

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY**
CAMPUS MONTERREY
PROGRAMA DE GRADUADOS EN MECATRONICA Y TECNOLOGIAS
DE INFORMACION



**TECNOLÓGICO
DE MONTERREY®**

SERVICIOS MEDICOS A DISTANCIA A COMUNIDADES RURALES DE
LA REPUBLICA MEXICANA UTILIZANDO COMUNICACIÓN BASADA
EN EL PROTOCOLO CDMA 450.

TESIS

PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO ACADEMICO DE:
MAESTRO EN CIENCIAS EN TECNOLOGIA
INFORMATICA

POR
GAEL ESPIVENT DE LA VILLESBOINET DE CATUELAN

MONTERREY N.L.

DICIEMBRE 2009

INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

PROGRAMA DE GRADUADOS EN MECATRONICA Y TECNOLOGIAS DE INFORMACION

Los miembros del comité de tesis recomendamos que la presente tesis de Gaël Espivent de la Villesboinet de Catuélán sea aceptada como requisito parcial para obtener el grado académico de Maestro en Ciencias en Tecnología Informática.

Comité de tesis:

Dr. Guillermo Jiménez Pérez, PhD.
Asesor

Dr. José Ignacio Icaza Acereto, PhD.
Sinodal

Dr. Alfonso Avila Ortega, PhD.
Sinodal

Dr. Ramon F. Brena Pinero, PhD.
Director de maestrías en Computación de la División de Mecatrónica y Tecnologías
de Información

Diciembre 2009

**SERVICIOS MEDICOS A DISTANCIA A COMUNIDADES RURALES
DE LA REPUBLICA MEXICANA UTILIZANDO COMUNICACIÓN
BASADA EN EL PROTOCOLO CDMA 450.**

POR:

Gael Espivent de la Villesboinet de Catuelan

TESIS

Presentada al Programa de Graduados en Mecatrónica y Tecnología de
Información

Este trabajo es requisito parcial para obtener el grado de

Maestro en Ciencias en Tecnología Informática

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY

DICIEMBRE 2009

Dedicatorias

A mis padres

Quienes estuvieron siempre a mi lado para ayudarme a lograr lo que quería de mi vida. A pesar de la distancia que nos separa, fueron muy presente y muy importante.

A mes parents

Qui ont toujours été à mes cotés pour m'aider à réussir ce que je voulais faire de ma vie. Malgré la distance qui nous sépare, ils ont toujours été très présents et très important.

A toda mi familia

Porque siempre desean lo mejor para mí.
Espero ser para todos un ejemplo de voluntad y de éxito.

A toute ma famille

Parce qu'ils ont toujours souhaité le meilleur pour moi.
J'espère être pour eux un exemple de volonté et de réussite.

Agradecimientos

**A mi asesor principal,
Doctor Guillermo Jiménez Pérez**

Le agradezco por su disponibilidad, su apoyo y su confianza.

**A mis sinodales,
Doctor Alfonso Avila Ortega y Doctor José Ignacio Icaza Acereto**

Les agradezco por su interés en esta tesis y sus comentarios.

Al Doctor Jorge Carlos Mex Perera

Le agradezco mucho por el tiempo que ha dedicado a esta tesis.
Por su presencia y su aportación en el trabajo.

Resumen

El problema en que se enfoca la tesis es sobre el tema de asistencia médica a personas que residen en áreas rurales.

Se presenta una propuesta para resolver el problema de atención médica a personas que residen en áreas aisladas y que no cuentan con acceso a la red telefónica. La idea es también permitir la comunicación de datos necesarios para el diagnóstico de un médico.

En México, en las ciudades más grandes no hay ningún problema para encontrar un médico, se les puede llamar por teléfono o ir fácilmente a su consultorio. No obstante, para una persona que vive en el área rural, la comunicación es mucho más difícil. La red telefónica normal no permite cubrir todo el país. Para resolver el problema de la red de acceso, se piensa utilizar el protocolo CDMA 450 que Teléfonos de México (TELMEX) ha empezado a instalar en 2006 en México (la instalación se terminara en 2009).

Se muestra un prototipo de un sistema que envía diversos datos a un servidor. Estos datos son referentes a las condiciones de indicadores del estado de salud de una persona. Los datos son utilizados por un médico para emitir un pre-diagnóstico. Los datos son captados por un teléfono celular y a partir de ello se puede iniciar la interacción entre el médico y un asistente médico para proceder al diagnóstico remoto de la enfermedad de un paciente.

Tabla de Contenido

DEDICATORIAS	IV
AGRADECIMIENTOS	V
RESUMEN	VI
TABLA DE CONTENIDO	VII
ÍNDICE DE FIGURA	IX
ÍNDICE DE TABLA	XI
CAPÍTULO 1. INTRODUCCION	1
1.1. INTRODUCCIÓN	1
1.2. OBJETIVOS DEL PROYECTO DE TESIS	1
1.3. HIPÓTESIS	3
1.4. ORGANIZACIÓN DEL DOCUMENTO DE TESIS	3
CAPÍTULO 2. FUNDAMENTO TEÓRICO	4
2.1. ESTADO DEL ARTE	4
2.1.1. <i>Tele-Salud y Telemedicina</i>	4
2.1.2. <i>Requerimientos Clínicos de un sistema de Telemedicina</i>	7
2.1.4. <i>Obstáculos para la Telemedicina</i>	8
2.1.5. <i>Telemedicina en México</i>	9
2.1.6. <i>Médico a distancia para fragatas en solitario sin asistencia</i>	14
2.1.7. <i>El Doctor 2.0: HelloHealth</i>	14
2.1.8. <i>STETAU</i>	15
2.1.9. <i>HealthPresence</i> ,	16
2.1.10. <i>Proyecto HealthVault</i>	16
2.1.11. <i>Una plataforma de cuidado a domicilio</i>	16
2.2. DESARROLLO DE APLICACIÓN WEB	17
2.2.1. <i>Java Servlet</i>	17
2.2.2. <i>JavaServer Pages (JSP)</i>	17
2.2.3. <i>Java Database Connectivity (JDBC)</i>	19
2.2.4. <i>MySQL</i>	20
2.2.5. <i>Patrón de arquitectura de software Modelo Vista Controlador</i>	22
2.2.6. <i>Ajax con Java</i>	25
2.3. PROTOCOLOS Y DESARROLLO DE APLICACIÓN MÓVIL	27
2.3.1. <i>Java Micro Edition</i>	27
2.3.2. <i>CDMA 450</i>	30
2.3.3. <i>Gammu</i>	32
2.4. RESUMEN DEL CAPITULO	35
CAPITULO 3. ANÁLISIS DEL SISTEMA	36
3.2. ARQUITECTURA DEL SISTEMA	37
3.3. CASOS DE USO	38
3.4. DIAGRAMA DE CLASES	39
3.5. DICCIONARIO DE DATOS	40
3.6. RESUMEN DEL CAPITULO	41

CAPITULO 4. IMPLEMENTACIÓN DEL SISTEMA	42
4.1. SUB-SISTEMA DEL LADO DEL MEDICO.....	42
4.1.1. Contexto del Sub-Sistema.....	42
4.1.2. Arquitectura del Sub-Sistema.	43
4.1.3. Los Servlets y las paginas JSP del Sub-Sistema.....	43
4.1.4. Desarrollo con páginas JSP y Servlets.....	46
4.1.4. Diagramas de Secuencia	47
4.1.6. Funcionalidades implementadas en el Sub-Sistema.....	54
4.1.7. RSS con Ajax.....	68
4.2. SUB-SISTEMA DEL LADO DEL PACIENTE	72
4.2.1. Contexto del Sub-Sistema.....	72
4.2.2. Arquitectura del Sub-Sistema.	72
4.2.3. Diagrama de secuencia	73
4.2.4. Desarrollar con JME	74
4.2.5. Funcionalidades del Sub-Sistema.....	74
4.3. SUB-SISTEMA DE RECUPERACIÓN DE LOS DATOS	76
4.3.1. Contexto del Sub-Sistema.....	76
4.3.2. Arquitectura del Sub-Sistema.	77
4.3.3. Diagrama de secuencia	78
4.3.4. Funcionalidad del Sub-Sistema.	78
4.4. RESUMEN DEL CAPITULO	81
CAPITULO 5. CONCLUSIONES Y TRABAJO FUTURO	82
5.1. RESUMEN	82
5.2. EXPERIENCIAS APRENDIDAS	82
5.3. CONCLUSIONES	83
5.4. TRABAJO FUTURO	84
REFERENCIAS BIBLIOGRÁFICAS.....	85
VITA	87

Índice de Figura

Figura 1: La portabilidad de cada tipo de onda	2
Figura 2: Esquema de interacción cliente-servidor	3
Figura 3: Situación de la Tele-Salud en México.....	13
Figura 4: STETAU, un proyecto para un estetoscopio electrónico.....	15
Figura 5: Funcionamiento de los JSP	18
Figura 6: El API Java JDBC.....	20
Figura 7: Patrón MVC.....	23
Figura 8: Patrón MVC con Java.....	24
Figura 9: Ejemplo de interacción Ajax	26
Figura 10: El Java Micro Edition o JME	28
Figura 11: Comparación entre J2SE, CDC y CLDC.....	29
Figura 12: JME – Los diferentes sub-tipos Displayable	29
Figura 13: JME – Los diferentes sub-tipos Item.....	30
Figura 14: La red Telmex	31
Figura 15: Mapa de la Red Telmex.....	32
Figura 16: Ejemplo de archivo gammurc	34
Figura 17: Organización del sistema	37
Figura 18: Comunicaciones del sistema.....	38
Figura 19: Casos de Uso del sistema	39
Figura 20: Diagrama de clases.....	39
Figura 21: Arquitectura del sub-sistema del Médico	43
Figura 22: Arborescencia de Tomcat.....	46
Figura 23: Arborescencia de una aplicación web.....	46
Figura 24: Ejemplo de fichero web.xml.....	47
Figura 25: Diagrama de secuencia - Acceso a usuario.....	48
Figura 26: Diagrama de secuencia – Creación de paciente.....	48
Figura 27: Diagrama de secuencia – Modificación de paciente	49
Figura 28: Diagrama de Secuencia – Supresión de paciente	50
Figura 29: Diagrama de Secuencia – Creación de Medico	50
Figura 30: Diagrama de Secuencia – Modificación de Médico.....	51
Figura 31: Diagrama de Secuencia – Supresión de Médico	52
Figura 32: Diagrama de Secuencia – Diagnosticar un Paciente	53
Figura 33: Página de Inicio.....	54
Figura 34: Página de fallo al inicio.....	54
Figura 35: Si se olvide su contraseña, se puede enviar un correo de respalda.....	55
Figura 36: En espera de llamada.....	55
Figura 37: Elegir una llamada.....	56
Figura 38: Llamada elegida	56
Figura 39: Diagnóstico paciente	57
Figura 40: En espera de SMS, empezando el diagnóstico.....	57
Figura 41: SMS recibido.....	57
Figura 42: SMS leído y añadido al diagnóstico.....	58
Figura 43: Diagnóstico guardado.....	58
Figura 44: Búsqueda de un paciente	59
Figura 45: Búsqueda afinada de un paciente	59

Figura 46: IdPaciente desconocido.....	60
Figura 47: Administración – Crear Paciente	60
Figura 48: Administración – Pagina de fallo de creación de un paciente	60
Figura 49: Administración – Ingresando Datos del Paciente.....	61
Figura 50: Administración – Paciente Creado.....	61
Figura 51: Administración – Verificar datos paciente	62
Figura 52: Administración – Página de modificación de los datos del paciente.....	62
Figura 53: Administración – Suprimir Paciente	63
Figura 54: Administración - Búsqueda afinada del paciente a suprimir.....	63
Figura 55: Administración - Creación de un médico.....	64
Figura 56: Administración - Pagina de fallo de creación de un médico	64
Figura 57: Administración - Ingresando los datos del médico a crear	65
Figura 58: Administración - Médico creado.....	65
Figura 59: Administración – Acceso para modificar los datos del médico	66
Figura 60: Administración – Modificar los datos del médico.....	66
Figura 61: Administración – Supresión de un médico.....	67
Figura 62: Administración – Búsqueda afinada del médico a suprimir	67
Figura 63: Archivo XML de respuesta al Servlet rssNumero con fila vacía.....	68
Figura 64: El archivo XML transformado en contenido XML	68
Figura 65: Archivo XML de respuesta al Servlet rssNumero.....	69
Figura 66: El archivo XML transformado en contenido HTML.....	69
Figura 67: Archivo XML de respuesta al Servlet rssSMS si SMS a leer.....	70
Figura 68: El archivo XML transformado en contenido HTML.....	70
Figura 69: Archivo XML de respuesta al Servlet rssSMS.....	70
Figura 70: El archivo XML transformado en contenido HTML.....	70
Figura 71: Archivo XML de respuesta al Servlet rssPaciente.....	71
Figura 72: El archivo XML transformado en contenido HTML.....	71
Figura 73: Organización del parte del paciente.....	73
Figura 74: Diagrama de Secuencia Paciente.....	74
Figura 75: Pantalla de confirmación de registración	74
Figura 76: Pantalla de selección de los datos.....	75
Figura 77: Diferentes pantallas para ingresar datos	75
Figura 78: Pantalla en caso de comentarios adicional	76
Figura 79: Pantalla de confirmación del SMS	76
Figura 80: Organización de la recuperación de los datos.	77
Figura 81: Diagrama de secuencia de recuperación de los SMS.....	78
Figura 82: Fallo de conexión	79
Figura 83: Éxito de conexión	79
Figura 84: Ausencia de SMS	79
Figura 85: Presencia de dos SMS	80
Figura 86: SMS de tipo 1, Oficina agregado a la base de datos	80
Figura 87: SMS de tipo 2, Datos agregados a la base de datos	80
Figura 88: Supresión del SMS.....	81

Índice de Tabla

Tabla 1: Ejemplos de información clínica.....	7
Tabla 2: Opciones de telecomunicaciones	8
Tabla 3: Indicadores para México.....	10
Tabla 4: Licencia y Ventajas o Desventajas de DBMS existentes.....	22
Tabla 5: Módulos de funciones de Gammu	33
Tabla 6: Modelos de celulares compatibles con Gammu para diversos fabricantes	35
Tabla 7: Diccionario de datos	41
Tabla 8: URL y Utilidades de los Servlets del sub-sistema	45
Tabla 9: Utilidad de las paginas JSP del sub-sistema.....	45

Capítulo 1. INTRODUCCION

1.1. Introducción

La situación actual de las personas aisladas que necesitan comunicarse con un médico es difícil. Tienen que ir en automóvil hasta el consultorio, el cual puede estar a decenas de kilómetros. Otro caso es cuando las personas tengan problemas de salud que les impidan viajar, o bien que no tengan medio de transporte adecuado. Ante esta situación, las personas se pueden quedar sin ayuda.

