the bluesentry; dont forget to flush after each write/
    }
    catch (Exception e){
        System.out.println("exception "+ e);}
}

```

Figura 5.9 Líneas de código para la conexión BT en Java ME

```

void conectaDispositivo(){
    try {
        socket = dispositivo.createRfcommSocketToServiceRecord(UUID.fromString("00001101-0000-1000-8000-00066602D100"));
        socket.connect();
        ins = socket.getInputStream();
        ons = socket.getOutputStream();
    }
    catch(Exception ex){
        error = ex.toString();
        println(error);
    }
}

```

Figura 5.10 Líneas de código para la conexión BT en Android

Las plataformas manejan de manera similar comunicación Bluetooth, en lo que varían es en que Android es necesario habilitar dentro del *manifest* de nuestra aplicación al menos uno de los permisos para poder acceder a la API de Bluetooth; los permisos disponibles son: *android.permission.BLUETOOTH* (permite únicamente realizar conexiones BT y transferir datos) y *android.permission.BLUETOOTH_ADMIN* (que permite, además de realizar conexiones BT y transferencias de datos, manipular las opciones del sistema en lo referente a Bluetooth, buscar otros dispositivos y realizar vínculos).

5.4 RESULTADOS de la evaluación del ESCENARIO 2.

5.4.1 Tiempos de conexión HTTP

A partir del escenario 2 que se explicó en la sección 5.2, se obtuvieron los resultados que a continuación se presentan.

En la tabla 5.6 se presentan los resultados del tiempo promedio de conexión del emulador y dispositivo móvil al servidor Web, con las 2 diferentes redes inalámbricas anteriormente mencionadas.

CONEXIÓN-Emulador			
Red Inalámbrica	javax.microedition.io /java.io	Apache.Org.*	Java.net
Red 1	140.26	439.93	166.73
Red 2	151.13	438.4	184.26
CONEXIÓN-Dispositivos Móviles			
Red Inalámbrica	javax.microedition.io /java.io	Apache.Org.*	Java.net
Red 1	521.13	482.86	468.6
Red 2	593	549.46	530.33

Tabla 5.6 Tiempos promedio (ms) de conexión al servidor web

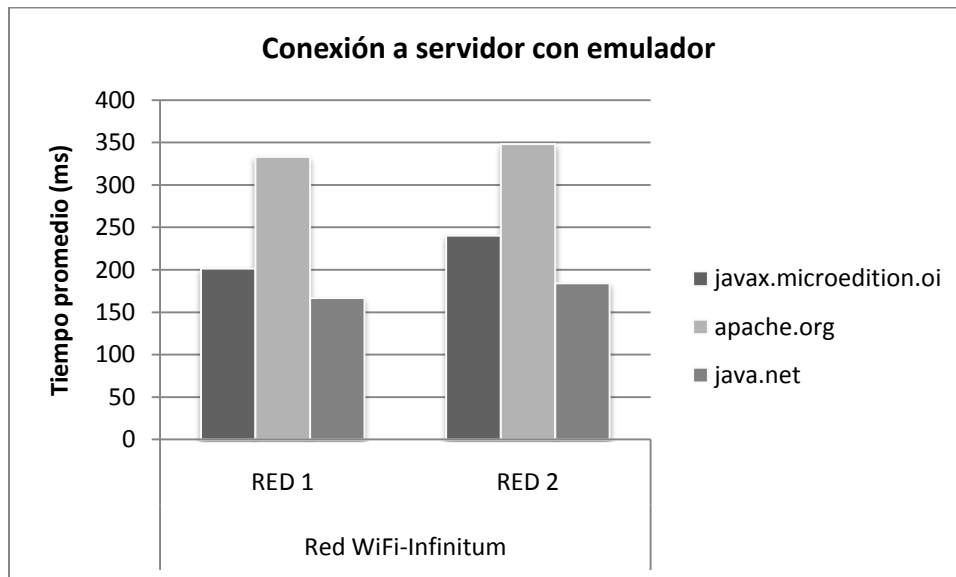


Figura 5.11 Gráfica de los tiempos de conexión al servidor con emulador utilizando 2 redes WiFi infinitum.

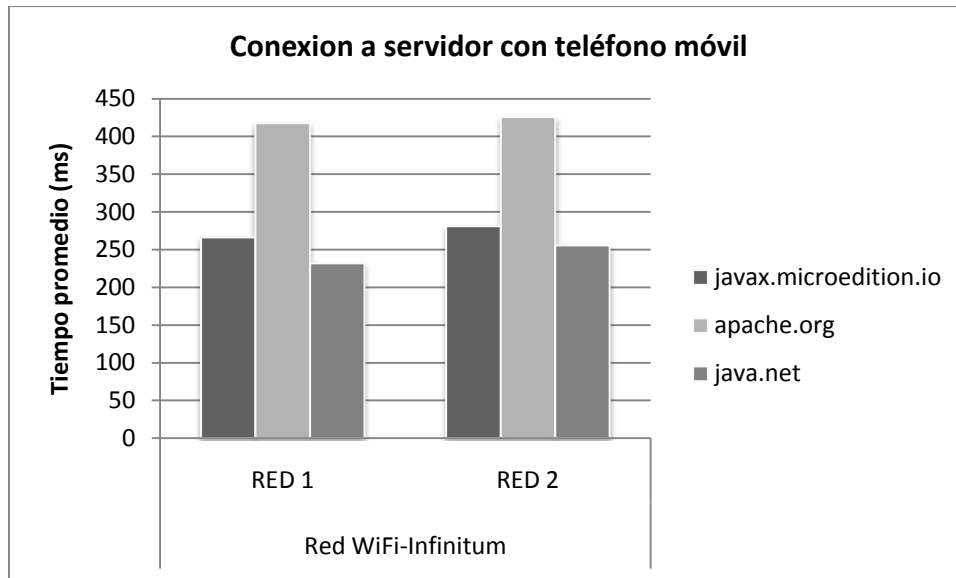


Figura 5. 12 Gráfica de los tiempos (ms) de conexión al servidor con los dispositivos móviles utilizando 2 redes WiFi infinitem.

Como se observa en las figuras anteriores los tiempos de conexión al servidor obtenidos son menores utilizando la **red 1: infinitem de CASA**. Sin embargo el comportamiento de los paquetes de las plataformas son los mismos en ambas redes WiFi, es decir, utilizando los emuladores los tiempos promedios menores fueron utilizando el paquete *javax.microedition.io* de la plataforma Java ME; mientras que en los teléfonos móviles el menor tiempo promedio de conexión al servidor fue al utilizar el paquete *java.net* de la plataforma Android.

Además de lo anteriormente mencionado, también se puede apreciar que el comportamiento del paquete de Java ME tuvo una gran variación en sus tiempos al utilizarse en el emulador y el teléfono móvil, mientras los paquetes de Android se comportaron de manera similar. Esta diferencia se presenta debido a los distintos SO que utilizan los teléfonos móviles. A lo cual podemos decir que los paquetes de Android presentan mayor estabilidad en diferentes SO.

Los promedios de tiempos de la lectura y escritura al servidor se muestran en las tablas 5.7 y 5.8 respectivamente, así mismo se presentan los resultados de manera gráfica en las figuras 5.13, 5.14, 5.15 y 5.16.

LECTURA-Emulador			
Red Inalámbrica	<i>javax.microedition.io</i> <i>/java.io</i>	<i>Apache.Org.*</i>	<i>Java.net</i>
Red 1	163.96	456.21	176.73
Red 2	186.56	489.93	201.33
LECTURA-Dispositivos móviles			
Red Inalámbrica	<i>javax.microedition.io</i> <i>/java.io</i>	<i>Apache.Org.*</i>	<i>Java.net</i>
Red 1	261.83	521.98	248.26
Red 2	283.18	598.36	263.40

Tabla 5.7 Tiempo promedio (ms) de lectura al servidor Web

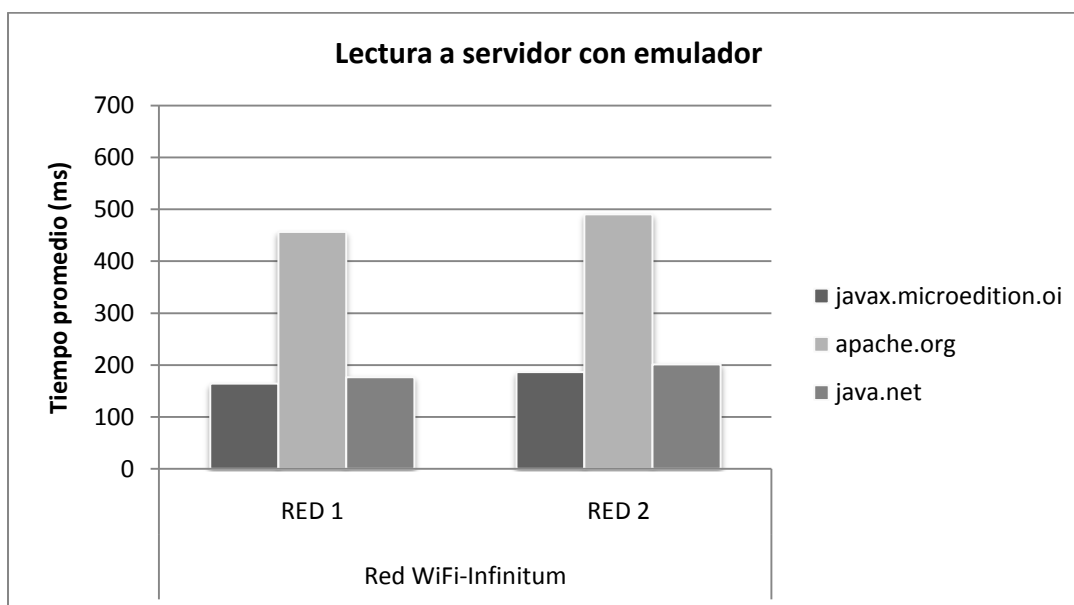


Figura 5.13 Gráfica de los tiempos de lectura al servidor con el emulador utilizando 2 redes WiFi infinitum.

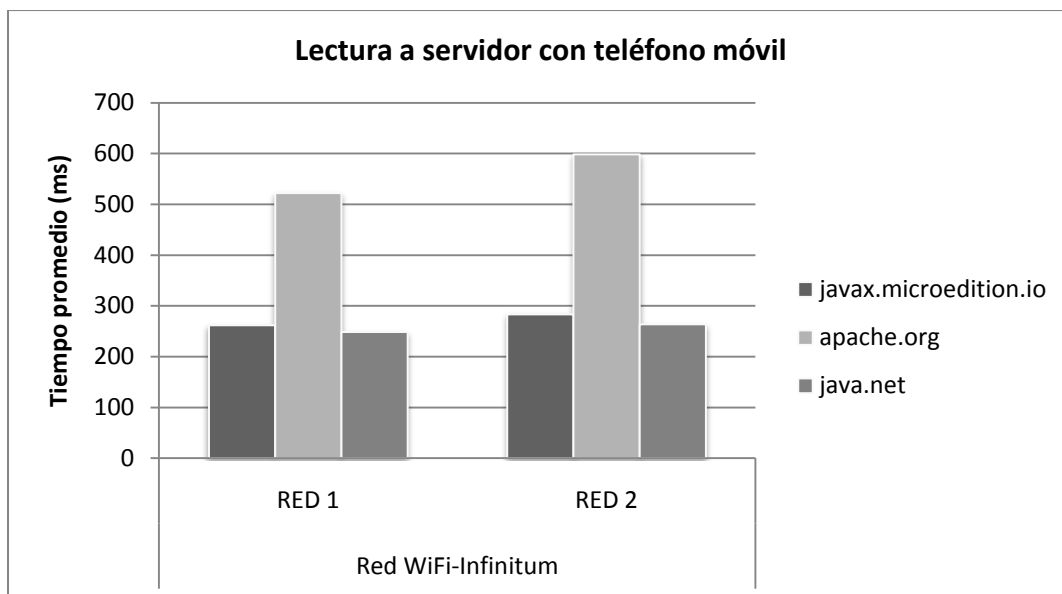


Figura 5.14 Gráfica de los tiempos (ms) de lectura al servidor con los teléfonos móviles utilizando 2 redes WiFi infinitum.

ESCRITURA-Emulador			
Red Inalámbrica	<i>javax.microedition.io</i> <i>/java.io</i>	<i>Apache.Org.*</i>	<i>Java.net</i>
Red 1	158.98	348.13	203.66
Red 2	178.33	394.26	211.73
ESCRITURA-Dispositivos móviles			
Red Inalámbrica	<i>javax.microedition.io</i> <i>/java.io</i>	<i>Apache.Org.*</i>	<i>Java.net</i>
Red 1	233.56	417.52	228.21
Red 2	262.83	439.79	251.56

Tabla 5.8 Tiempo promedio de escritura al servidor Web.

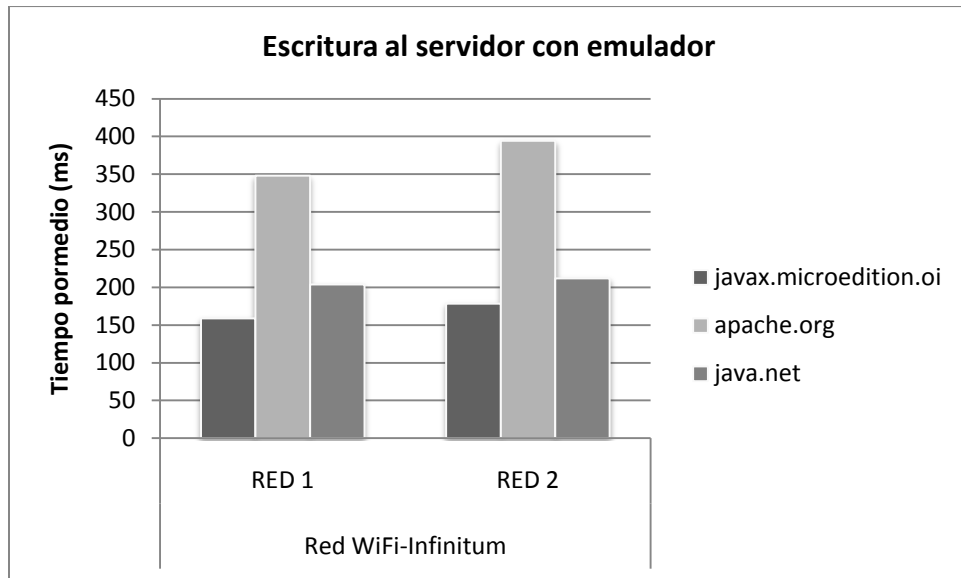


Figura 5.15 Gráfica del promedio de tiempos (ms) de escritura con el emulador al servidor utilizando 2 redes WiFi infinitem.

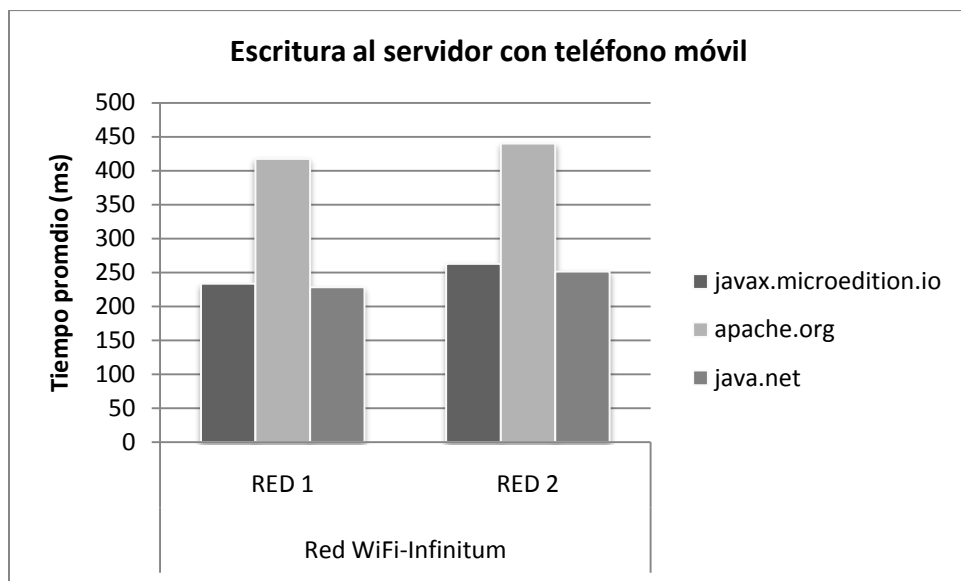


Figura 5.16 Gráfica del promedio de tiempos (ms) de escritura el teléfono móvil al servidor utilizando 2 redes WiFi infinitem.

En las gráficas presentadas anteriormente (fig. 5.15 y 5.16) tanto en los promedios de tiempo para la de lectura y escritura se presentan el mismo comportamiento que se mencionaba en la conexión al servidor.

5.4.2 Análisis de líneas de código

Para esta evaluación se realizó una comunicación al servidor de manera común, primeramente se inicia con la conexión al servidor y se crea un objeto de conexión *stream*, el cual permite enviar y recibir datos. Para los 3 diferentes paquetes que se utilizaron se necesitan convertir el objeto que se regresa. En la tabla 5.9 se muestra el no. las líneas de código necesarias para la comunicación al servidor y en las figuras 5.17, 5.18 y 5.19 se presentan las líneas de códigos utilizadas para dicha comunicación.

Plataforma móvil	No. de líneas de código
Javax.microedition.io (Java ME)	23
Apache.org (Android)	20
Java.net (Android)	25

Tabla 5.9 Líneas de código para la comunicación al servidor