El núcleo de la tesis es la propuesta de una arquitectura de software que ayude en este entorno, para brindar un medio de ofrecer atención médica a personas que viven en regiones aisladas de las zonas urbanas.

La propuesta se basa en el uso de telefonía móvil. La telefonía móvil de mayor difusión actualmente, usa antenas repetidoras que utilizan frecuencia de banda de 900 hasta 1800 MHz, las cuales tienen una cobertura limitada a 10km (Tisal, 2003). Esta tecnología es apropiada para las grandes ciudades con una densidad de población importante. Para un país como México, no es suficiente porque un cuarto de la población no tiene acceso a la telefonía móvil (mucho menos a telefonía local).

En esta tesis se propone un sistema computacional basado en el uso de un cliente móvil, el cual sería utilizado para la transmisión de datos (número de identificación, síntomas, etc.) y un servidor que puede crear páginas Web a partir de los datos recibidos del cliente móvil. Las páginas Web serían entregadas a un médico para que éste a su vez se ponga en contacto vía telefónica con el cliente móvil para ayudar en la emisión de un diagnóstico.

La parte de software para móvil se desarrolló en el entorno Java ME, que permite la producción de aplicaciones móviles. En la parte servidor, hemos utilizado la tecnología Java Server Pages (JSP) y Servlet para la producción de páginas Web dinámicas, además del Sistema de Administración de Base de Datos (DBMS por su siglas en inglés) MySQL para el almacenamiento de la base de datos. Para conectar las páginas web y el DBMS hemos utilizado el api Java DataBase Connectivity (JDBC).

Aunque pudiera existir un dispositivo que permita obtener los datos directamente de las lecturas de datos vitales de un paciente, esta tesis se limita con la introducción manual de los datos por el usuario.

1.2. Objetivos del proyecto de tesis

Según estadísticas, existen en México más de 199,398 pueblos con poblaciones menores a 5,000 habitantes. En 2005, según el INEGI, 23.5% de la población mexicana vive en la zona rural (24,266,896 habitantes).

Teléfonos de México (Telmex) está introduciendo una tecnología (el protocolo CDMA450) que permitirá utilizar el teléfono en esta zona. El protocolo CDMA 450 permite la utilización de ondas de frecuencias bajas de la banda UHF, la cual tiene un alcance de las señales de hasta 100 kilómetros (Figura 1).

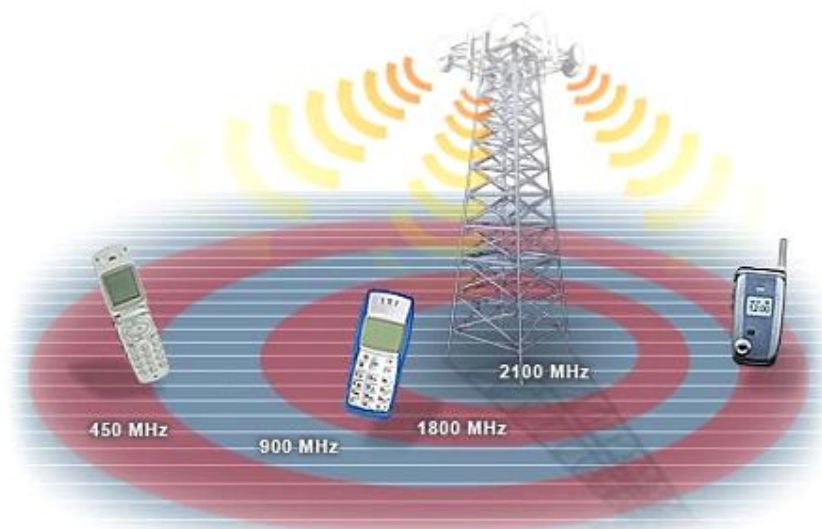


Figura 1: La portabilidad de cada tipo de onda

Podemos ver que la portabilidad aumenta cuando la frecuencia disminuye. Con una frecuencia de 450 MHz, el alcance es de 100 kilómetros mientras que con 900MHz el alcance ese reduce a 10 kilómetros.

La idea es comunicar mediante el teléfono a un medico ubicado en un centro urbano con un paciente en un área rural. El médico puede preguntar los datos necesarios para el diagnóstico, el usuario utiliza el teléfono móvil para enviar los datos. El servicio es personal basándose en un número de identificación personal (número de paciente) el cual será preguntado al paciente al inicio de la conversación.

Se necesita un sistema de cliente/servidor con una parte móvil donde la información sería introducida. Luego esta información sería enviada al servidor por medio de la tecnología CDM 450. Se tiene que entender los protocolos para enviar datos y el tamaño máximo posible. En la parte servidor se tendría una base de datos que permitirá hacer la gestión de los clientes y sus datos importantes; a partir de los datos se construye una página Web dinámica donde el médico encontrara los datos necesarias por su diagnostico (Figura 2).

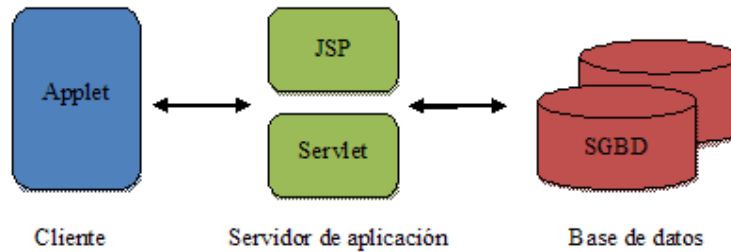


Figura 2: Esquema de interacción cliente-servidor

El Applet es la interfaz de un auxiliar médico, quien introduciría los datos requeridos para un diagnóstico. Se accede la información del paciente en la base de datos y a partir de esto se crean páginas web dinámicas mediante los Servlets o JSP.

1.3. Hipótesis

Es posible proporcionar servicios médicos a distancia a comunidades rurales de la Republica Mexicana utilizando comunicación basada en el protocolo CDMA 450.

1.4. Organización del documento de tesis

La estructura y presentación de este documento de tesis es la siguiente.

El capítulo 1 resume el problema de investigación e integra los temas fundamentales del proyecto de investigación.

El capítulo 2 describe la investigación realizada. La investigación aborda tres temas principales; el estado del arte, el desarrollo de aplicaciones Web y el desarrollo de aplicaciones móviles.

El capítulo 3 describe el sistema de manera global. Tratando de entender el contexto del sistema, se explica la arquitectura diseñada y la representación del modelo de los datos.

El capítulo 4 se analiza la implementación del sistema tomando en cuenta cada parte de sistema: el sub-sistema del médico, del paciente y el sub-sistema de recuperación de los datos.

El capítulo 5 presenta las conclusiones del trabajo de tesis, resume las experiencias aprendidas y destaca aspectos que pudieran ser añadidos al sistema para cubrir una gama más amplia de servicios accesibles tanto a los médicos como a los pacientes.

Capítulo 2. Fundamento Teórico

En este capítulo, se abordan tres temas; el estado del arte, el desarrollo de aplicaciones Web y el desarrollo de aplicaciones para móviles.

La investigación sobre el estado del arte explica los conceptos de la Tele-Salud y de la Tele-Medicina, explica sus obstáculos y detalla varios proyectos relacionados a este tema.

El desarrollo de aplicaciones Web se enfoca a las tecnologías de información y las nuevas tendencias en el desarrollo de aplicaciones para ofrecer soluciones innovadoras de software en general.

La aplicación móvil se enfoca en las tecnologías para construir aplicaciones móviles, el protocolo de comunicación y una librería (API) que permitiera utilizar un móvil directamente conectado al servidor.

2.1. Estado del Arte

2.1.1. Tele-Salud y Telemedicina

La Tele-Salud (en inglés TeleHealth) se define (Gantenbein, Cadez, & Stamm, 2004) como las tecnologías de información electrónicas y de telecomunicaciones para soportar, a larga distancia, atención clínica en salud, educación de pacientes y profesionales de la salud, salud pública y administración de la salud.

La intención de la Tele-Salud es proporcionar servicios de salud donde la distancia es un factor crítico, los profesionales de la Salud utilizan tecnologías de información y de comunicaciones (ICTs) para intercambiar datos de información importantes para proporcionar diagnósticos, tratamientos, prevención de enfermedades y heridas, se utiliza también para continuar la educación de los proveedores de atención en salud (International Telecommunication Union & Pan American Health Organization, 2003).

La Tele-Salud es un recurso tecnológico que posibilita la optimización de los servicios de atención en salud, ahorrando tiempo, dinero y facilitando el acceso a zonas distantes para tener atención de especialistas. Otra de las utilidades que presta el uso de la transmisión de datos médicos sobre redes adecuadas es la educación, donde los alumnos de medicina y enfermería pueden aprender semiología remotamente, apoyados por su profesor y con la presencia del paciente.

Se destacan cuatro áreas de aplicaciones:

- Telemedicina
- Tele-Salud
- Investigación de redes y Tele-Epidemiología.
- Redes de Administración de Salud.

E-medicina y E-Salud son nuevos términos de la E-cultura. Etimológicamente significan Medicina-Electrónica y Salud-Electrónica. Esos términos se utilizan para actividades de la Medicina y de la Salud donde no se requiere el ser humano. Un sinónimo sería medicina en línea y salud en línea. Es importante diferenciar la E-Medicina de la Telemedicina y la E-Salud de la Tele-Salud.

La Telemedicina es parte de la Tele-Salud pero concierne únicamente actividades por profesionistas registrados (certificados).

La Telemedicina es una de las formas de cooperación en el ejercicio médico, utilizando las tecnologías de información y de comunicación para poner en contacto a distancia un paciente (y/o los datos médicos necesarios) y uno o varios médicos y profesionales de la salud para fines medicinales como un diagnóstico, una decisión, un tratamiento o una manera de encargarse del paciente con respeto de las reglas de la deontología médica (Legmann & Lucas, 2009).

El acto de Telemedicina es un acto médico que se efectúa controlado y con la responsabilidad de un médico mediante los medios de comunicaciones apropiados para comunicarse con el paciente (Legmann & Lucas, 2009). Se necesita completar la definición de la Telemedicina con sus posibles actos: Tele-consulta, Tele-valoración, Tele-vigilancia, Tele-asistencia médica.

- La Tele-consulta:

La Tele-consulta se efectúa en relación con el paciente en dos tipos de situaciones. El caso más común es la regulación médica (15 millones de llamadas por año): el paciente contacta por teléfono un centro donde el médico puede establecer el diagnóstico de gravedad y tomar la decisión de orientación del paciente. Esa práctica ya tiene protocolos de buena práctica y puede apoyarse sobre sistemas expertos.

Existe otro tipo de Tele-consulta. El médico es consultado a distancia por el paciente al lado del cual se encuentra otro médico u otro profesional de la salud. Ese modo de Tele-consulta es útil para atender a sitios aislados (zonas rurales) o móviles (navío mercante o de guerra).

- La Tele-valoración

La Tele-valoración concierne un intercambio profesional entre dos o varios médicos, o sea para la concertación entre médicos, o sea para la repuesta de un médico distanciado solicitado por un médico encargado de un paciente. No es muy diferente de la consulta especializada o de un segundo informe de valoración médica. Se distingue en el hecho de que no es el paciente quien se desplaza sino los datos clínicos, biológicos o de imágenes. La Tele-valoración se ha desarrollado para el diagnóstico prenatal y en cancerología para facilitar las reuniones de concertación pluridisciplinarias. Se puede extender a todas las disciplinas.

- La Tele-vigilancia

La Tele-vigilancia se distingue de la Tele-consulta en el sentido de que concierne un paciente conocido del médico. Resulta de la transmisión de uno o varios indicadores fisiológicos recuperados, ya sea por el paciente, por otro profesional de salud o por un auxiliar de salud. El médico interpreta los datos a distancia y modifica la manera de encargarse del paciente cuando se necesita. Esa Tele-vigilancia médica se distingue claramente de la Tela-asistencia médica. Pueden completarse pero no deben ser confundidos. La Tele-vigilancia se utiliza por ejemplo en caso de insuficiencia cardiaca, respiratoria o renal pero también en caso de diabetes.

- La Tele-Asistencia médica

La Tele-Asistencia médica es el acto durante el cual un médico asiste técnicamente a otro médico a distancia. Su aplicación más mediatizada, en el tema de la tele-asistencia médica, es la tele-cirugía. Esa aplicación ya ha sido desplegada en varios países para acelerar la obtención de una segunda opinión que permita la valoración operatoria en un territorio alejado.

- La Tele-Educación

Existen muchas aplicaciones de aprendizaje a distancia, en tiempo real o no. La Tele-Educación permite llevar a cabo varias actividades de manera remota, entre otros:

- Actividad de prevención
 - o Prevención Primaria: Campaña de prevención (como anti-tabaco)
 - o Prevención Secundaria: Detección anticipada de enfermedad o condición pre-patológica.
 - o Prevención Terciaria: Seguir y actuar para corregir el habito patógeno crónico o post-critico de un paciente para reducir las complicaciones y recaídas
- Entrenamiento de Currículo
 - o Entrenamiento a distancia
 - o Desarrollo profesional
 - o Evaluación y posibilidad de retroalimentación entre los alumnos y los profesores
- Entrenamiento medico y desarrollo profesional

- Tele-Administración

La Tele-Administración se aplica a la Salud para ayudar el manejo de citas, facturación, mantener un inventario, planear estrategia. Es muy útil para manejar emergencia, crisis o catástrofes.

2.1.2. Requerimientos Clínicos de un sistema de Telemedicina.

Un sistema de Telemedicina necesita aparatos apropiados pero no únicamente eso. Los tres componentes esenciales son: el personal, la tecnología y una medida liberal de perseverancia.

El personal necesita tener una formación adecuada para asegurarse de dos cosas: poder confiar en su personal y tener un personal suficiente.

La perseverancia es importante, en el hecho de que la experiencia ha mostrado la importancia de poder contar con al menos una persona, dedicada y comprometida, con la perseverancia de vencer la inercia inherente que se establece en la rutina clínica.

La tecnología para la telemedicina se puede dividir en tres partes (Harnett, 2006):

- Aparatos para recuperar la información clínica en cada sitio.
- Aparatos para transmitir la información clínica entre sitios.
- Aparatos para mostrar la información clínica en cada sitio.

Pero antes de elegir esos aparatos es importante conocer la información que se va a transmitir entre los sitios porque eso podría determinar la decisión. Se tiene que considerar varios factores (Harnett, 2006):

- El tipo de información a transmitir.
- La cantidad de información.
- La seguridad y la privacidad que se necesita proveer.

En telemedicina, el tipo de información clínica puede dividirse en cinco categorías:

- Documentos
- Grabación médica electrónica
- Imágenes
- Audio
- Video

La Tabla 1 da varios ejemplos de información clínica con su tamaño promedio. Esa información nos permite elegir los medios de comunicaciones.

Fuente de informacion	Tipo	Tamaño Promedio
Electronic stethoscope	Audio	100kBytes
ECG recording	Datos	100kBytes
Chest X-Ray	Imagen	1 Mbytes
Fetal ultrasound recording (30seg)	Video	10 MBytes

Tabla 1: Ejemplos de información clínica

En la Tabla 2, se encuentra opciones de telecomunicaciones para transmitir datos. Podemos analizar la tabla tomando en cuenta la tasa de transferencia de cada opción, su cobertura y su costo. Además, elegir una opción posiblemente nos obliga al uso de un tipo de dato.

Sistema	Taza de transferencia	Cobertura	Costo	Uso
PSTN	33.6 - 56kbit/s	Omnipresente	Bajo	Datos(texto)
Mobile Phone	19.2 - 100kbit/s	Común en países industrializados	Bajo	Datos(texto)
Cable Modems	>1Mbit/s	Común en países industrializados	Bajo	Datos, imagen, video limitada
xDSL	>1Mbit/s	Común en países industrializados	Bajo	Datos, imagen, video limitada
ISDN(basic rate)	128kbit/s	Común en países industrializados	Medio	Datos (texto, imagen), video
ISDN(primary rate)	1.54 - 2Mbit/s	Común en países industrializados	Alto	Cualquier
Leased lines	64kbit/s- 2Mbit/s	Accesible en muchos países	Bajo / Medio	Cualquier
Microwaves	<30Mbit/s	Accesible en muchos países	Muy Bajo	Cualquier
Satellites	64kbit/s- 2Mbit/s	Cobertura Global	Alto	Cualquier
ATM	>15Mbit/s	Uso Restringido	Medio/Alto	Cualquier

Tabla 2: Opciones de telecomunicaciones

2.1.4. Obstáculos para la Telemedicina

Existen varios obstáculos para la Telemedicina; obstáculos técnicos, de derecho, de ética, etc. En los obstáculos técnicos podemos hablar de la seguridad. Saber si podemos confiar en la arquitectura sobre la cual se construye el sistema. Otro problema es la privacidad de los datos y la confidencialidad. La medicina es un área donde el tema de privacidad de los datos y de su confidencialidad es muy importante. En la deontología médica, existe el secreto médico que prohíbe la divulgación de la información de salud de una persona. Para resolver eso se utiliza varios medios como por ejemplo la encriptación de los datos (Legmann & Lucas, 2009). Otro problema es la calidad de la transmisión. El ejemplo de las imágenes es interesante. Hemos visto que el tamaño promedio de las imágenes es de 1MB. Dependiente de la acción que se requiere hacer después del transporte de manera electrónica de la imagen, es posiblemente que su transformación y su encriptación van a disminuir su calidad y que no se podría utilizar la imagen. Existe este problema en cuestión de tele-radiografía por ejemplo.

Se dice también que la cuestión legal es muy importante y que es un obstáculo importante para la Telemedicina por eso se explica que se necesita un cambio de las leyes tomando en cuenta la especificidad en la Telemedicina y su utilidad (Fitzmaurice, 1998). Otro problema es la cuestión de la responsabilidad de los diferentes actores del sistema. Es importante poder definir el nivel de responsabilidad del paciente, del médico pero también de otros actores involucrados en el sistema como el proveedor de la red o el constructor de los aparatos. Por fin, la cuestión de la remuneración es importante. Como todo se hace a

distancia: como se hace los pagos, como se puede confiar en el otro asegurándose que el médico por ejemplo va ser pagado pero también asegurándose que el paciente va a pagar un precio equivalente a su problema. Además se tiene que pensar en todos los actores involucrados como el paramédico que ayuda el paciente u otra persona.

2.1.5. Telemedicina en México.

México es uno de los primeros países hispanohablantes de la región en empezar actividades en Telemedicina (International Telecommunication Union & Pan American Health Organization, 2003). El gobierno se ha involucrado directamente en el desarrollo para promover la Telemedicina y la Tele-salud, y ha desarrollado proyectos a través de dos programas del sector público.

En 1995, empezó la fase piloto del programa nacional de Tele-Salud con el Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado (ISSSTE). Utilizando una parte de las capacidades de las comunicaciones del satélite de telecomunicaciones mexicano. Esa porción se utiliza para la educación pública y la salud. Varios estados se involucraron: Baja California Sur, Baja California Norte, Chiapas, Chihuahua, Guerrero, Durango, Guanajuato, Jalisco, Michoacán, Nuevo León, Oaxaca, Sonora, Sinaloa, Tabasco, Tampico, Veracruz, Yucatán y el Distrito Federal. El gobierno local tiene un papel importante para el éxito del programa de Tele-Salud Nacional.

Al inicio del mandato del Presidente Fox, una consulta democrática de los ciudadanos fue realizada y revelaba la convicción y el interés de los ciudadanos a propósito del potencial de las tecnologías de telecomunicaciones y de comunicaciones para mejorar los servicios de salud proveído al público en particular para grupos marginados y vulnerables.

El resultado ha provocado la creación del programa e-México. La parte de la cuestión de la Salud en este programa es de la competencia del programa de acción e-Salud. Empieza otra fase piloto que permite incorporar las nuevas tecnologías de comunicación, adicional a aplicaciones de tecnología de información, en particular aplicaciones basadas sobre el IP, y nuevos servicios para la comunidad y el sector profesional, sobre diversas áreas geográficas. El presidente de la Republica asignó la coordinación del proyecto a la secretaría de comunicaciones y de transportes (SCT), ejecutado por la Secretaria de Salud (SSA) en una demostración de la capacidad operacional de la interrelación entre diferentes áreas de salud pública. Además participan, en adición de la Secretaria de Salud, el Instituto Mexicano de Seguridad Social, el Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado, Petróleos Mexicanos, la Secretaria de Defensa, instituciones y organizaciones del sector privado, en coordinación con el sistema nacional e-México.

El propósito del programa de acción e-Salud es de contribuir en mejorar la salud pública y aumentar la cobertura del servicio, asignando prioridad a la población en áreas muy marginadas a través de un sistema telemático de contenido social, para poner accesible al público vía Internet, información de salud, contribuyendo al desarrollo humano e individual, y desarrollo de la sociedad: con el fin de proveer soporte para la educación y el desarrollo profesional del personal de la salud.

En paralelo, el portal e-Salud permite informar a la población de actividad de promoción de la salud, de prevención de accidentes y trámites gubernamental en el área de la Salud.

Con respecto al tema de la legislación y de la regulación del uso de los ICT en el área de la salud, México ha definido un marco legal. Disposiciones han sido establecidas en métodos, leyes y obligaciones en la práctica profesional, las responsabilidades de los profesionales de la Telemedicina y protecciones de los datos, especialmente cuando se transmitan por una red ICT o vía aplicaciones de ICT.

México es el único país de la región que ha proporcionado, en 2003, regulaciones en cooperación médica transnacional a través del uso de las ICTs. Esas regulaciones se aplican únicamente para actividades, soportadas por equipos de profesionales extranjeros, como segunda opinión médica y continuar educación médica.

La Tabla 3: Indicadores para México muestra indicadores sobre México recibidos de un reporte de la Organización Mundial de la Salud (WHO) (WHO Global Observatory for eHealth, 2006).

Indicadores para México					
Población (x1000)	103 718	País del OECD	Si	Líneas de teléfonos local*	17.22
GDP per cápita (Int \$)	9 445	Categoría del banco mundial	2	Usuarios de Internet*	13.38
Total gastos por la Salud (% of GDP)	6.2	Índice de difusión de los ICTs	0.2969	Suscriptores de teléfonos celulares*	36.64

* Por 100 habitantes

Tabla 3: Indicadores para México

Según el mismo reporte, muchas cosas, si se implementan, podrían ser muy útiles para México:

- Sistema de información geográfica(GIS)
- Tele-Salud
- Sistema de soporte a la decisión (DSS)
- Directorios de profesionales de asistencia en salud e instituciones
- Registro nacional de droga
- Registro electrónico Nacional
- Sistema de información para los médicos especializados en medicina general(GPIS)
- Sistema de información para Hospitales (HIS)
- Sistema de información para pacientes(PIS)
- Guardar la información de salud en forma Electrónica (eHR)
- Ayuda en el desarrollo de recursos humanos para eSalud
- Ayuda en programas de eAprendizaje
- Información en las tendencias y desarrollos en eSalud
- Ayuda en las normas y estándares en eSalud
- Información sobre las mejoras prácticas en eSalud
- Ayuda en métodos para monitorear y evaluar servicios de eSalud
- Ayuda en políticas y en estrategias de eSalud
- Ayuda en evaluación de la necesidad nacional en eSalud

Gracias al portal e-Salud, se puede encontrar información sobre el estado de la Telemedicina en varios Estados de la República.

- Chiapas

El estado de Chiapas forma parte de la Red Nacional de Telemedicina desde marzo del 2006.

Chiapas cuenta con una población estimada de 4,357,301 habitantes, de los cuales 3,590,608 no se encuentran asegurados. El 54.3% de la población vive en zonas rurales, por lo que Chiapas ocupa el octavo sitio dentro de los estados de la República que cuenta con un gran número de habitantes rurales.

Existen 119 municipios, 10 de ellos pertenecen al grupo de los 50 Municipios con menor Índice de Desarrollo, la mayor parte de ellos ubicados en la Región de los Altos.

Uno de los principales problemas en salud es el alto índice de mortalidad materna estatal. Actualmente se encuentran equipados los dos Hospitales Regionales de Alta Especialidad (Ciudad Salud en Tapachula y el Hospital de Especialidades Pediátricas de Tuxtla Gutiérrez) y el Hospital General de Palenque para brindar consulta médica de especialidad a distancia y sesiones de tele-educación.

El estado de Chiapas está incluido en el Programa de Caravanas de la Salud, de la Secretaría de Salud. Se contempla 1 Unidad Tipo 3 para el Estado.

- Estado de México

De acuerdo al Censo de Población y Vivienda del 2005, residían en el estado de México un total de 14,007,495 personas, de las cuales 51.2% son mujeres y 48.8% son hombres, representando 13.6% de la población total nacional. Existen diferentes etnias: mazahua, otomí, nahuas, matlatzincas, ocuitecos, mixteca, zapoteca, totonaca, mazateca, mixe, purépecha y maya.

En cuanto a Telemedicina se refiere, existe un equipo de videoconferencia en los Servicios de Salud Estatales, el cual se utiliza principalmente para recibir sesiones de Tele-educación.

- Guerrero

Guerrero cuenta con una población total de 3,260,576 habitantes hasta el 2005, y una población no asegurada de 2,466,568 habitantes, según la fuente de información de CONAPO

Guerrero cuenta con 81 Municipios de los cuáles 11 de ellos se encuentran en el Menor índice de Desarrollo Humano a nivel Nacional. (Copalillo, Acatepec, Alcozauca, Atlamajalcingo del Monte, Atlixac, Copanatoyac, Metlatonoc, Cochoapa, Zapotitlan Tablas, Tlacoachistlahuaca y Xochistlahuaca).

El estado cuenta en sus cabeceras Municipales y algunas de sus comunidades más importantes con los servicios mínimos indispensables para establecer comunicación efectiva a través del correo y radio a nivel nacional. El estado cuenta con medios de telecomunicación mediante los servicios de telégrafo, banda civil, prensa, telefonía celular,

comunicación a través de redes internacionales como: internet y recepción de señal de televisión sobresaliendo los municipios de Chilpancingo, Acapulco de Juárez y Coyuca de Benítez

El estado de Guerrero está integrado a la Red Nacional de Telemedicina desde 2003. Debido a su complicada orografía, la comunicación por caminos y carreteras en el estado ha resultado sumamente complicada y por lo tanto costosa; esto provoca que 23 municipios no tengan acceso por vías pavimentadas, situación que provoca el rezago que prevalece en el desarrollo del estado.

El estado cuenta con municipios altamente marginados como es la zona de la montaña y la zona sureste de la parte del estado.

La zona más marginada del estado es la región de la Montaña, donde se encuentran ocho de los municipios con menor índice de desarrollo humano. Dentro de estos ocho municipios, se encuentra Metlatónoc, el municipio más pobre del país, con extremos rezagos en salud, prácticamente carece de infraestructura en hospitales, carreteras y drenaje, así como elevados índices de migración y analfabetismo.

Dentro del estado, se encuentran dentro de la Red de Telemedicina tres Hospitales Generales, dos Hospitales Materno – Infantil, siete Hospitales Básicos Comunitarios y 2 centros de Salud.

La cobertura del resto de los municipios requiere de la implementación de un sistema de videoconferencia, equipo médico y medios de comunicación más eficientes, de mayor alcance y de menor costo.