```
public void run() {
    System.out.println(url + "?" + id + " en run de request");
    HttpURLConnection httpConn = null;
    InputStream is = null;
    OutputStream os = null;
    StringBuffer b = new StringBuffer("");
    try {
        httpConn = (HttpURLConnection)Connector.open(url);
        httpConn.setRequestMethod(HttpURLConnection.POST);
        httpConn.setRequestProperty("User-Agent", "Profile/MIDP-1.0 Configuration/CLDC-1.0");
        httpConn.setRequestProperty("Accept-Language", "en-US");
        httpConn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        os = httpConn.getOutputStream();
        os.write(id.getBytes());
        is = httpConn.openDataInputStream();
        int chr;
        while ((chr = is.read()) != -1) { // leer caracteres de lo que muestra servidor
            b.append((char) chr);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    try {
        if (is != null)
            is.close();
        httpConn.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    res = b.toString().trim();
}
```

Figura 5.17 Líneas de código para la comunicación con el servidor con el paquete *javax.mocroedition.io* de Java ME

```
public void run() {
    BufferedReader in=null;
    HttpClient client = new DefaultHttpClient();///1
    HttpGet httpget = new HttpGet(URL);///2
    try {
        HttpResponse response = client.execute(httpget);///3
        in = new BufferedReader(new InputStreamReader(response.getEntity().getContent()));
        StringBuffer sb = new StringBuffer("");
        String line = "";
        while ((line = in.readLine()) != null) {
            sb.append(line);
        }
        in.close();
        resultado = sb.toString();
    } catch (IOException ee) { Log.d("HTTP: ", "No se puede leer la información");}
    if (in != null) {
        try {
            in.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Figura 5.18 Líneas de código para la comunicación con el servidor con el paquete *org.apache.** de Android

```
public class getHttpResponse {
    public String getHttp(String location, String parametro) {
        String result = null;
        URL url = null;
        try {
            url = new URL(location+parametro);
        } catch (MalformedURLException e) {
        }
        if (url != null) {
            try {
                HttpURLConnection urlConn =(HttpURLConnection) url.openConnection();
                BufferedReader in = new BufferedReader( new InputStreamReader(urlConn.getInputStream()));
                StringBuffer sb = new StringBuffer("");
                String inputLine;
                int lineCount = 0;
                while ((inputLine = in.readLine()) != null) {
                    sb.append(inputLine);
                }
                in.close();
                result = sb.toString();
                urlConn.disconnect();
            } catch (IOException e) {
            }
        }
        return result;
    }
}
```

Figura 5.19 Líneas de código para la comunicación con el servidor con el paquete *java.net* de Android

5.5 Resumen

Este capítulo presenta la metodología que se utilizó para evaluar las plataformas de programación Java Me y Android, donde se establecieron 3 escenarios para realizar las evaluaciones que apoyan a obtener los resultados deseados para lograr la evaluación de dichas plataformas. Los resultados que se obtuvieron se dividieron en dos partes: primero resultados cuantitativos, donde se presentan los promedios de tiempos de comunicación BT y HTTP; y segundo resultados cualitativos, donde se dan a conocer las opiniones de los usuarios que utilizaron la aplicación en ambas plataformas. Dentro de este capítulo solo se dan a conocer los resultados cuantitativos de los cuales se observó que en comunicación BT entre el dispositivo móvil y el sensor médico la plataforma Android presentó mayor estabilidad que Java ME, mientras que en la comunicación HTTP entre el dispositivo móvil y el servidor utilizando el paquete org.apache de Android los tiempos fueron mayores que al utilizar el paquete de java.net también de Android. Para el paquete javax.microedition.io de Java ME los tiempos fueron menores al ser utilizado en el emulador, pero no sucedió lo mismo al ser utilizado en el dispositivo móvil real. Con lo cual podemos decir que la plataforma Android es más estable al realizar la comunicación BT y HTTP.

CAPITULO VI

6. EVALUACIÓN CUALITATIVA

6.1 Evaluación del Escenario 3: aceptación del usuario

La evaluación del escenario 3 fue diseñado con el propósito de cubrir las métricas establecidas en el capítulo 4 para el módulo “Interfaz de usuario” definido en el capítulo 3 para obtener los resultados de las opiniones de usuarios que al utilizar la misma aplicación desarrollada en las plataformas Java ME y Android.

En esta evaluación se utilizó la aplicación de salud móvil “Frecuencia Cardiaca Móvil (FCM)” (Alaníz Sida, 2011), desarrollada en la plataforma Android, la cual cumple con los módulos establecidos en capítulo 3. Como parte de este trabajo la aplicación FCM posteriormente fue desarrollada en la plataforma Java ME para poder realizar la prueba de aceptación por parte del usuario. En la figura 6.1 se presenta el escenario 3 utilizado para esta evaluación.

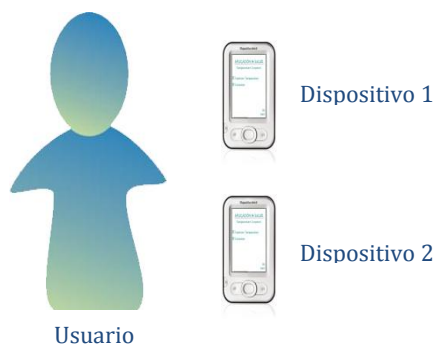


Figura 6.1 Escenario 3. Aceptación del usuario

Para la presentación de gráficos se utiliza la herramienta “Google Chart Tools” (Google, Google Chart Tools, 2010), la cual es una API de Google, que permite crear potentes y atractivos gráficos de gran alcance, fácil de usar y gratuito. En (Google, Google Chart Tools, 2010) se presentan varios ejemplos que muestran no solo la forma de implementarlo si no también de aclarar a los usuarios cual es más conveniente utilizar de acuerdo a sus necesidades. Para poder mostrar estas gráficas de resultados en el móvil fue necesario

cargar las bibliotecas mediante 2 scripts, las cuales permiten cargar la visualización de Google y las bibliotecas de gráficos. Estos scripts ya están definidos.

Una vez desarrollada la aplicación FCM en la plataforma Java Me, se diseñó una encuesta que nos permitiera conocer las opiniones de los usuarios una vez que tuvieran la experiencia de utilizar la aplicación en ambos dispositivos.

La encuesta que se aplicó fue diseñada tomando en cuenta las preguntas que se presentan en 2 de los cuestionarios de usabilidad estandarizados: SUS (por sus siglas en inglés System Usability Scale) y CSUQ (por sus siglas en inglés Computer System Usability Questionnaire). El cuestionario SUS desarrollado por Digital Equipment Corp., consiste en 10 preguntas; cada pregunta es una afirmación y tienen una calificación en una escala de cinco puntos desde "Totalmente en desacuerdo" hasta "Totalmente de acuerdo". Por otra parte el cuestionario CSUQ desarrollado por IBM, consiste en 19 preguntas; cada pregunta es una afirmación y tienen una calificación en una escala de siete puntos desde "Totalmente en desacuerdo" hasta "Totalmente de acuerdo". En (Tullis & Stetson, 2004) se hace un estudio de 4 cuestionarios de usabilidad estandarizados, donde se obtiene que la aplicación de los cuestionarios SUS y CSUQ dan resultados muy similares de las opiniones de los usuarios. Por tal motivo el cuestionario que se aplicó a los usuarios en este trabajo fue una combinación de SUS y CSUQ. (Ver anexo 1)

En la evaluación del escenario 3 además de obtener las opiniones de los usuarios nos permite realizar la evaluación de las herramientas o funcionalidades que tiene cada plataforma para el desarrollo de las Interfaces de Usuario, el almacenamiento Persistente y la presentación de la información de manera gráfica.

6.2 RESULTADOS de la evaluación del ESCENARIO 3

6.2.1 Aceptación del Usuario

A partir del escenario 3 descrito en la sección 6.1 de este capítulo se obtuvieron los resultados que se presentan en este capítulo.

Se seleccionó una muestra de usuarios para probar la aplicación FCM desarrollada en ambas plataformas, la única restricción que se tuvo para esta selección de usuarios fue que supieran manejar un teléfono móvil.

De las 26 personas o usuarios encuestados se tiene que el 50% fueron mujeres y el 50% hombres; de los cuales solo 4 personas (2 hombres y 2 mujeres) indicaron que tenían alguna enfermedad, siendo estas las enfermedades de Diabetes y Taquicardia. La edad promedio de los encuestados fue 34.88 años y la escolaridad de los mismos fue a nivel universitario de 42%, a nivel preparatoria de 35% y a nivel secundaria de 23%, esto se muestra de manera grafica en las figuras 6.2, 6.3.

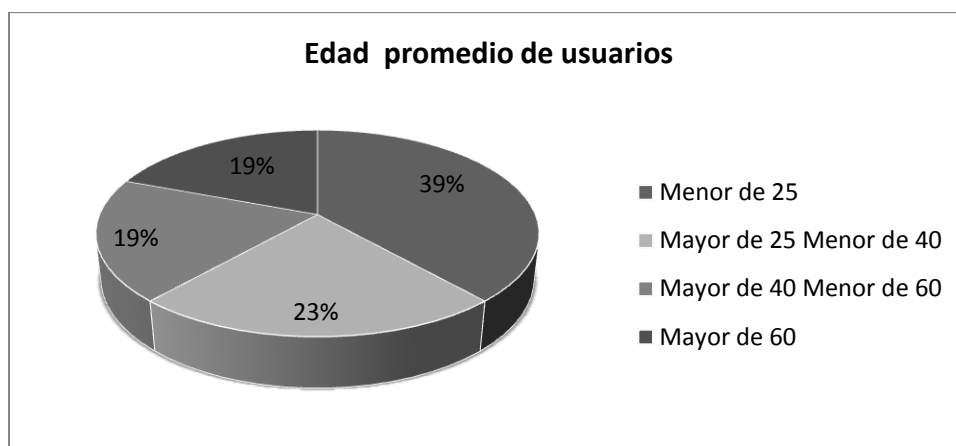


Figura 6.2 Edad Promedio de los Encuestados

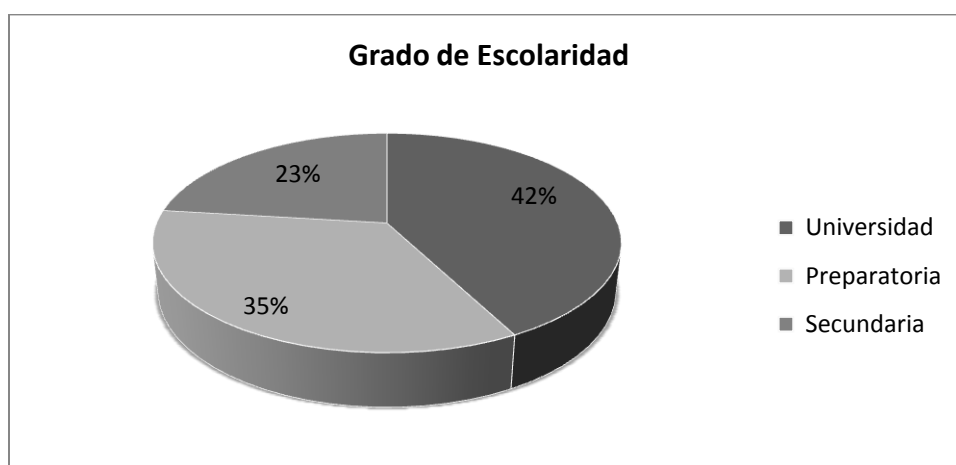


Figura 6.3 Escolaridad de encuestados

En las figuras 6.4 y 6.5 se muestra de manera gráfica la utilización de los diferentes dispositivos móviles existentes en el mercado, así como el sistema operativo (SO) que cada uno de ellos utiliza, respectivamente.

Como se puede observar el 20% de los encuestados desconocen el sistema operativo que tiene su teléfono móvil y el SO más utilizado es Symbian siendo los menos utilizados Android y Windows Mobile.