- Guanajuato

En el estado de Guanajuato el acceso de la población a los servicios médicos es por medio de organismos públicos y privados. La población derechohabiente del Estado es de 2 millones 492 mil personas en instituciones públicas de salud. De éstas, el 84.23% pertenece al IMSS, el 13.6% al ISSSTE, 1.2% a PEMEX y 0.8 a SDN. Por otro lado, el Estado de Guanajuato cuenta con 613 unidades médicas de salud del sector público.

Basados en la creciente demanda de la población por acceder a servicios de atención médica de especialidad en la región, se concretó la necesidad de desarrollar un Hospital Regional de Alta Especialidad, que forma parte de una Red de Servicios de Alta Especialidad. Desde un principio se pensó en la posibilidad de incluir los servicios de Telemedicina, por lo cual en el equipamiento se incluyeron los aditamentos básicos para poder brindar este tipo de Servicios. Hasta el momento, el Hospital del Bajío, ubicado en la capital del Estado, se ha incorporado a los servicios de tele-educación.

- Nuevo León

En febrero del 2001 se inicia la Telemedicina en el estado ya que para los Servicios de Salud de Nuevo León cada día era más difícil contar con médicos especialistas en las zonas rurales.

Actualmente no sólo se brinda atención a la población en zonas remotas, sino también se proporciona atención médica, enseñanza y capacitación a diferentes grupos de la población, como a universidades, hospitales privados y centros penitenciarios, por mencionar algunos.

Hasta noviembre del 2007, hay 30 sitios activos con enlaces dedicados y los procesos que se llevan a cabo en el programa de Telemedicina ahora llamado "Telesalud Nuevo León".

La Figura 3 muestra la situación de la Tele-Salud en la República Mexicana. Muestra un aislamiento médico de muchos estados del país. Además de eso, la Tele-Salud, implementada en México, trata de responder al problema de la ausencia de especialistas en los Estados pero no se toma en cuenta la utilización de la Tele-Salud para brindar un servicio médico a municipios o pueblos que no cuentan con centro de salud.



Figura 3: Situación de la Tele-Salud en México.

Las siguientes secciones son ejemplos de proyectos de Tele-Salud que ya existen y que permiten tener una idea más clara del tipo de cosa que se puede obtener.

2.1.6. Médico a distancia para fragatas en solitario sin asistencia.

La Asistencia Médica a Distancia fue conocida con un drama marítimo. Bertrand de Broc está solo en su velero. Estaba navegando en los famosos Rugientes Cuarentas (en inglés Roaring Forties), una zona de vientos fuertes que se ubica entre las latitudes 40° y 50° de los océanos australes, cuando un cable le azota la cara. Con el golpe, Bertrand se corta la lengua. Desamparado, hace una llamada para solicitar asistencia. A miles de kilómetros, el doctor Jean-Yves Chauve transmite al herido sus instrucciones para que pueda suturar y curarse él mismo su herida.

Desde entonces el doctor Jean-Yves Chauve ha desarrollado y experimentado sus investigaciones para proveer una consulta a distancia. Ha escrito un libro que aborda las situaciones que tiene una persona en desplazamiento (Chauve, 2002). Sus consejos son prácticos y alcanzables por todo el mundo.

2.1.7. El Doctor 2.0: HelloHealth

HelloHealth es una plataforma innovadora para la atención primaria que ayuda a médicos a comunicarse, documentar y operar a pacientes en persona y en línea (Johnson, 2009). HelloHealth permite disminuir dos gastos indirectos de los médicos, porque elimina la necesidad de recepcionista, administración y equipo de paramédicos. Un buen ejemplo de quienes utilizan este sistema es el Doctor Jay Parkinson. Es el cofundador de HelloHealth y un médico especialista en pediatra y en medicina preventiva. Comenzó a trabajar en Nueva York el 24 de septiembre de 2007. Explica su organización en 6 pasos (Williams, 2009):

- Sus pacientes visitan su sitio Web
- Checan su calendario Google
- Encuentran una fecha e ingresan sus síntomas.
- Su Iphone le alerta.
- Hace una llamada
- Puede ser pagado vía Paypal.

HelloHealth explica que permite mejorar el servicio al cliente disminuyendo el tiempo de espera. Anteriormente, para recibir un diagnóstico se suponía que se tenía influenza, una persona debía esperar 4 horas, hoy puede recibir un diagnóstico en 1 hora. Para un diagnóstico de asma, antes se deberían esperar 5 horas, ahora se puede tener el diagnóstico en 30 minutos; y si se corta el dedo debería pasar 8 horas de espera en emergencias, ahora basta con 2 horas.

2.1.8. STETAU.

STETAU (en francés *Système et Technologies d'Enregistrement et de Traitement des Sons Auscultatoires*) es un proyecto Europeo que tiene como objetivo desarrollar un estetoscopio electrónico comunicado a un PDA para guardar, tratar y explorar los sonidos de las datos pulmonares y cardiacos.

Explican (Gass, Reichert, Gass, & Andrès, 2007) que la auscultación, desde su creación en 1816 es un elemento importante de la formación y de la vida profesional de un médico. La auscultación es un proceso:

- Subjetivo porque depende de la memoria auditiva del médico.
- Efímera porque es imposible guardar y documentar.
- Personal porque es escuchado por una sola persona.
- Compleja porque es difícil diferenciar el ruido de la información útil
- Útil, casi indispensable.

STETAU propone modernizar la eficiencia médica en la auscultación; es desarrollado en colaboración con la unidad de cardiología del CHRU de Strasbourg en Francia, es un proyecto patrocinado por la empresa Alcatel-Lucent.

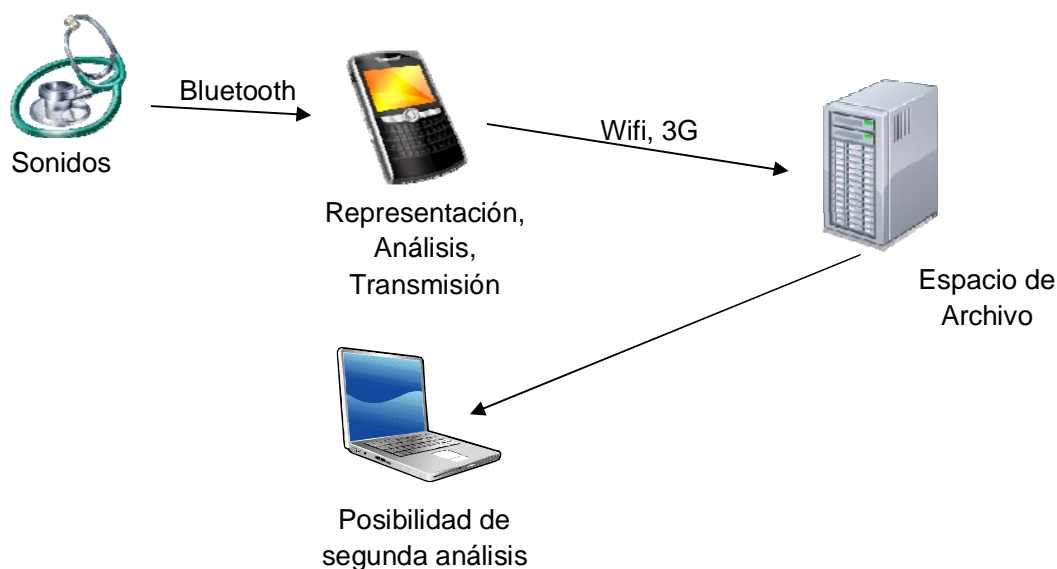


Figura 4: STETAU, un proyecto para un estetoscopio electrónico.

El proyecto STETAU de la plataforma MERCURE permitiría suprimir las características nefastas de la auscultación. La Figura 4 muestra su funcionamiento. El estetoscopio electrónico obtiene los sonidos gracias a un micrófono ultra poderoso sin filtraje y les transmite vía Bluetooth hacia un aparato capaz de guardar esa señal: PDA o celular, computadora, etc. Después la señal será retransmitida vía una red (Internet o GSM/3G) hacia un centro de archivo y monitoreo de los pacientes. El médico puede acceder y analizar la grabación sonora y visual.

Las principales ventajas de ese aparato son:

- Una mejora del proceso de auscultación
- El monitoreo del corazón y de los pulmones
- La grabación de los sonidos podrían permitir escuchar de nuevo y analizar la señal.
- La detección de anomalías.
- Una presentación del sonido de manera grafica.

Ese proyecto podría beneficiar a millones de personas en el mundo (BPCO, asma, insuficiencia cardiaca, etc.)

2.1.9. HealthPresence,

HealthPresence es un proyecto escocés en que se realiza mediante colaboración del centro escocés de Telemedicina, Cisco y NHS Boards. El propósito de este proyecto es demostrar la viabilidad de un servicio innovador para ligar al paciente y al médico a distancia para evaluar datos clínicos (Augustinos & Shehata, 2009).

El sistema de Cisco fue pensado al inicio como una aplicación para videoconferencias empresariales. Fue adaptado para su uso en un centro de salud. Este dispositivo de comunicación por video “al mismo tamaño” se utiliza con diferentes equipos de diagnósticos – estetoscopio, oxímetros de pulso, etc. La alta resolución de la pantalla permite que el dispositivo pueda ser útil para realizar consultas médicas a distancia en un modo realista.

Esa iniciativa permitirá disminuir el déficit de servicios médicos especializado en este país, principalmente en zona rural.

2.1.10. Proyecto HealthVault

Microsoft ha empezado un proyecto llamada HealthVault. Es una plataforma de administración de información de salud personal. Podría constituir un soporte común de información de la salud para el usuario y los proveedores (Zheng, 2007).

HealthVault es un concepto innovador, permite al mismo tiempo recuperar, guardar y compartir la información de salud de una persona. El objetivo es simplificar los trámites de salud proporcionando un soporte común de la información.

2.1.11. Una plataforma de cuidado a domicilio.

Un servicio de cuidado a distancia permite a los profesionales de la salud controlar el estado de un paciente sin importar hora o lugar. El sistema está basado sobre Global Health Monitoring Platform, una plataforma interoperable de administración de datos sobre la cual los profesionales pueden comunicar y conectar todos los instrumentos de medida y de control necesarios para seguimiento de un paciente. Desarrollado en colaboración entre el

centro de investigación técnica de Finlandia (en inglés VTT Technical Research Centre) y el A*STAR (en inglés Agency for Science, Technology and Research) de Singapur. Este dispositivo se podría aplicar para seguir varias patologías como enfermedades cardíacas pero igualmente para el seguimiento del estado de salud de personas de edad avanzada. Muchas enfermedades necesitan ser seguidas de manera regular pero sin estar en un hospital para curarse.

2.2. Desarrollo de aplicación web

En esta parte se describen los Java Servlet, las paginas JSP, el API JDBC para comunicación con un DBMS, el DBMS MySQL, el patrón de diseño MVC y las interacciones AJAX.

2.2.1. Java Servlet

Un Servlet es un objeto que se ejecuta en un servidor de aplicaciones o contenedor JEE (Java Enterprise Edition), especialmente diseñado para ofrecer contenido dinámico desde un servidor Web, generalmente con representaciones HTML. Otras opciones que permiten generar contenido dinámico en páginas Web son los lenguajes ASP, PHP, JSP (un caso especial de Servlet que se describe en la sección 2.2.2. JavaServer Pages (JSP)) y Python.

Los Servlets forman parte de JEE (Java Enterprise Edition), que es una ampliación de JSE (Java Standard Edition).

Un Servlet es un objeto Java que implementa la interfaz `javax.servlet.Servlet` o hereda alguna de las clases más convenientes para un protocolo específico (ej: `javax.servlet.HttpServlet`) (Crawford & Hunter, 2002). Al implementar esta interfaz, el Servlet es capaz de interpretar los objetos de tipo `HttpServletRequest` y `HttpServletResponse` los cuales contienen la información de la página que invocó al Servlet.

Entre el servidor de aplicaciones (o administrador de contenido Web) y el Servlet existe un contrato que determina como han de interactuar. La especificación de éste se encuentra en los JSR (Java Specification Requests) del JCP (JavaCommunity Process).

A la fecha (Octubre de 2009), la especificación más reciente de los Servlets se documenta en el JSR 154, que define la versión 2.4 de los Servlets.

2.2.2. JavaServer Pages (JSP)

Los JSPs pueden considerarse como una manera alterna y simplificada de construir Servlets. Es por esto que un JSP puede hacer todo lo que un Servlet puede hacer, y viceversa. Cada versión de la especificación de JSP está fuertemente vinculada a una versión en particular de la especificación de Servlets.

El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en un archivo JSP para construir el código Java del Servlet a generar (Perry, 2004). Este Servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario (Figura 5).

Es posible enriquecer el lenguaje de etiquetas utilizado por JSP. Para ello podemos extender la capa de alto nivel JSP mediante la implementación de Librerías de Etiquetas (Tags Libraries). Un ejemplo de estas librerías son las proporcionadas por Sun bajo la denominación de JSTL o las distribuidas por Apache junto con el Framework de Struts.

TagLibs - JSP - Servidor Aplicaciones (Servlets) - Cliente (Navegador)

El rendimiento de una página JSP es el mismo que tendría el Servlet equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la maquina virtual compilará dinámicamente a código de maquina las partes de la aplicación que lo requieran (Perry, 2004). Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías Web que ejecutan el código de una manera puramente interpretada.

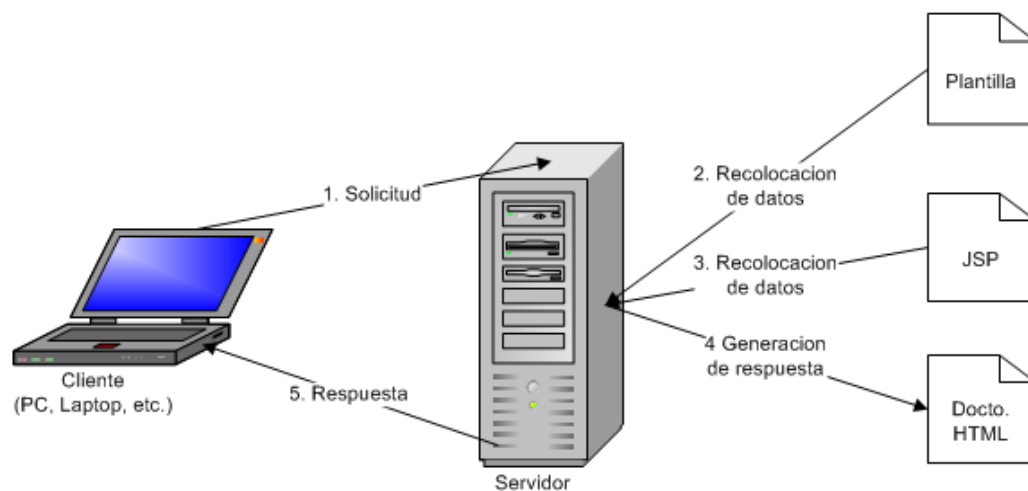


Figura 5: Funcionamiento de los JSP

Se envía al servidor una petición. El servidor busca en la base de datos y genera el Servlet, gracias al archivo JSP, que crea una página Web conteniendo lo solicitado por el cliente.