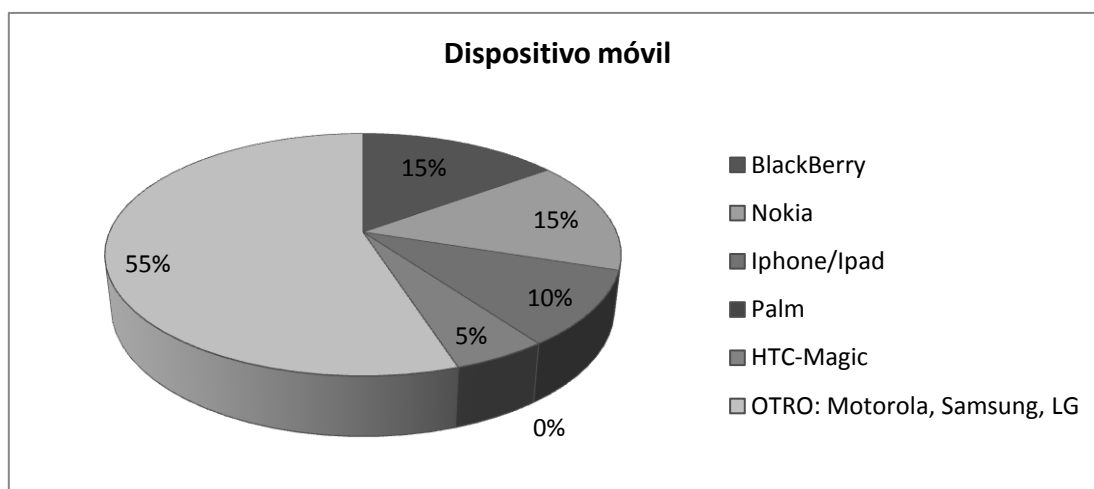


Figura 6.4 Tipo de Dispositivo móvil que utilizan

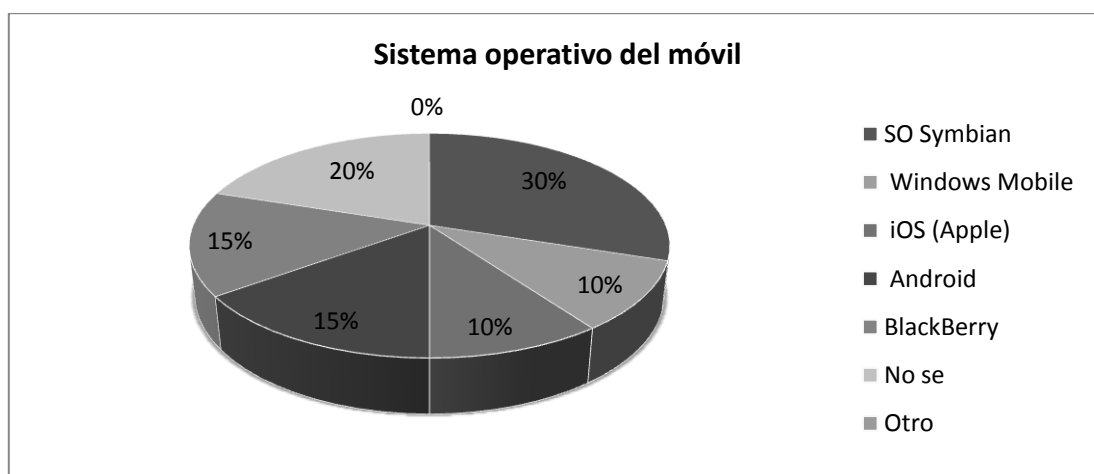


Figura 6.5 Sistema operativo del dispositivo móvil

En la tabla 6.1 se muestran los resultados de las preguntas que se les realizaron a los usuarios después de utilizar la aplicación FCM en ambos dispositivos móviles. En donde se puede observar algunas diferencias importantes que se obtuvieron; por ejemplo 80.77% de los usuarios no consideran que necesitan mucho tiempo para aprender antes de utilizar la aplicación en el dispositivo 2 contra un 76.92% no lo consideran así en el dispositivo 1. Otra diferencia importante se encontró en la pregunta 9 donde el 73.02% de los usuarios les gusto más la utilización en la aplicación que se encontraba en el dispositivo 2 contra un 26.92% que solo les gustó ligeramente la interfaz en el dispositivo 1.

RESULTADOS DE LA ENCUESTA					
DISPOSITIVO 1	Totalmente en Desacuerdo	Ligeramente Desacuerdo	Neutral	Ligeramente de Acuerdo	Totalmente de Acuerdo
1. Considero necesito mucho tiempo para aprender antes de utilizar la aplicación con eficacia	0%	76.92%	11.54%	11.54%	0%
2. Considero que la aplicación es excesivamente compleja	76.92%	23.08%	0%	0%	0%
3. Considero que la aplicación es fácil de usar	0%	0%	26.92%	61.54%	11.54%
4. Considero que sería necesario soporte/ayuda para poder utilizar esta aplicación	0%	34.61%	53.85%	11.54%	0%
5. Considero simple las opciones que presenta la aplicación	0%	0%	15.38%	57.69%	26.92%
6. Considero que la información proporcionada por la aplicación es fácil de entender	0%	0%	30.77%	53.85%	15.38%
7. Considero que la organización de la información en las páginas de la aplicación (pantallas) es claro	0%	38.46%	61.54%	0%	0%
8. Considero que la interfaz de esta aplicación es agradable	0%	26.92%	57.69%	15.38%	0%

9. Me gusto usar la interfaz de esta aplicación	0%	42.31%	30.77%	26.92%	0%
DISPOSITIVO 2	Totalmente en Desacuerdo	Ligeramente Desacuerdo	Neutral	Ligeramente de Acuerdo	Totalmente de Acuerdo
1. Considero necesito mucho tiempo para aprender antes de utilizar la aplicación con eficacia	11.54%	80.77%	7.69%	0%	0%
2. Considero que la aplicación es excesivamente compleja	80.77%	19.23%	0%	0%	0%
3. Considero que la aplicación es fácil de usar	0%	0%	11.54%	69.23%	19.23%
4. Considero que sería necesario la ayuda de un experto para poder utilizar esta aplicación	0%	46.15%	30.77%	23.08%	0%
5. Considero simple las opciones que presenta la aplicación	0%	0%	23.08%	34.62%	42.30%
6. Considero que información proporcionada por la aplicación es fácil de entender	0%	0%	23.08%	38.46	38.46
7. Considero que la organización de la información en las páginas de la aplicación (pantallas) es claro	0%	0%	11.54%	46.15%	42.31%
8. Considero que la interfaz de esta aplicación es agradable	0%	0%	0%	38.46%	61.54%
9. Me gusto usar la interfaz de esta aplicación	0%	0%	0%	26.92%	73.08%

Tabla 6.1 Resultados de Encuesta de Aceptación del Usuario

Además de estas 9 preguntas se realizó una última pregunta a los usuarios: Si usted tuviera la necesidad de utilizar una aplicación para el cuidado de su Salud ¿En qué Dispositivo lo preferiría utilizar?, de este cuestionamiento se obtuvo que un 77 % de los usuarios preferirían utilizar el Dispositivo 2 contra un 23 % que preferirían utilizar el Dispositivo 1.

6.2.2 Análisis de líneas de Código

La tabla 6.2 muestra el tamaño (en kilobytes) del archivo de instalación, el número de archivos generados y el total de las líneas de código en cada plataforma. El archivo de instalación es el que los usuarios deben descargar e instalar en sus respectivos teléfonos móviles. Donde podemos ver que el tamaño de la aplicación desarrollada en la plataforma Java ME es tres veces menor que el tamaño de la aplicación desarrollada en la plataforma de Android.

Plataforma móvil	Tamaño de la aplicación en Kb	No. de archivos	No. total del líneas de código
Java ME	239	5 archivos .java	Java- 429
Android	820	6 archivos .java 9 archivos .xml	Java- 1035 XML- 125

Tabla 6. 2Tamaño de la aplicación en kilobytes

6.3 Aplicación desarrollada en las plataformas Java ME y Android

La aplicación FCM desarrollada en las plataformas Java ME y Android presenta interfaces con diferencias muy marcadas entre ellas, hablando tanto de programación como de diseño de interfaces. Es importante dar a conocer estas diferencias a los programadores de aplicaciones móviles.

6.3.1 Interfaz de Usuario en Java

Java ME está compuesta de un conjunto de APIs (Application Programming Interface) separadas en dos niveles: *Configuraciones* y *Perfiles*. El primero de ellos, *Configuraciones*, define las clases básicas tomadas de la versión estándar de Java. La configuración que corresponde a los dispositivos pequeños es conocida como CLDC (Connected Limited Device Configuration). El segundo nivel, *Perfiles*, se refiere a las clases específicas creadas para los dispositivos móviles, es decir define el conjunto básico que corresponde a un grupo particular de dispositivos (Knudsen, 2003). Un *Perfil* tiene dos propósitos básicos: La especialización en un tipo de dispositivo y garantizar la

compatibilidad entre dispositivos similares. El perfil MIDP (Mobile Information Device Profile) está especialmente diseñado para teléfonos celulares y es además soportado por algunos PDAs. MIDP contiene paquetes de clases definidas para realizar tareas específicas. Especialmente útiles para el desarrollo de servicios móviles son las clases que permiten realizar conexiones remotas y administrar el sistema de almacenamiento persistente de datos en los dispositivos. Las aplicaciones basadas en MIDP son conocidas como *MIDlets* y pueden ser instaladas directamente en los dispositivos móviles que soportan este perfil.

En esta plataforma la programación de la interfaz gráfica es realizada directamente en Java, utilizando el paquete *javax.microedition.lcdui*, el cual contiene las clases y métodos requeridos para implementar una interfaz de usuario. Cada una de las pantallas debe ser invocada a través de un MIDlet, los cuales son los componentes a ejecutar directamente en un proyecto. Cada MIDlet contiene un objeto Display en el cual puede mostrar uno o más objetos del tipo Displayable, como se muestra en la figura 6.6.

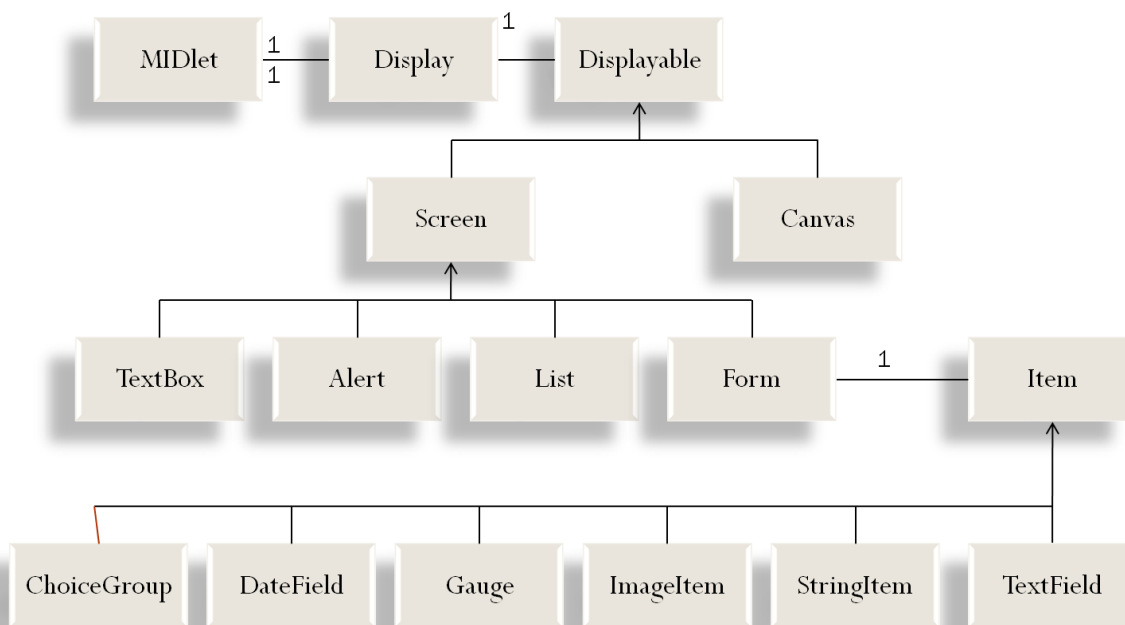


Figura 6. 6 Estructura de la interfaz de usuario en Java ME.

6.3.2 *Interfaz de Usuario Android*

Por otro lado, la plataforma Android permite el desarrollo de aplicaciones en lenguaje Java estándar soportando una gran cantidad de paquetes no disponibles en Java ME. La plataforma incluye además un conjunto muy extenso de nuevas y novedosas APIs. De igual modo provee utilerías que permiten hacer uso de las aplicaciones incluidas por defecto en todos los teléfonos Android.

La principal diferencia en el diseño de interfaces gráficas en Android, es que estas pueden ser realizadas únicamente en Java o utilizando XML, siendo esta última opción la recomendada en las especificaciones de Android. Los objetos utilizados para la implementación de interfaces gráficas son las *Actividades*. Una *Actividad* puede mostrar un conjunto de *Vistas*, las cuales son objetos que permiten la interacción entre los usuarios y el dispositivo. Cada *Actividad* debe ser declarada en el archivo *AndroidManifest.xml*. El manejador de proyectos de Android se encarga de mantener una pila de las actividades visitadas por el usuario, de manera que el programador no debe preocuparse por establecer las secuencias entre ellas. Al compararlo con Java Me, Android proporciona más objetos visuales conocidos como *widgets*, que permiten la creación de interfaces gráficas más ricas y atractivas. Al usar la programación XML es posible definir la estructura y organización de los elementos en pantalla, pudiendo organizarlos de manera horizontal, vertical, en una tabla, etc. Para ello se requiere generar un archivo XML para cada *Actividad* contenida en el proyecto. Cada uno de los elementos contenidos en los archivos XML tiene una referencia en Java generada automáticamente en el archivo R.Java, el cual puede ser utilizado para interactuar con dichos elementos dentro de las aplicaciones (Vázquez, Méndez, Nieto, Jimenez, & Sanchez, 2010). Como se muestra en la figura 6.7.

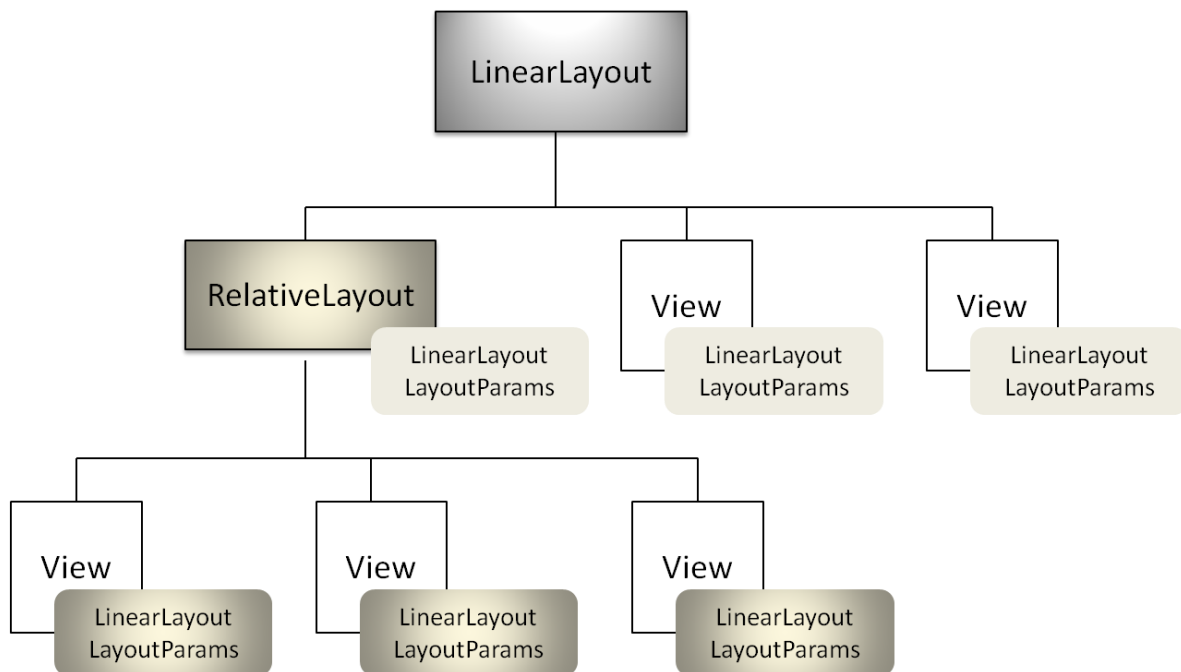


Figura 6.7 Estructura de la interfaz de usuario en Android

6.3.2 Resultados de las Interfaz de Usuarios creadas en las plataformas Java ME y Android.

A continuación se muestran las interfaces con las que los usuarios interactuaron en ambos dispositivos. En las figuras 6.8, 6.9 y 6.10 se muestran las pantallas creadas en la plataforma Java ME.

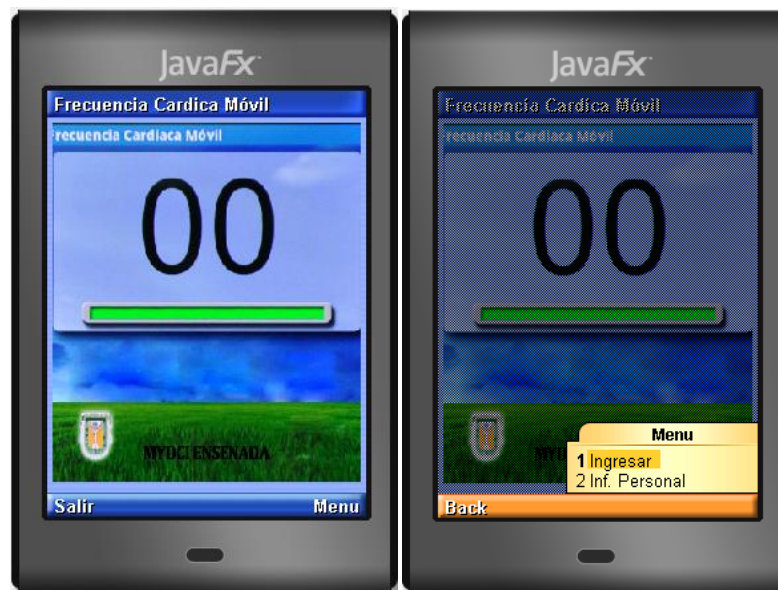


Figura 6.8 Pantalla Principal en la Plataforma Java ME



Figura 6.9 Menús creados en Java ME



Figura 6.10 Resultados de las opciones del menú

Por otra parte en las Figuras 6.11, 6.12 y 6.13 se muestran las Pantallas creadas en la plataforma ANDROID.