La principal ventaja de JSP frente a otros lenguajes es que el lenguaje Java es un lenguaje de propósito general que accede al mundo Web y que es apto para crear clases que manejen lógica de negocio y acceso a datos de una manera prolija. Esto permite separar en capas las aplicaciones Web, dejando la parte encargada de generar el documento HTML al archivo JSP.

Otra ventaja es que JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios. Es común incluso que los desarrolladores trabajen en una plataforma y que la aplicación termine siendo ejecutada en otra.

Los Servlets y Java Server Pages (JSPs) son dos métodos de creación de páginas Web dinámicas en el servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP, ASP o los CGIs (Common Gateway Interface), que también permiten escribir programas que generan páginas Web en el servidor. Sin embargo, se diferencian de ellos en otras cosas.

Para empezar, los JSPs y Servlets se ejecutan en una maquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de plataforma computacional, siempre que exista una maquina virtual Java para él. Cada Servlet se ejecuta en su propio hilo (thread), Es decir, en su propio contexto. Sin embargo, no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + interprete). Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo.

Los JSPs son en realidad Servlets: un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un Servlet. La principal diferencia entre los Servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un Servlet es un programa Java que recibe peticiones y genera a partir de ellas una página Web.

2.2.3. Java Database Connectivity (JDBC)

JDBC es una API que permite la ejecución de operaciones sobre bases de datos relacionales desde el lenguaje de programación Java independientemente del sistema operativo donde se ejecute o de la base de datos relacional a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice (Reese, 2001).

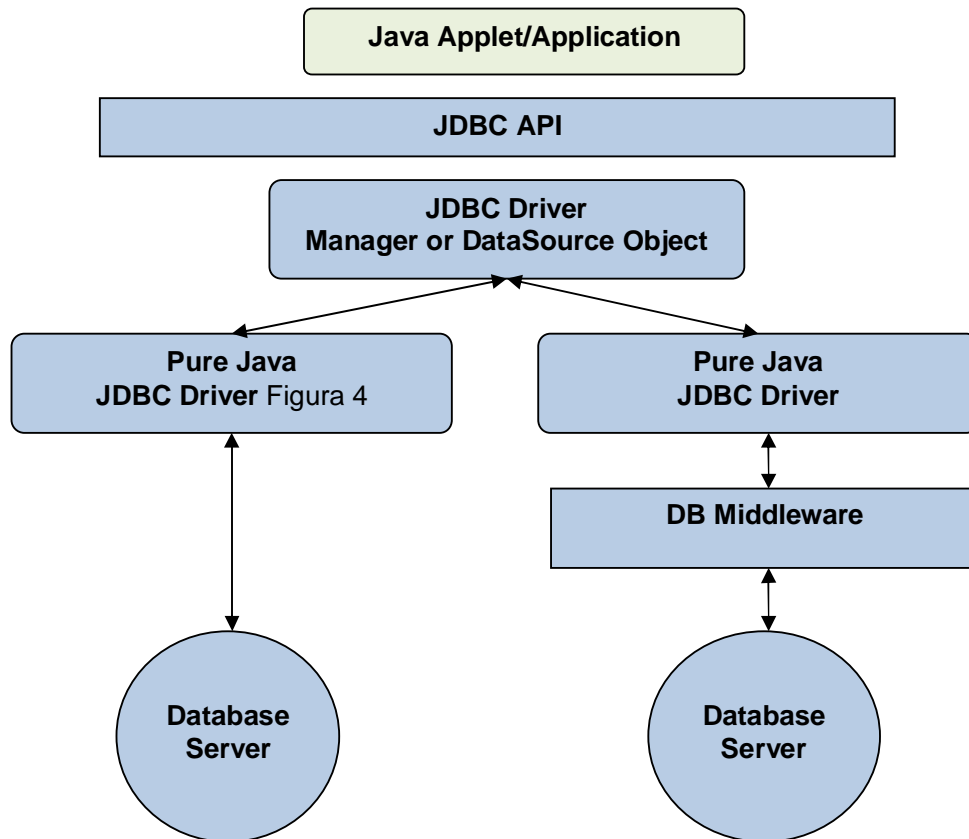


Figura 6: El API Java JDBC

Este Api permite la comunicación entre una aplicación y una base de datos. Si es una base de datos distribuida requiere un middleware para ayudar la comunicación

El API JDBC (Figura 6) se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Un manejador de conexiones hacia un DBMS en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos que pueden manejar. Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la librería de conexión apropiada al DBMS específico, y accede a ella estableciendo una conexión, para ello provee un localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar cualquier tipo de tareas con la base de datos a las que tenga permiso: consulta, actualización, creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.

2.2.4. MySQL

MySQL es un sistema para la administración de base de datos (DBMS) relacionales, multi-hilo y multiusuario con más de seis millones de instalaciones (Dubois, 2000). Por un lado se ofrece bajo la licencia de uso GNU GPL para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privados pueden comprar a la

empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C. Al contrario que proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privadas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet.

De la Tabla 4, podemos ver que existen muchos DBMS y que cada uno tiene sus ventajas y desventajas. Si se quiere un DBMS de manera gratis, hay varias posibilidades. Podríamos tomar DBMS tal que InterBase, Firebird, H2, MySQL, PostgreSQL, MaxDB, SQLite, etc. Si se desea sencillez y velocidad, una buena alternativa es MySQL (Dubois, 2000). Además de eso, su característica de fácil integración con Apache le hace apropiado en aplicaciones Web.

Editor	DBMS	Licencia	Ventaja o Desventaja
Apache	Derby	Open Source Apache 2	Escrito en Java Herramienta grafica minimalista.
CodeGear	InterBase	OpenSource (MPL) o Comercial	Sencillez de Administración Problema con los grandes volúmenes. Carece de muchas funciones
CodeGear	FileMaker	Comercial	Sencillez
FirebirdSQL Foundation	Firebird	OpenSource (IBPL, IDPL)	Sencillez de Administración
FyOracle	H2	OpenSource (MPL)	Sencillez de utilización y administración Mala gestión de la memoria Problema con grandes volúmenes de datos
IBM	DB2-400	Comercial, gratis por su versión Express	Escrito sobre OS400. Una tabla es un archivo físico; una vista o un índice es un archivo lógico.
IBM	DB2-UDB	Comercial, gratis por su versión Express-C	SQL cerca de la norma Gestión centralizada de varias instancias. Costoso pero con muy buen rendimiento.
IBM	Informix	Comercial	Administración muy sencilla y funcional Buen rendimiento
Microsoft	SQL Server	Comercial	Administración muy fácil Muy buen rendimiento Distribución ligada con el sistema operativo.
Microsoft	Access	Comercial	Muy poderoso. Mucha ayuda Gasta mucha memoria y no funciona muy bien para aplicaciones a distancia.
Oracle Corporation	Oracle DataBase	Comercial, gratis por su versión Express	Riqueza funcional Gestión centralizada de varias instancias. Precio alto Administración compleja
Oracle Corporation	Berkeley DB	Comercial o OpenSource	No reconoce el SQL
Oracle Corporation	Hyperion EssBase	Comercial	Modelización posible Precio alto.

Oracle Corporation	MySQL	GPL y comercial	Integración fácil con el entorno Apache Fácil de desplegar y usar. Poca funcionalidad Problema con los volúmenes.
PostgreSQL	PostgreSQL	BSD y comercial	Fiable, buen rendimiento Simple de uso y de administración No tiene servicio Web.
SAP	MaxDB	GPL y comercial	Administración fácil
SQLite	SQLite	BSD	Simple de uso y de administración Problema con los volúmenes.
Sybase	Adaptive Server Enterprise	Comercial, gratis por Linux en su versión Express	No necesita muchos recursos para funcionar. Administración compleja.
TeraData	TeraData	Comercial	Administración simple Precio muy elevado.
4e Dimension	4e Dimension	Comercial	Sencillez de administración

Tabla 4: Licencia y Ventajas o Desventajas de DBMS existentes

2.2.5. Patrón de arquitectura de software Modelo Vista Controlador

En ciencias computacionales y en desarrollo de aplicación Web en particular, se busca la separación de aspectos (en inglés: separation of concerns o SoC), por lo que resulta adecuado construir las aplicaciones mediante varios módulos según su funcionalidad o su comportamiento.

El patrón de arquitectura de software Modelo Vista Controlador separa los datos, la interfaz de usuario y la lógica de control en tres componentes distintos. Ese patrón se ve muy frecuentemente en aplicaciones Web. El patrón fue descrito por primera vez en 1979 por Trygve Reenskaug (Reenskaug, 2003).

El modelo es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos. La vista presenta el modelo en un formato adecuado para interactuar, usualmente es la interfaz de usuario. El controlador responde a eventos, usualmente acciones del usuario e invocaciones provocadas por cambios en el modelo y para probablemente actualizar la vista (ver Figura 7).

Muchos sistemas informáticos utilizan un sistema de administración de base de datos para gestionar los datos: MVC corresponde a este modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y accesos a datos, MVC no pretende discriminar entre capa de negocio de capa de presentación pero si pretende separar la capa visual grafica de su correspondiente programación y acceso a datos algo que mejora el desarrollo y mantenimiento de la vista y del controlador en paralelo ya que ambos cumplen ciclos de vida muy distintos entre sí.

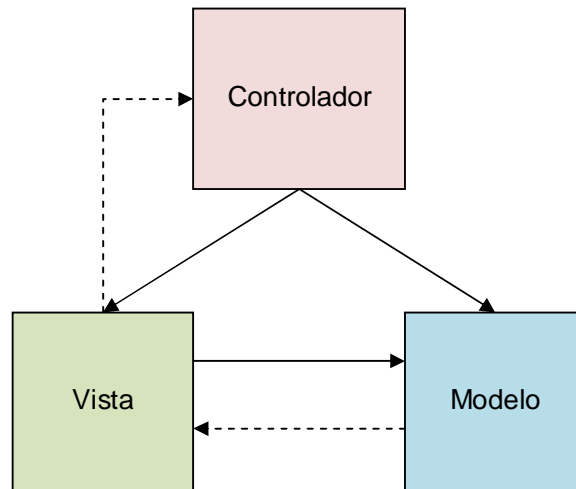


Figura 7: Patrón MVC

Las líneas solidas indican una asociación directa y las punteadas una indirecta.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz (vista) de usuario de alguna forma (por ejemplo, el usuario pulsa un botón)
- El controlador recibe (por parte de los objetos de la interfaz vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos son a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirectación entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en si mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio a la vista aunque puede dar la orden a la vista para que se actualice. En algunas implementaciones, la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Con Java Enterprise Edition es muy fácil de utilizar ese patrón de diseño (Figura 8). El modelo sería un sistema de administración de base de datos, la vista la interfaz del cliente

construida a partir de páginas JSP. El controlador serian los Servlets, un conjunto de clases y las paginas JSP que construyen la vista a partir de la información recibido del Servlet.

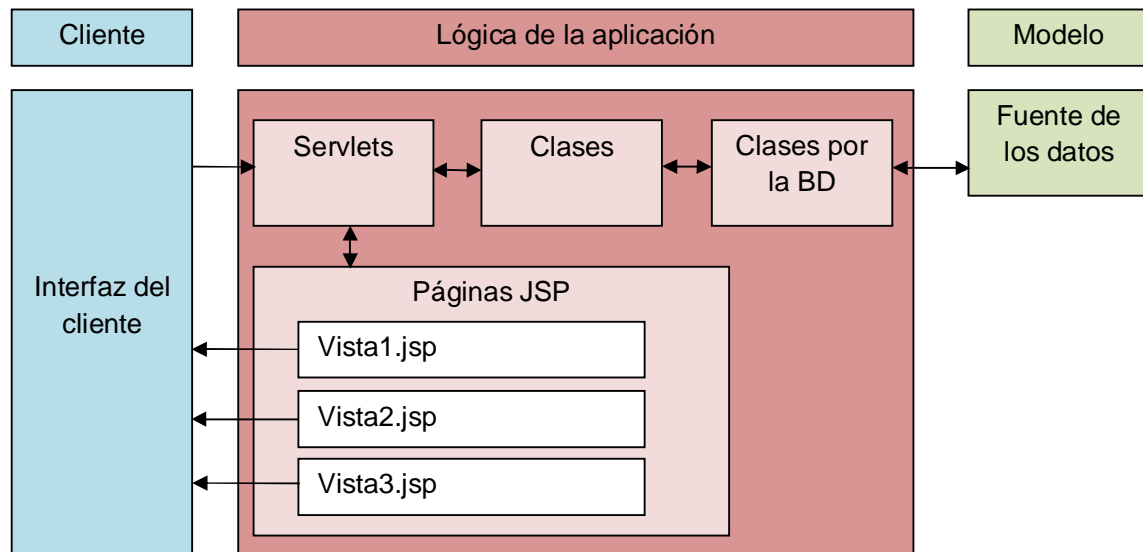


Figura 8: Patrón MVC con Java

Las paginas JSP se pueden ver a partir de sus entradas y sus salidas. Las entradas son los datos que el Servlet provee a la página JSP y se pueden encontrar en la petición (request) o en la sesión (session). Las salidas son los datos que la páginas JSP provee al Servlet y se puede encontrar a partir de un archivo HTML y el Servlet lo recuperara haciendo una operación del tipo `request.getParameter(...)`.