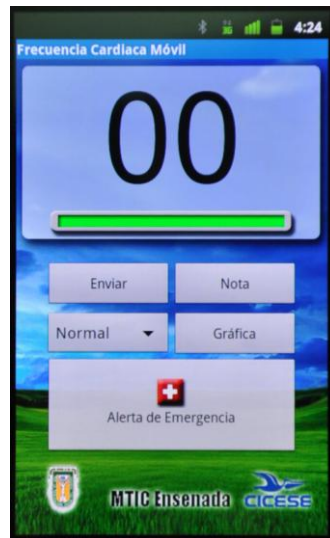


Figura 6.11 Pantalla principal en la plataforma Android

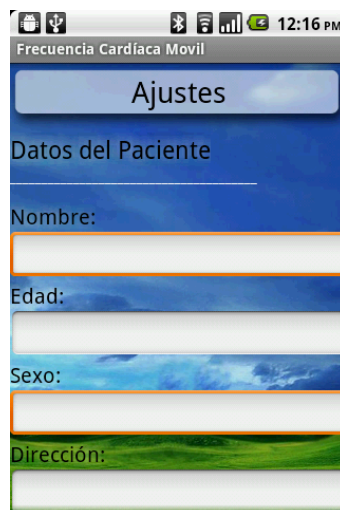


Figura 6.12 Ajuste/ captura de datos del paciente

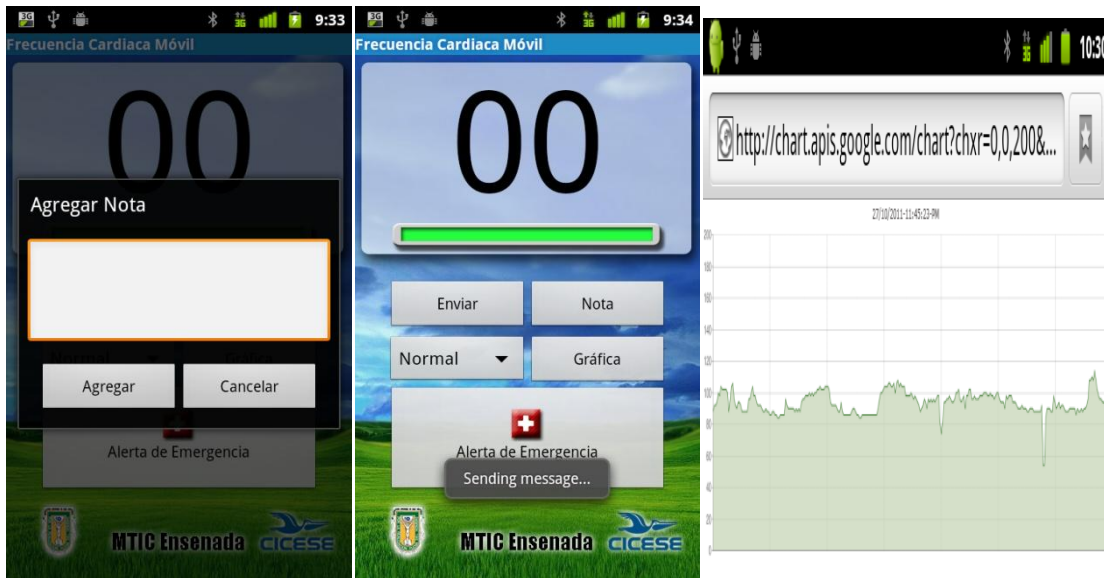


Figura 6.13 Resultados obtenidos del menú

Como se puede ver en las figuras anteriores, la aplicación desarrollada en la plataforma Java ME contiene más pantallas por las cuales se tiene que navegar para capturar la frecuencia cardíaca comparada con la aplicación realizada en la plataforma Android que todas las opciones se presentan en la pantalla inicial y desde ahí se captura la frecuencia cardíaca. También otra diferencia que se puede observar es que las opciones presentadas en la aplicación desarrollada en Java ME se muestran en un Menú, mientras que en la aplicación desarrollada en Android se muestran en manera de Botones. Estas diferencias se presentan debido a las diferentes funcionalidades y/o herramientas que presenta cada plataforma anteriormente mencionadas en esta sección.

6.4 Resumen

En este capítulo se presenta la evaluación del escenario 3, tanto la descripción del escenario como los resultados obtenidos al desarrollar la aplicación FCM en ambas plataformas, las cuales contienen diferentes funcionalidades y herramientas para el desarrollo de interfaces de usuario. Así mismo se presentan los resultados de las encuestas aplicadas a usuarios que utilizaron la aplicación en los dos dispositivos móviles correspondientes para cada plataforma Java ME y Android.

CAPITULO VII

7. RECOMENDACIONES

A pesar de que la programación de aplicaciones en las plataformas Java ME y Android está basada en el lenguaje Java Estándar, existen notables diferencias al llevar a cabo el desarrollo en una u en otra plataforma. Java ME utiliza un subconjunto limitado de Java Estándar y un conjunto de APIs propias de la plataforma que no son compatibles con Android. Android, por su parte, tiene soporte para una mayor cantidad de paquetes de Java Estándar además de sus propias APIs. Independientemente de que el programador debe familiarizarse con las nuevas APIs incluidas en la plataforma Android.

El desarrollo de una aplicación en Java ME se compone básicamente de los archivos .java y .class. Además de ellos se requiere un archivo *manifest.mf* el cual contiene un listado de las clases que conforman el proyecto de la aplicación. El archivo *manifest.mf* y las clases compiladas Java son empaquetadas en un archivo JAR, el cual es el que se instala en el dispositivo móvil. Por otra parte, para el desarrollo de aplicaciones en Android tiende a ser un poco más complicado debido a que se requiere de más archivos y directorios.

La programación de una aplicación en la plataforma Android se hace separando completamente la interfaz grafica así como las constantes del resto de la programación. Los archivos que contienen las interfaces graficas son conocidos como “Recursos”, los cuales son definidos mediante archivos en lenguaje XML, por lo que un proyecto de una aplicación contendrá un conjunto de archivos .java y un conjunto de archivos .xml. Una vez generados los *Recursos* utilizando el lenguaje XML, se crea el archivo R.java. Este archivo contiene las referencias a los objetos definidos en XML que serán utilizadas en la programación Java del proyecto. Un proyecto en Android también requiere un archivo manifiesto, conocido como *AndroidManifest.xml*. Este archivo es mucho más complejo que el utilizado en Java ME. En Android, este archivo define la estructura completa de la aplicación, el programador debe encargarse de modificar el archivo *AndroidManifest.xml* el cual es básico para el buen funcionamiento del proyecto. Finalmente la aplicación es

empaquetada en un archivo APK (Android Package), el cual debe ser instalado en el dispositivo para ejecutar la aplicación.

Se pudiera creer en un principio que es un poco mas complicado programar en Android debido a que tiene 4 bloques una aplicación en su ciclo de vida (*Activity, Intent, Service y Content Provider*), sin embargo estos bloques permiten una mayor organización y entendimiento en el desarrollo de las aplicaciones; además no es necesario que una aplicación completa incluya los 4 bloques. Por otra parte tenemos que en Java ME la clase *MiDlet* es la encargada del ciclo de vida de una aplicación, esta clase puede tener 3 estados posibles (*startApp, pauseApp y destroyApp*), los cuales deben estar en una aplicación para su ejecución. En mi opinión personal ninguno de los 2 me pareció difícil de aprender, sin embargo encuentro una mayor ayuda en la organización del desarrollo de la aplicación en la plataforma de Android.

En base a la bibliografía consultada y al desarrollo de las aplicaciones se clasificaron las herramientas que brindan ambas plataformas en los módulos que se determinaron debe contener una aplicación de Salud Móvil, que se presentan en la tabla 7.1.

Descripción	Java ME	Android	Evaluación
Comunicación con Sensores Médicos / Adquisición de Datos de los sensores médicos.			
Comunicación Bluetooth	API JSR-82 con los paquetes javax.bluetooth y javax.obex	Dentro de las API de Bluetooth de Android se encuentra el paquete android.bluetooth	Las plataformas presentan diferentes reglas de sintaxis que permiten realizar las acciones de los servicios de BT, pero la diferencia que se presenta entre ellas es que en Android es necesario declarar permisos en el archivo <i>manifest</i> para poder acceder a la API de Bluetooth.
Almacenamiento de la Captura de Datos.			
Comunicación Servidor	Paquetes javax.microedition.io y java.io	Provee numerosos paquetes para conexiones en red como apache.org, java.net	La programación en esta parte es prácticamente lo mismo en ambas plataformas, la única diferencia son los paquetes que utiliza cada plataforma.
Almacenamiento Persistente y Soporte de Base de Datos	RMS y PerstLite from mObject	Content Provider y APIs de Android contiene soporte para Datos SQLite	Android ofrece la clase android.content.ContentProvider para el manejo y almacenamiento de datos, conocido también como “proveedor de contenido”. Este bloque de construcción, permite compartir datos entre procesos y aplicaciones. Además, Android soporta BD SQLite y proporciona funciones de control que permiten almacenar datos complejos en forma de objeto. Java ME ofrece la librería javax.microedition.rms para

			memoria persistente, lo cual consiste en una base de datos binaria sencilla (sin usar SQL) orientada a guardar registros (RecordStore). Para esta aplicación que se desarrolló me parece que la librería ContentProvider es la adecuada, ya que se puede manejar como una base de datos, con los campos que sean necesarios almacenar, por otra parte la librería de Java SE ME puede utilizar en aplicaciones donde se requiera guardar poca información.
Interfaz de usuario			
Componentes de la Interfaz de Usuario (UI)	Componentes de alto nivel LCDUI, como el formulario o lista, de bajo nivel LCDUI para controlar cada pixel de interfaz de usuario, soporte para SVG (definido en JSR 287)	Los objetos View y ViewGroup, sirven para un rápido diseño de UI.	Android separa la programación de interfaces de usuario y los eventos o acciones de las aplicaciones. La interfaz de usuario es definida en código XML, y cada pantalla que se muestra al usuario tiene código XML diferente. El lenguaje XML permite realizar interfaces más completas y sencillas para el usuario. Por otro lado en Java ME la creación de las pantallas para los usuarios se encuentra dentro de todo el desarrollo de la aplicación. Otra Diferencia que se encuentra en la manera que se pueden diseñar los
Modelo de Eventos	Gestión de eventos mecanismo basado en objetos de comando	Hereda el modelo de eventos de Java, usa una clase especial (intent) que permite la respuesta de aplicación a los acontecimientos externos, tales como una llamada telefónica	

			comandos, estos en java se realizan utilizando únicamente menús dentro de la aplicación, mientras que en Android se pueden diseñar además en manera de botones.
--	--	--	---

Tabla 7.1 Herramientas para Módulos de una aplicación de Salud Móvil

En la Tabla 7.2 se muestra información o indicadores que se debe considerar antes de decidir en qué plataforma que se desarrollará una aplicación Móvil.

INDICADORES DE USUARIO PROGRAMADOR		
Descripción	Java ME	Android
Lenguaje de Programación	JAVA (CLDC/MIDP)	JAVA (Android SDK)
Disponibilidad del Entorno de Desarrollo Integrado (IDE)	Eclipse, NetBeans, kit Java™ Platform SDK 3.0.	Eclipse, NetBeans(con Android plug-in)
Disponibilidad del Emulador	Emulador Gratis, Sun Java Wireless Toolkit	Emulador Gratis
Costo de las Herramientas de Desarrollo	Gratis	Gratis
Requisitos de Memoria en Ejecución	Mayor a 0.5 Mbyte	Mínimo 32 Mbyte de RAM
Dispositivo(s) que soporta	Todos los dispositivos compatibles con CLDC (Connected Limited Device Configuration) y MIDP (Mobile Information Dance Profile),	Solo los dispositivos que tengan SO Android por la Máquina Virtual Dalvik. En su mayoría los dispositivos HTC (Magic, Hero, Tattoo), también T-Mobile (G1, pulso), Motorola Dext, Samsung Galaxy i7500, Hacer Liquid, Sony Ericksson Xperia X10.
Funcionalidad	Varía según el dispositivo, en función de JSR disponibles, sin imágenes de alta resolución, el acceso limitado de archivos	Pantalla táctil, acelerómetro, GPS, identificador de celda, la comunicación entre aplicaciones

Tabla 7.2 Indicadores para el Usuario Programador

Para la programación en estas plataformas se encuentran los IDE de manera gratuita, los cuales se pueden descargar de la red e instalarlos sin ningún problema en la PC. En este proyecto se utilizó NetBeans para la programación de la aplicación en Java ME y Eclipse para Android. En la instalación de los IDE tuve un pequeño problema con Eclipse para Android debido a que este IDE busca la dirección de los paquetes de Android para su compilación en C:/Android, y al momento de instalar estos paquetes se colocan automáticamente en D:/Android, por lo cual tuve que realizar un “re direccionamiento” con la opción de **variable de entorno** la cual me sirvió para que desde la consola de Windows se puedan compilar los archivos de Android.

7.1 Recomendación del uso de las plataformas Java ME y Android

Con esta evaluación se puede realizar la siguiente recomendación en la utilización de las plataformas Java ME y Android para los diferentes tipos de Aplicaciones de Salud Móvil que existen.

La plataforma Java ME es recomendable utilizarla en aplicaciones de tipo: **Educación y Conciencia, Apoyo a Diagnostico, Tratamiento, Comunicación y Formación de Trabajadores del Sector Salud**, debido a que este tipo de aplicaciones no requieren de las comunicaciones con servidores con conexión WiFi y con sensores médicos, además no se requiere de un diseño de interfaces extensas, ya que estas aplicaciones permiten enviar y recibir mensajes de información médica entre los trabajadores del sector salud y pacientes.

La plataforma Android se recomienda utilizarla para el desarrollo en aplicaciones de tipo: **Recolección de Datos Remotos, Monitorización Remota, Enfermedad y Seguimiento de Brote Epidémico, Diagnostico y Tratamiento de apoyo**, ya que estas requieren de interfaces mas intuitivas y sencillas, de comunicación con servidores Web donde se almacena información que es compartida entre los trabajadores del sector salud e incluso se requiere de adquisición de signos vitales de pacientes utilizando sensores médicos y esta información debe ser capturada en tiempo real y almacenada para su posterior análisis.

CAPITULO VIII

8. CONCLUSIONES

En este documento se presenta una investigación sobre las plataformas Java ME y Android para el desarrollo de aplicaciones Móviles específicamente para el sector de Salud Móvil. Donde se realiza una evaluación de estas plataformas, tomando en cuenta las características que debe contener una aplicación de salud móvil; logrando definir los módulos que debe contener una aplicación de salud móvil del tipo monitoreo de paciente; con este logro se diseñó una aplicación que cumpliera con dichos módulos, la cual fue desarrollada en ambas plataformas, esto permitió presentar las funcionalidades que contiene cada una de ellas para aplicaciones de salud móvil.

También se evaluó la aplicación en tiempo real, trabajando en 2 dispositivos móviles los cuales soportan las 2 diferentes plataformas, donde se tomaron en cuenta los tiempos de conexión y comunicación entre el (los) dispositivo (s) móvil(es) y el sensor medico; y entre el (los) dispositivo (s) móvil(es) y Servidor Web, con la finalidad de obtener resultados del comportamiento de cada plataforma, ya que los tiempos de comunicación son un factor muy importante en aplicaciones de salud móvil, donde se requiere que dicha comunicación sea lo más rápida y estable posible, para dar una atención de calidad a los pacientes.

La aplicación fue utilizada por 26 usuarios en los 2 diferentes dispositivos, una vez que interactuaron con ambos se les aplicó una encuesta para conocer sus opiniones sobre las opciones e interfaces que se diseñaron en cada plataforma. Estos resultados se encuentran en el capítulo 6 de este trabajo.

En conclusión general puedo decir que es posible desarrollar aplicaciones completas con ambas plataformas de programación móvil, sin embargo en la organización del desarrollo de las aplicaciones mi percepción personal fue que es más sencillo en la plataforma Android, además el funcionamiento del emulador para Android es mucho más real que para Java ME. También la manera en que se visualizan las interfaces de usuario me pareció más fácil en Android, debido a la separación de la vista gráfica y la lógica mediante los archivos XML que hacen el código más manejable.

Las API's de ambas plataformas son muy completas para utilizar los graficos, conexiones, transferencias de datos o la comunicación, pero encuentro una importante diferencia entre las API's para la realización de un almacenamiento persistente, ya que Android presenta una verdadera base de datos SQLite y Java ME cuenta con el RecordStore que permite guardar secuencias de bytes. El RecordStore es mucho mas pequeña que una base de datos real y mas lenta. Si se desean guardar una cantidad pequeña de datos no se encuentra una diferencia entre ellas, pero por el contrario si se desea guardar una cantidad considerable de información como es común en aplicaciones de salud móvil si se debe exigir el utilizar una base de datos real; pero hay que tener en cuenta que la API utilizada por Java ME tiene mas tiempo años siendo utilizada y por mas empresas, mientras que la de Android solo desde finales del 2007. Por otra parte me pareció que Android ofrece medios más interesantes con el navegador web o la pantalla táctil para todos los diferentes tipos de aplicaciones.

Con base a mi experiencia en el desarrollo de aplicaciones de salud móvil y a lo presentado en este trabajo yo prefiero y recomiendo la utilización de la plataforma móvil Android en comparación de la plataforma Java ME.

8.1 Trabajo Futuro

Como trabajo futuro se recomienda lo siguiente:

- Realizar evaluación de programación con usuarios que se dediquen a desarrollar aplicaciones para dispositivos móviles.
- Realizar la comunicación entre el dispositivo móvil con un sensor médico utilizando la tecnología 802.15.4 y ZigBee,
- Evaluar estos resultados con la plataforma iOS.

REFERENCIAS

Alaníz Sida, J. (2011). *Aplicacion Móvil para el Monitoreo de la Frecuencia Cardiaca utilizando el Sistema Operativo ANDROID*. Ensenada B.C.: Universidad Autónoma de Baja California.

Anta, R., Shireen, E.-W., & Guiffrida, A. (2009). Salud Móvil: El Potencial de la Telefonía Celular para llevar la Salud a la Mayoría. *BID-Banco Interamericano de Desarrollo*, 1-34.

Aparicio, S. (Noviembre de 2009). *Plataformas Linux para Móviles*. Recuperado el noviembre de 2010, de CEDITEC-ETSI Telecomunicación UPM: www.ceditec.etsit.upm.es/index.php/CIDETEC/Breves/Plataformas-Linux-para-moviles.html

Baz, Ferreira, Álvarez, & García. (2010). *Dispositivos Moviles*. (U. d. Oviedo, Ed.) Recuperado el Diciembre de 2011, de <http://156.35.151.9/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf>

BlueSentry. (s.f.). *Roving NetWorks Wireles for Less*. Recuperado el Octubre de 2010, de BLUESENTRY RN-800S-CB Bluetooth Sensor Interface "Install Guide and User Manual Version 1.0": <http://www.rovingnetworks.com/files/resources/Bluetooth-RN-800C-SB-UM.pdf>.

Bluetooth SIG, I. (2011). *Bluetooth*. Recuperado el Agosto de 2011, de <http://www.bluetooth.com>

Candolfi Arballo, N. (Diciembre de 2008). *Análisis del Desempeno de la Tecnologia inalámbrica Bluetooth*. Ensenada, Baja California, Mexico: Universidad Autonoma de Baja California.

Céspedes, H. (2011). e-Salud: Tecnologías y la nueva realidad de los Sistemas de Salud a nivel Global. *HC Global-Digital Economy*, (págs. 1-30). 21 de Febrero 2011.

Chan, Ray, & Parameswaran. (2008). Mobile e-health monitoring: an agent-based approach. *IET Communications*, 223-230.

Consulting, V. W. (2009). *mHealth for Development The Opportunity of Mobile Technology for Healthcare in the Developing World*. Washington, D.C. and Berkshire, UK: UN Foundation-Vodafone Foundation.

Consulting, V. W. (2009). *mHealth for Development: The Opportunity of Mobile Technology for Healthcare in the Developing World*. Washintong D.C.and Berkshire, UK: UN Foundation-Vodafone Foundation Partnership.