La ventaja de ese modelo es la claridad de la arquitectura que simplifica la tarea del desarrollador que necesita efectuar el mantenimiento o una mejora del proyecto. La modificación de los formatos de datos no cambia nada a la vista. Por ejemplo, se puede pasar de una base de datos de SQL a XML cambiando simplemente la forma de interacción con la base de datos y las vistas no son afectadas.

El MVC ha mostrado sus limitaciones con aplicaciones utilizando las tecnologías Web construidas a partir de servidores de aplicaciones. Capas adicionales y mecanismos de inversión de control y de inyección de dependencia tienen que ser introducidos. Este es el tema de la siguiente sección.

2.2.6. Ajax con Java

Utilizando la tecnología JavaScript, una página HTML puede de manera asíncrona llamar al servidor del cual ha sido cargado y traer información en documentos XML, en contenido HTML, en texto puro, o en JavaScript Object Notation (JSON). La tecnología JavaScript puede después utilizar el contenido por modificar o refrescar el modelo del objeto del documento (en inglés Document Object Model o DOM) de la página HTML. El término Asynchronous JavaScript Technology and XML (Ajax) describe ese modelo de interacción.

Ajax no es algo nuevo, esas tecnologías han estado disponibles por muchos años. Esa tecnología se llamaba “Web remoting” o “remote scripting”. Los desarrolladores Web también pudieran utilizar una combinación de plug-ins, Java Applets y frames ocultos para emular ese modelo de interacción. La inclusión del soporte del objeto XMLHttpRequest en la ejecución de JavaScript de los navegadores más importantes ha cambiado todo. En la especificación de la tecnología JavaScript no se encuentra ese objeto pero hoy la mayoría de los navegadores lo soportan (Murray, 2006).

Lo que hace a los clientes basados en Ajax únicos, es que la tecnología JavaScript se encarga de la lógica de control del contenido de la página. La página interactúa con la tecnología JavaScript con eventos como la carga del documento, el clic del ratón, el cambio de foco o hasta un reloj. Las interacciones Ajax permiten una clara separación de la lógica de presentación de los datos. Ajax requiere una parte del servidor diferente para soportar ese modelo de interacción. Tradicionalmente, la parte servidor de las aplicaciones Web se enfocan en la generación de documentos HTML por cada evento de un cliente resultado de una llamada al servidor. Los clientes podrían después hacer un refresco de la página y recuperar de nuevo la página HTML completa por cada llamada. Las aplicaciones de Internet enriquecidas se enfocan en la generación de documentos HTML como contenedores en los cuales se puede inyectar datos. A partir de eventos del cliente, un componente del servidor provee datos XML los cuales pueden ser modificados para inyectarlos en su contenedor HTML.

Ajax se utiliza para la validación en tiempo real de datos, auto-terminación de datos, refresco de datos y servidor push, a la página de la aplicación.

Se puede utilizar Ajax con las tecnologías que hemos descrito antes. Una aplicación Web contiene una página HTML estática o una página HTML generada por la tecnología JSP que contiene un formulario HTML u otra cosa que necesite una lógica del lado del servidor sin hacer un refresco de la página. Un componente Web del lado servidor (Servlet) podría proveer la lógica necesaria.

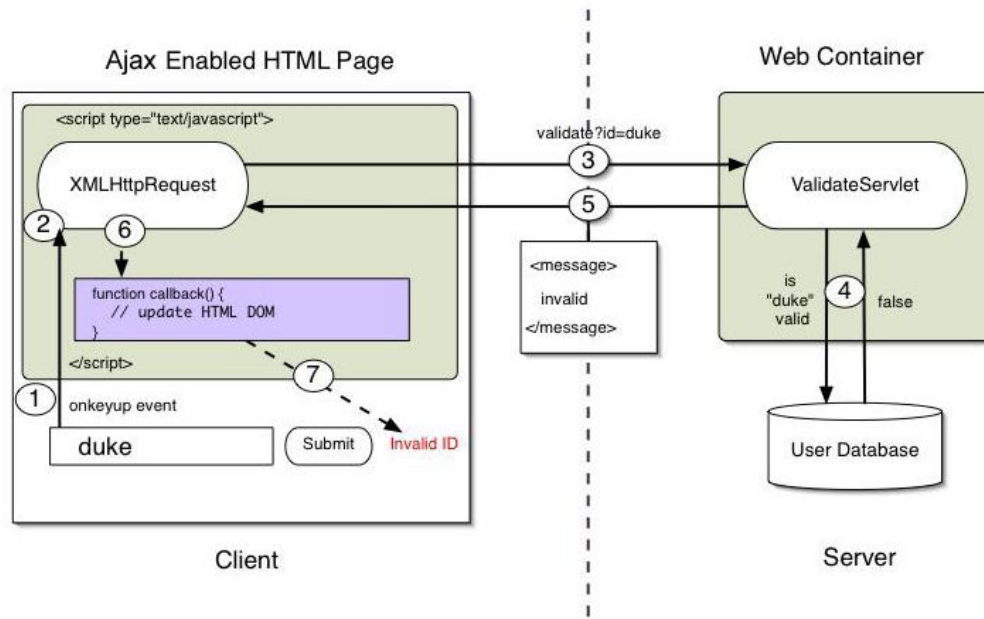


Figura 9: Ejemplo de interacción Ajax

Hay 7 pasos entre el inicio y el final de la interacción Ajax (Figura 9):

1. El evento ocurre
2. Se crea el objeto XMLHttpRequest
3. El objeto XMLHttpRequest hace una llamada al servidor
4. La llamada es tratada por un Servlet en la parte del servidor
5. El Servlet regresa un documento XML
6. El objeto XMLHttpRequest llama la función callback() y procesa el resultado
7. El DOM de la página es modificado

La Figura 9 muestra un ejemplo de aplicación que utiliza Ajax para verificar la validez de un formulario. Si el nombre está en la base de datos, es válido sino no lo es. A partir del clic del botón submit se ejecutan los 7 pasos para que aparezca o no la frase "Invalid ID". La llamada al servidor es del tipo GET entonces todas las variables se encuentran en la llamada de la forma siguiente:

nombre_servlet?variable1=valor1&variable2=valor2...

La respuesta del servidor es construido en el Servlet gracias al objeto HttpServletResponse. En el ejemplo, el documento XML de respuesta es muy sencillo pero si la respuesta tiene que contener los elementos de la base de datos el DOM del documento XML puede ser mucho más complejo. Antes de construir la respuesta con la función `getWriter().write(...)` de HttpServletResponse, se necesita inicializar el tipo del contenido y el encabezado. Básicamente la construcción de la respuesta se hace de la forma siguiente:

```
response.setContentType("text/xml");
response.setHeader("Cache-Control", "no-cache");
response.getWriter().write("<message>invalid</message>");
```

Una vez recibido la respuesta del servidor, con la tecnología JavaScript se puede acceder a la información gracias a las etiquetas. Se necesita conocer la arborescencia del documento XML.

La tecnología JavaScript puede acceder a cualquier elemento de la página HTML gracias al DOM. La manera más simple es acceder a un elemento con su Id y modificar su propiedad innerHTML de la siguiente forma:

```
Var elemento = document.getElementById("Id_elemento");  
elemento.innerHTML = "Invalid User"
```

El innerHTML cambia el contenido del elemento. Entonces la manera habitual es tener en la página web varios `<div id="Id_elemento"></div>` vacíos donde se inyectará la información recibida del servidor. La sugerencia es tener un id diferente por cada `<div>` así se puede agregar la información en varios sitios de la página Web.

2.3. Protocolos y desarrollo de aplicación móvil

En esta parte, vamos a ver la plataforma Java Micro Edition, el protocolo CDMA450 y el API Gammu. Éste último permite la utilización de un teléfono celular conectado al servidor.

2.3.1. Java Micro Edition

La plataforma Java Micro Edition (JME) (Figura 10), o anteriormente Java 2 Micro Edition (J2ME), es una especificación de un subconjunto de la plataforma Java orientada a proveer una colección certificada de APIs de desarrollo de software para dispositivos con recursos restringidos. Está orientada a productos de consumo como PDAs, teléfonos móviles o electrodomésticos (Hamphill & White, 2002).

High-end PDA's
TV set-top boxes
Embedded device

Mobile phones
& entry-level
PDAs

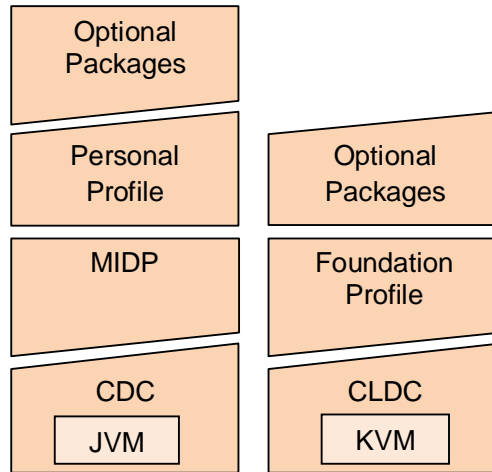


Figura 10: El Java Micro Edition o JME

Es muy importante entender la configuración de la plataforma porque define el entorno de ejecución como un conjunto de clases y una máquina virtual java (JVM). Por el momento, existen dos tipos de configuraciones por JME (ver Figura 10). La configuración limitada o Connected Limited Device Configuration (CLDC) y la configuración no limitada o Connected Device Configuration (CDC). La configuración CLDC es utilizada específicamente por la máquina virtual K (KVM) y se sugiere para arquitecturas de 16-bits o 32-bits con una memoria limitada. Esa configuración (y la máquina virtual) se utilizan por aplicaciones JME de tamaño pequeño. La configuración CDC se utiliza con la máquina virtual C (CVM) y se usa en arquitecturas de 32-bits que necesitan más de 2MB de memoria. La Figura 11 compara JSE, CDC y CLDC. Se puede ver que la configuración CLDC de JME está incluida en CDC. La configuración CDC de JME no está totalmente incluida en JSE, es decir, tiene funcionalidades adicionales.

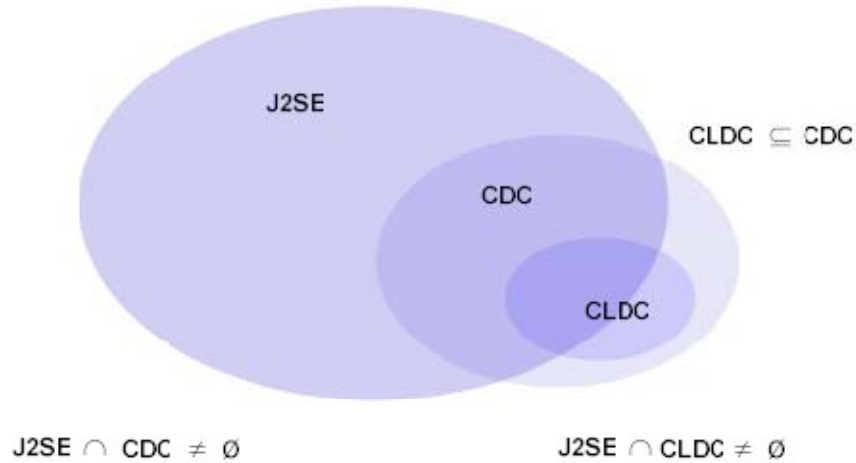


Figura 11: Comparación entre J2SE, CDC y CLDC

Además de la configuración, el perfil define el tipo de herramienta soportada por la aplicación. Específicamente agrega clases específicas del dominio. Dos perfiles fueron contruidos sobre CLDC. El KJava y el Mobile Information Device Profile (MIDP). En nuestro proyecto vamos a utilizar el perfil MIDP, por ser el que nos brinda la mejor opción, de acuerdo a como se describe a continuación.

MIDP propone 2 librerías para construir una interfaz grafica o IHM. Una bajo nivel que permite dibujar. La otra, de alto nivel, permite agregar componentes visuales como los de AWT o SWING. Las clases se encuentran en el paquete javax.microedition.lcdui.

Para construir una IHM en J2ME, se necesita entender que se puede ver nada más una pantalla a la vez. La pantalla lógica se llama un display y se incluye lo que va a contener según cual tipo de display se necesita. La Figura 12 muestra las diferentes pantallas lógicas de las dos librerías. Todos los elementos graficas heredan de Displayable.

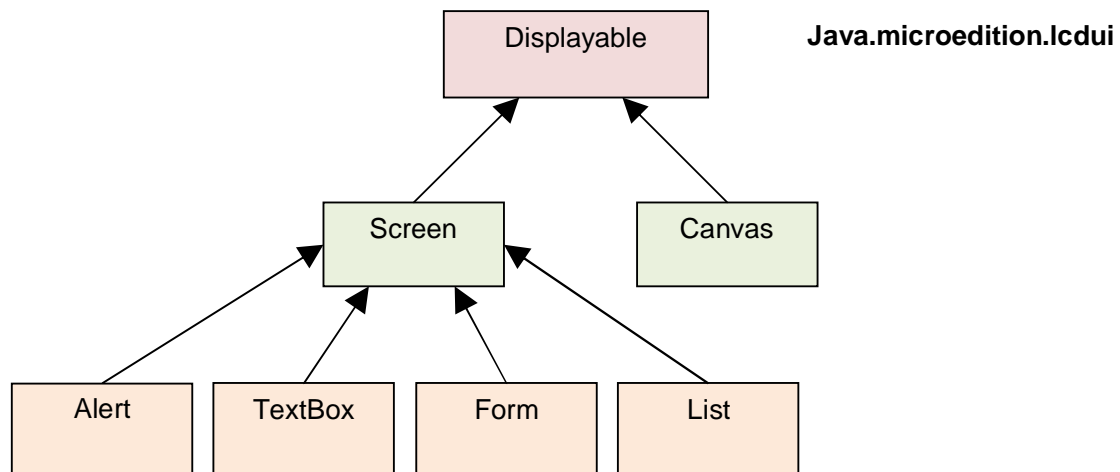


Figura 12: JME – Los diferentes sub-tipos Displayable

Canvas es un componente sobre el cual se puede dibujar; se utiliza a través de un API de bajo nivel. La pantalla TextBox es una pantalla que contiene una zona de texto multi-líneas. El objeto List permite incluir una lista de varios tipos: botón radio o botón check. El objeto Form es el componente estándar para construir una pantalla con diferentes elementos o ítems (ver la Figura 13).