Developers, A. (2010). *What is Android?* Recuperado el Abril de 2010, de <http://developer.android.com/guide/basics/what-is-android.html>

Earl, O. (2008). A Survery of Platforms for Mobile Networks Research, Mobile Computing and Communications Review. *ACM SIGMOBILE*, 12(4), 56-63.

EDC. (2005). *Wireless Developers Lament Current State of Important Development Tools*. Recuperado el Noviembre de 2011, de Evans Data Corporation: <http://www.evansdata.com/press/viewRelease.php?pressID=96>

Fernández, R., Ordieres, J., Martínez, F., González, A., Lostado, R., & Alpha, V. (2009). *Redes Inalámbricas: Teoría y Aplicaciones practicas*. Universidad de la RIOJA servicios de Publicaciones.

Frenk, J. (2011). La Salud Móvil y los Sistemas de Salud: Determinantes del Progreso en la Salud Global. *Simposio Internacional sobre la Estrategia de la Salud Móvil para América Latina*, (págs. 1-11). 25 de Marzo del 2011.

Froufe Quintas, A., & Jorge Cárdenas, P. (2006). *J2ME Java 2 Micro Edition*. México, D.F.: AlfaOmega.

Galavas, D., & Economou, D. (2011). Development Platforms for Mobile Applications: Status and Trends. *IEEE Software*, 76-86.

Galvez, R. S., & Ortega, D. L. (2007). *Java a Tope: J2ME*. Málaga, Espana: Depto. de Lenguajes y Ciencias de la Computacion E.T.S. de Ingeniería Informática Universidad de Málaga.

Gavalas, D., & Economou, D. (2011). Development Platforms for Mobile Applications: Status and Trends. *IEEE Software*, 76-86.

Google, C. (2008). *Google Projects for Android*. Recuperado el Abril de 2010, de <http://code.google.com/intl/es-MX/android/>

Google, C. (2010). *Google Chart Tools*. Recuperado el Enero de 2011, de <http://code.google.com/intl/es-MX/apis/chart/>

Hu, F., Tillett, J., Ziobro, J., & Sharma, N. (2003). An Energy-Efficient approach to Securing treezone-based Sensor Networks. *Congreso Global de Telecomunicaciones de IEEE: GLOBECOM 2003* (págs. 1430-1434). 1-5 de Diciembre del 2003: IEEE .

Istepanian, R. S., & Lacal, J. C. (2003). Emerging Mobile Communication Technologies for Health: Some Imperative notes on m-health. *25th Congreso Anual Internacional* (págs. 1410-1416). IEEE-MBS.

Istepanian, R. S., Jovanov, E., & Zhang, Y. T. (2004). Introduction to the Special Section on M-Health: Beyond Seamless Mobity and Global Wireless Health-Care Connectivity. *IEEE Computer Society*, 4(8), 405-414.

Istepanian, R. S., Laxminarayan, S., & Pattichis, C. S. (2005). *Topic in Biomedical Engineering International Books Series: M-Health: Emerging Mobile Systems*. Piscataway New Jersey: Springer.

Istepianiann, R. S., Philip, N. Y., & Martini, M. G. (2009). Medical QoS Provision Based on Reinforcement Learning in Ultrasound Streaming over 3.5G Wireless System. *IEEE Journal on Selected Areas in Communications*, 4(27), 566-574.

Jovanov, E. (2006). Wireless Technology and System Integration in Body Area Network form m-Health Applications. *27th Congreso Anual Internacional de Engineering in Medicine and Biology Society*, (págs. 7158-7160). 17-18 de Enero 2006.

Jovanov, E., Price, J., Raskovic, D., Kavi, K., Martin, T., & Adhami, R. (2000). Wireless Personal Area Networks in Telemedical Environment. *Congreso Internacional de*

Tecnologías de la Información Biomedicina (págs. 74-78). 9-10 Noviembre del 2000: IEEE EMBS.

Kioskea.net. (Octubre de 2008). *Redes inalámbricas de área personal*. Recuperado el Noviembre de 2009, de <http://es.kioskea.net/contents/wireless/wpan.php3>

Knudsen, J. (2003). *Wireless Java: Developing with J2ME*. The Author's Press.

Liu, H., Ming-lu, & Lin, X. (2008). Mapping Web Services Standards to Federated Identity Management Requirements for m-Health. *Congreso Internacional Internet Computing in Science and Engineering*, (págs. 459-466). 28-29 Enero 2008.

LM35, D. (s.f.). *DataShet Catalog LM35*. Recuperado el Octubre de 2010, de www.datasheetcatalog.net/es/datasheets_pdf/L/M/3/5/LM35.shtml

Mobile Information Device Profile Reference. (s.f.). Recuperado el febrero de 2012, de <http://java.sun.com/javame/reference/apis/jsr118/>

Netmite, C. (2011). *Java Persistence Tools*. Recuperado el 2 de Julio de 2012, de <http://www.netmite.com/android/>

Nokia. (s.f.). *Nokia Connecting People*. Recuperado el Octubre de 2010, de www.nokia.com.mx

Olla, P., & Tan, J. (2008). Desingning a M-Health Framework for Conceptualizing Mobile Health Systems. En *Healthcare Information System & Informatic* (págs. 1-24). Hershey, Pennsylvania : IGI Global.

Oracle. (2010). *What is Java for Mobile Devices?* Recuperado el Abril de 2010, de <http://www.oracle.com/technetwork/java/javame/javamobile/overview/about/index>.

Pindter, J., González, V., & Tovar, C. (2009). Proposal for an m-Health System. *CERMA '09*, (págs. 55-59). 1 de Diciembre 2009.

Prajakta, K., & Yusuf, Ö. (2007). Requirements and Desing Spaces of Mobiles Medical Care. *ACM*, 11(3), 12-30.

Samsung. (s.f.). *Samsung México*. Recuperado el Agosto de 2011, de www.samsung.com/mx/

Tullis, T. S., & Stetson, J. N. (2004). *A Comparison of Questionnaires for Assessing Website Usability*. Boston: Human Interface Design Department, Fidelity Center for Applied Technology.

Vázquez, M., Méndez, M., Nieto, J. I., Jimenez, E., & Sanchez, J. d. (2010). Estudio de la Migración de Aplicaciones Desarrolladas en Java ME a la. *CiComp'10*, 1-7.

Voxiva, P. H. (2009). *Case studies: Rwanda Tracnet*. Recuperado el Mayo de 2010, de http://www.voxiva.com/content/case_studies/RwandaMinistryofHealth.pdf

Wikipedia. (2011). *M-health*. Recuperado el Enero de 2012, de <http://en.wikipedia.org/wiki/MHealth>

Yoorah, J. R. (2008). *Mobile Health 2.0*. Recuperado el Enero de 2012, de CEDITEC-ETSI Telecomunicación UPM:
http://www.ceditec.etsit.upm.es/index.php?option=com_content&view=article&id=21389&Itemid=1370&lang=es

Zephyr. (s.f.). *Zephyr HxM BT Brochure*. Recuperado el Agosto de 2011

ANEXO 1



Tesis: “Evaluación de Plataformas de desarrollo de aplicaciones móviles Java ME y Android para la implementación de servicios *de Salud Móvil*”

Evaluación: Interfaz de Usuario

Cuestionario sobre la experiencia de pacientes utilizando la aplicación “**Frecuencia Cardíaca Móvil**” desarrollada en las plataformas **Java ME** y **Android**

Perfil			
Edad: _____	Sexo: Femenino ☐	Masculino ☐	Fecha: __/__/__
Grado de Estudios: _____			
¿Padece alguna Enfermedad? _____ ¿Cuál? _____			

¿Qué dispositivo Móvil Utiliza?			
☐ BlackBerry	☐ Nokia	☐ Iphone/Ipad	☐ HTC
☐ Palm	☐ Otro: _____		

¿Qué sistema operativo tiene su Móvil?			
☐ Symbia OS	☐ Windows Mobile	☐ Apple	☐ Android
☐ BlackBerry	☐ Otro: _____	☐ No se	

Instrucciones: Lee cuidadosamente cada pregunta y marque con una X la opción que se adecue más a su opinión. Para cada Dispositivo Utilizado. (dispositivo1=Nokia N95, dispositivo2=Samsung-GALAXY)

Dispositivo 1				
1. Considero que necesito mucho tiempo para aprender antes de utilizar la aplicación con eficacia				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
2. Considero que la aplicación es excesivamente compleja				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
3. Considero que la aplicación es fácil de usar				

☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
4. Considero que sería necesario la ayuda de un experto para poder utilizar esta aplicación				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
5. Considero simple las opciones que presenta la aplicación				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
6. Considero que información proporcionada por la aplicación es fácil de entender				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
7. Considero que la organización de la información en las páginas de la aplicación (pantallas) es claro				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
8. Considero que la interfaz de esta aplicación es agradable				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
9. Me gusto usar la interfaz de esta aplicación				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo

Dispositivo 2				
1. Considero que necesito mucho tiempo para aprender antes de utilizar la aplicación con eficacia				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
2. Considero que la aplicación es excesivamente compleja				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
3. Considero que la aplicación es fácil de usar				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
4. Considero que sería necesario la ayuda de un experto para poder utilizar esta aplicación				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
5. Considero simple las opciones que presenta la aplicación				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
6. Considero que información proporcionada por la aplicación es fácil de entender				

☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
7. Considero que la organización de la información en las páginas de la aplicación (pantallas) es claro				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
8. Considero que la interfaz de esta aplicación es agradable				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo
9. Me gusto usar la interfaz de esta aplicación				
☐ Totalmente en Desacuerdo	☐ Ligeramente en Desacuerdo	☐ Neutral	☐ Ligeramente de Acuerdo	☐ Totalmente de Acuerdo

1. Si usted tuviera la necesidad de utilizar una aplicación para el cuidado de su salud, ¿en qué dispositivo le preferiría utilizarlo?

☐ Dispositivo 1

☐ Dispositivo 2