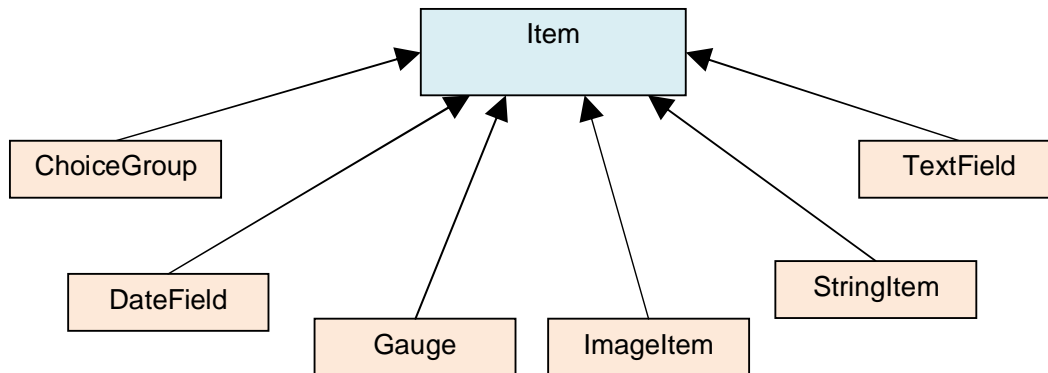


Figura 13: JME – Los diferentes sub-tipos Item

Para lograr que los objetos en la pantalla interactúen, se tiene que añadir algunos comandos de la clase Command y un Listener. Los Commands se agregan al Displayable gracias a la función addCommand() y el Listener se agrega al Displayable gracias a la función setCommandListener(). El Listener puede activar la función commandAction().

En JME, no hay eventos, al momento de la interacción; se necesita todo el contexto.

Java ME se ha convertido en una buena opción para crear juegos en teléfonos móviles debido a que se puede emular en un PC durante la fase de desarrollo y luego subirlos fácilmente al teléfono. Al utilizar tecnologías Java el desarrollo de aplicaciones o videojuegos con estas APIs resulta bastante económico de portar a otros dispositivos (Hamphill & White, 2002).

2.3.2. CDMA 450

Permite proveer servicios de voz a través de la telefonía básica; y datos por medios de la tecnología asociada EVDO. El ancho de banda puede alcanzar hasta 2.4 megabits por segundo. Permite cubrir todos los segmentos del mercado, desde corporativo a Pyme. Es compatible y complementaria a las redes de telefonía ya existentes (Garavaglia, Murray, Baker, Chevalier, & Brunner, 2006).

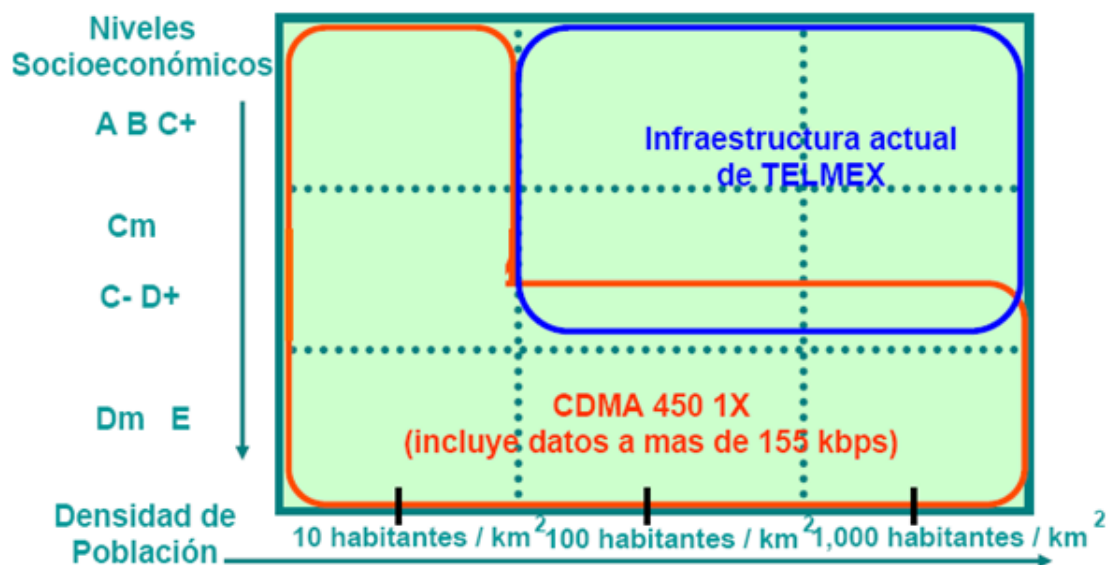
Cobertura y costos:

Un equipo cubre entre 45 y 200 kilómetros de radio, es decir, 3 veces más lo que cubre la frecuencia de la telefonía celular de 900 MHz, y 11 veces más la de 1,900 MHz (Henni, 2003). Es decir, que la cantidad de bases de radio que se necesitan para cubrir determinadas áreas es menor a la de otras tecnologías disponibles.

Con una tecnología GSM900, la población rural no tendría acceso a la red telefónica inalámbrica que está disponible únicamente en zona de alto densidad de población. Como no se necesita una cantidad menor de bases de radio para cubrir una distancia mayor, la red cuesta mucho menos que antes y se pone disponible por una población de bajo ingreso (Figura 14).

En México, las frecuencias de 450MHz presentan mucha interferencia ya que en su mayoría son utilizadas por radioaficionados, policías locales, ejército y dependencias de gobierno.

Por tanto los FILTROS fueron la mejor solución para este problema. Con filtros ANF (Adaptive Notch Filter) de anti-interferencia, las llamadas pueden realizarse a 98Km de la BTS, generalmente esto se logra con terminales fijas ya que en varios de los sitios la interferencia es muy intensa.



Innovación y Estrategia

Telmex Mayo 2007

Figura 14: La red Telmex

La nueva red Telmex aumentará la posibilidad de acceso a regiones de densidad de hasta 10 habitantes por kilómetros y con ingresos inferiores gracias a la tecnología CDMA 450.

La nueva tecnología no permitirá una cobertura total del país (Figura 15) pero mejora mucho la disponibilidad y el acceso a la tecnología. Abre la puerta de la modernidad a un tipo de población que no tenía antes la posibilidad.

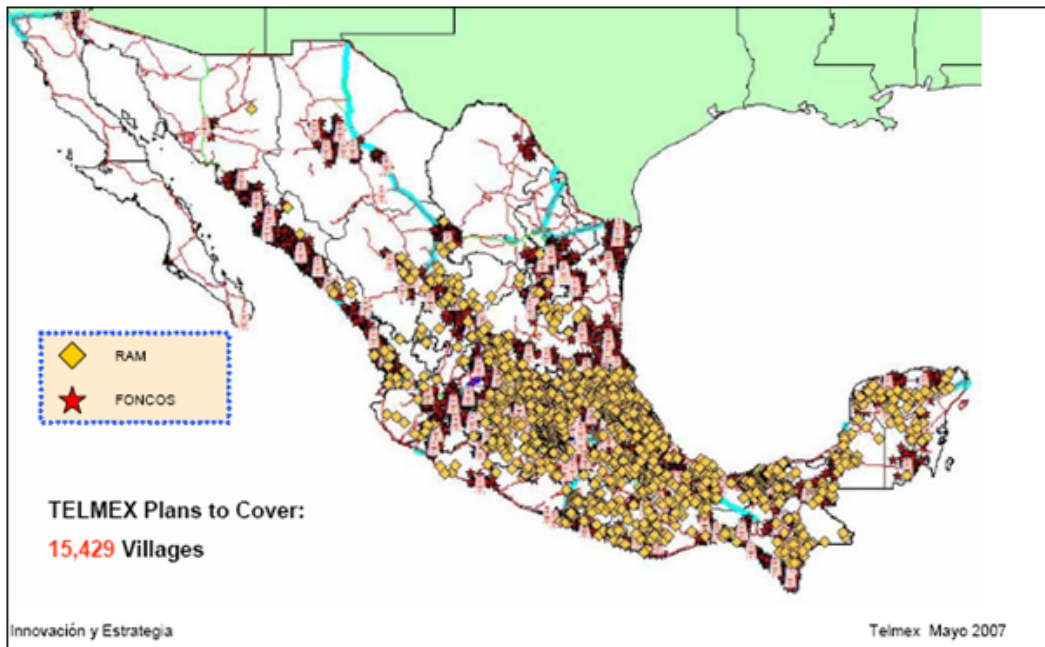


Figura 15: Mapa de la Red Telmex

2.3.3. Gammu

Gammu (**GNU All Mobile Management Utilities**) es un proyecto para crear una herramienta apropiada para desarrollar aplicaciones para interacción con teléfonos celulares o herramientas similares. Las funciones disponibles dependen del celular utilizado y del medio de conexión con una computadora. Al inicio fue una aplicación solo accesible mediante una línea de comandos, pero con los años desarrollaron una API para utilizar las funciones de Gammu en proyectos. Se puede utilizar Gammu con lenguajes de programación PHP, C o C++ mediante las librerías que se pueden descargar desde el sitio que la desarrollan actualmente. Aquí solo nos referiremos al uso de Gammu con C.

En lenguaje C, se tienen que agregar las funciones Gammu en la carpeta lib de su distribución, e incluir al inicio de su programa las funciones Gammu con "include <gammu.h>". Contiene muchos tipos de módulos (ver Tabla 5); por ejemplo se puede manipular la información del celular, las llamadas, la memoria, los SMS del celular conectado al computadora, etc.

Módulos disponibles con Gammu	Utilidad
Backup	Manipulación de los backups
Bitmap	Manipulación de bitmaps
Calendar	Manipulación de eventos Calendar
Note	Manipulación de Notes
Todo	Manipulación de entrada Todo
Call	Manipulación de llamadas
Divert	Manipulación de entradas de diversión

Callback	Capa para llamadas a máquina de estados genérica.
Category	Manipulador de Categories
Date and time	Manipulador de Date and time
Debug	Manipulador de Debuging
Error handling	Manipulación y manipulador de Error
File	Manipulador de Files
Info	Información del Phone
INI files	Esas funciones “parsean” un archivo ini y les hace accesible de una manera sencilla. El formato del archivo es el estándar archivo ini. Comentarios son # y ;
Keys	Manipulación del Keyboard
Limits	Definición de los limites
Memory	Manipulación de las entradas de memoria
Messages	Manipulación de mensajes USSD, CB, SMS y MMS
Nokia	Manipulación Nokia
Ringtone	Manipulación Ringtone
Security	Operación con un teléfono relacionadas con la seguridad
Settings	Manipulación de las características de un teléfono
SMSD	Manipulación del daemon SMS
State machine	Capa de máquina de estados genérica
Types	Definición de varios tipos útiles
Unicode	Funciones de manipulación de Unicode
WAP	WAP bookmarks y manipulación de settings

Tabla 5: Módulos de funciones de Gammu

Para que la conexión del celular a la computadora funcione, se tiene que escribir un archivo de configuración de manera manual llamado gammurc. Ese archivo se puede poner en la carpeta de Gammu o en la carpeta del proyecto en desarrollo. Se necesita configurar el tipo de celular conectado y el medio utilizado para conectarse. La conexión del celular a la computadora se puede hacer mediante varios medios. Por ejemplo, se puede hacer vía conexión infrarrojo; hay dos métodos infrared o irda. El método irda es mucho mejor y es compatible con todos los sistemas operativos modernos. Se puede hacer vía conexión bluetooth. Posiblemente la mejor solución pero se necesita tener una antena bluetooth en la computadora. También mediante una conexión USB. Para cada uno de esos medios, se necesita verificar el tipo de la conexión, la cual depende de la marca del celular. La Figura 16 muestra un ejemplo de archivo de configuración para un celular Sony Ericsson conectado a la computadora mediante una conexión USB.

```

[gammu]
port = com10:
model = at
connection = at19200
synchronizetime = yes
logfile = gammulog
logformat = textall
use_locking = yes
gammuloc = locfile
startinfo = yes
gammucoding = utf8

```

Figura 16: Ejemplo de archivo gammurc

Como lo muestra la Tabla 6, Gammu es compatible con más de 900 modelos de celulares. Las marcas más importantes son Nokia, Sony Ericsson y Samsung. Por cada uno de esos celulares, hay un tipo de funciones disponibles; por lo cual, antes de empezar, se necesita verificar el funcionamiento de las funciones del celular que será utilizado.

Fabricante	Celulares compatibles con Gammu
Alcatel	17
Appel	1
BenQ-Siemens	10
Elson	1
Emgeton	0
Falcom	15
Fly	1
Gionee	1
HTC	14
Huawei	15
Hughes	1
Jinpeng	0
Lenovo	1
LG	52
Mitsubishi	2
Motorola	90
Nokia	256
Onda	0
Openmoko	1
Option	6
PalmOne	1
Philips	4
Sagem	36
Samsung	136
Sanyo	0
SciPhone	7
Sharp	6
Siemens	56
SierraWireless	5
Simcom	2

Sony Ericsson	170
Teltonika	1
Toshiba	1
Vodafone	0
Wavecom	4
ZTE	3
Total	916

Tabla 6: Modelos de celulares compatibles con Gammu para diversos fabricantes

2.4. Resumen del Capitulo

A lo largo del capítulo se abordaron principalmente tres temas, el estado del arte en Telemedicina, el desarrollo de aplicaciones Web y el desarrollo de aplicaciones para móvil. El principal interés era conocer el ámbito en el cual viene esta tesis. El segundo interés de esa investigación era buscar tecnologías capaces de responder a la propuesta de sistema del capítulo 1.

Hemos visto que el protocolo CDMA 450 podría permitir la comunicación entre un paciente en una zona rural y un medico en la zona urbana. Hemos visto que se puede construir un sistema muy poderoso con la tecnología Java utilizando Servlet y JSP, y que junto con el patrón MVC se puede mejorar el desarrollo de un sistema. Además de eso hemos visto el énfasis de JME para construir una aplicación sobre aparatos con memoria limitada. Un problema residual era como ligar un celular con una computadora para guardar los datos. La biblioteca Gammu permite resolver ese problema con un número importante de celulares (916 en Noviembre 2009).

El siguiente capítulo describe por lo tanto, la esencia del sistema que podría resolver el problema explicado en el capítulo 1.

Capítulo 3. Análisis del sistema

En este capítulo, se analizara el sistema de atención médica de manera global; un análisis más específico se hace en el capítulo siguiente.

3.1. Contexto del problema

La situación actual de México de la atención médica a personas que viven en áreas rurales obliga una cierta organización del sistema de salud. Sería utópico creer que se puede instalar un medico en cada pueblo de México. Primero porque hay muchos pueblos en México y segundo porque ningún médico aceptaría aislarse en un pueblo. Pero la idea de tener una oficina en cada pueblo es una posibilidad que aquí se explora, una parte de lo cuál sería a quién se encargaría.

Nuestra propuesta es que se necesita instalar a médicos en la ciudad para que funcione la idea. Posiblemente utilizar el año de servicio social para tener más médicos. Sería entonces posible una asistencia médica profesional a distancia. Si se quiere comparar esa asistencia con los estándares que existen en la ciudad claro que sería siempre mejor tener una relación paciente medico lo más cerca posible. Entonces la mejor relación posible que pudiera existir para la zona rural sería una relación vía la telefonía.

Los médicos podrían acceder al sistema informático médico vía un navegador Web y cada uno tendría su teléfono para llamar al paciente. En cada pueblo, sería instalada una oficina de la cual se encargara un paramédico o una persona del pueblo que hubiera sido formada sobre el sistema. Todas las comunicaciones vocales se harían gracias al protocolo CDM450; los pacientes no tendrían otra posibilidad de acceso a la red de telefonía. Las comunicaciones de datos se harían vía dos tipos de SMS. El primero para registrarse en la pila de espera y el segundo tipo para enviar la información para el diagnostico del médico. En cada oficina de los pueblos se necesitarían dos celulares; uno para las comunicaciones vocales y otro para las comunicaciones por SMS. La causa de eso es que no se puede enviar un SMS cuando ya hay una comunicación vocal.

El sistema informático tiene que conocer cada una de sus oficinas para poder hacer el mapeo entre un número de teléfono que envía un SMS, una oficina y un numero a llamar para el diagnostico.

En la sección que sigue, se muestra la arquitectura del sistema, según lo que acabamos de plantear.

3.2. Arquitectura del sistema

La Figura 17 muestra la organización del sistema según el contexto. El sistema se compone de tres sub-sistemas. El sub-sistema del lado del paciente, el cual estaría repetido en cada pueblo. Se necesita disminuir al mínimo la funcionalidad de ese componente. Como ya se mencionó antes, va a constar de dos celulares. Uno que contiene una aplicación desarrollada en JME y que permite hacer el registro del paciente al sistema y el envío de los datos. El segundo permite la comunicación vocal con el médico. El sub-sistema del lado del médico reside en el servidor y se carga a la computadora del médico mediante el navegador Web. En el lado del médico se tendrían dos componentes: una computadora con un navegador Web y un teléfono. El teléfono permitiría llamar al paciente y el navegador permitiría acceder a la aplicación sin tenerla instalada en el computadora del médico. El último sub-sistema estaría encargado de la recuperación de los datos. Incluye un teléfono para recibir SMS y también una aplicación que permitiría la recuperación y el tratamiento en continuo de los SMS recibidos.

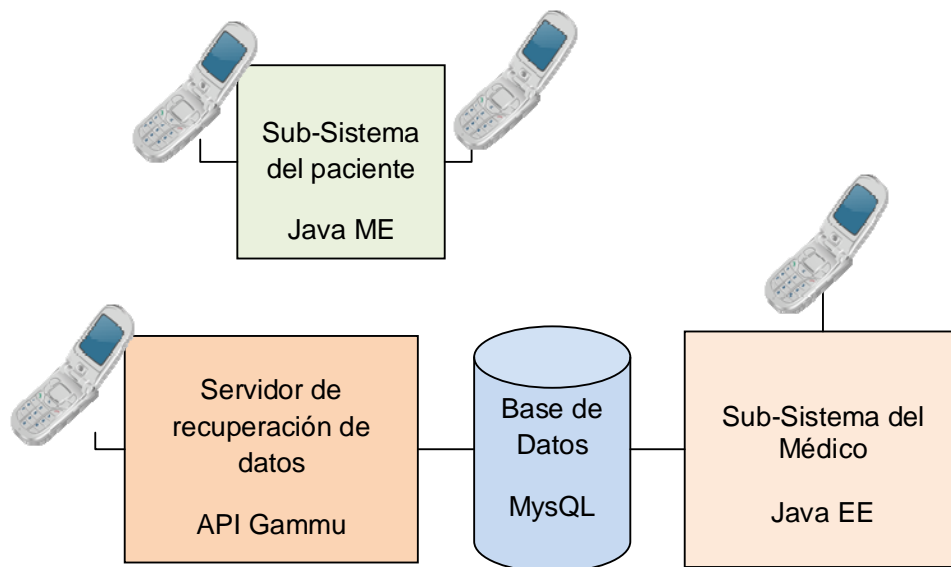


Figura 17: Organización del sistema

La Figura 18 muestra la comunicación entre los diferentes sub-sistemas. El paciente envía un SMS para registrarse en el sistema. El servidor de recuperación de los datos recibe el SMS y lo guarda en la base de datos de los pacientes en espera de recibir una comunicación por parte de un médico. El paciente tiene que esperar hasta que un médico lo elige y lo llama. Para completar su diagnóstico, el médico necesitaría el envío de datos adicionales de las condiciones de salud del paciente. Un segundo tipo de SMS sería enviado al servidor de recuperación de los datos, el cual puede distinguir los dos tipos de SMS. El primer tipo de SMS permite registrar la consulta en la fila de espera, el segundo tipo de SMS permite guardar los datos del paciente en la base de datos.

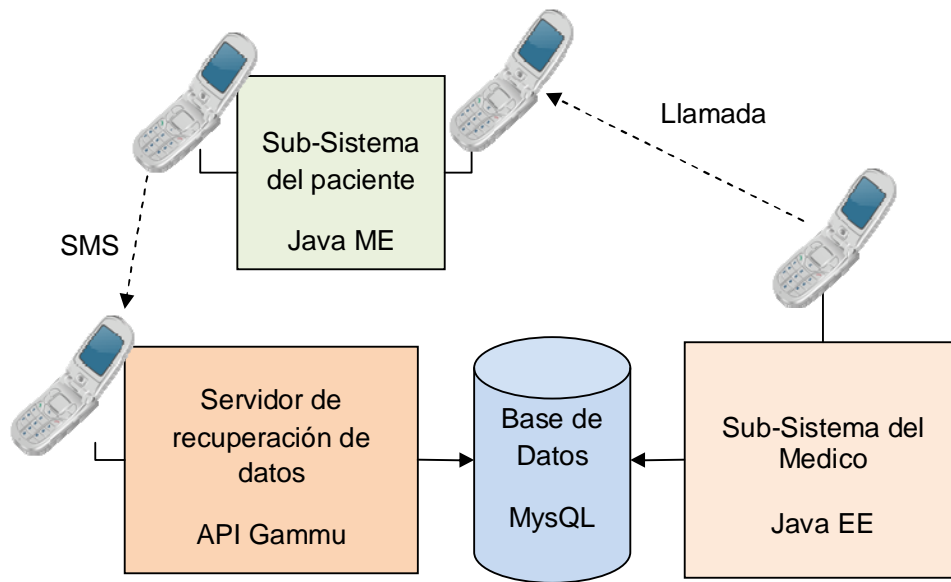


Figura 18: Comunicaciones del sistema.

3.3. Casos de Uso

La Figura 19 muestra los diferentes casos de uso del sistema. En el sistema, hay dos usuarios, el paciente y el médico, que pueden interactuar con el sistema de diferentes maneras. El paciente puede registrarse al sistema vía el envío de un SMS y enviar sus datos por el diagnóstico. Además puede recibir una llamada. El médico puede conectarse al sistema, elegir una llamada, llamar al paciente, leer los SMS del paciente y guardar en el histórico el diagnóstico del paciente. Además de eso puede buscar los datos de un paciente y administrar los datos de los pacientes, puede también crear y suprimir médicos y modificar su propia información.

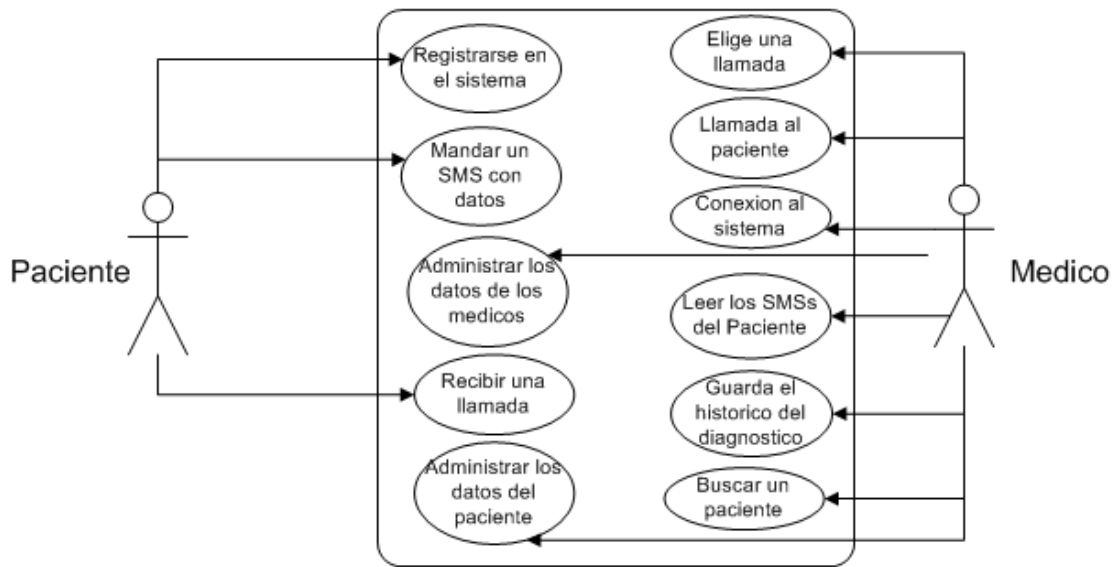


Figura 19: Casos de Uso del sistema

3.4. Diagrama de Clases

Las clases involucradas en el funcionamiento del sistema se muestran en la Figura 20 y se detallan a continuación:

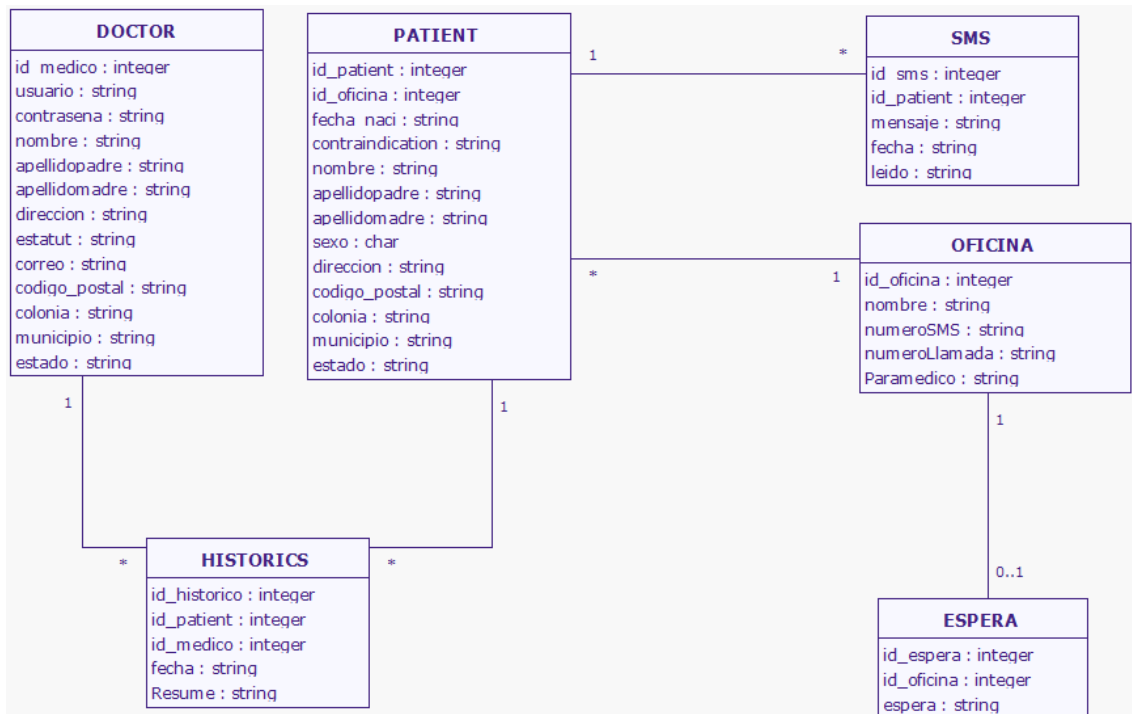


Figura 20: Diagrama de clases

- Doctor. Entidad que contiene información relevante del médico que tiene acceso al sistema. Se almacenan sus datos personales, su clave de usuario y contraseña.
- Patient. Entidad que contiene información relevante del paciente que tiene una cuenta en el sistema. Se almacenan datos personales y de su consultorio.
- Historics. Entidad que contiene información relevante del histórico de los diagnósticos de un paciente.
- SMS. Entidad que contiene información relevante de los SMS que ha enviado un paciente.
- Oficina. Entidad que contiene información relevante por las oficinas registradas en el sistema. Contiene información tal como los números de llamadas o de envíos de SMS.
- Espera. Entidad que contiene información relevante por la fila de solicitudes de asistencia enviadas por SMS y que están pendiente de ser atendidas en las oficinas de los médicos registrados en sistema.

3.5. Diccionario de datos

Mientras que con el diagrama de clases es muy sencillo identificar la relación de los elementos principales del sistema, con el diccionario de datos podemos entender la estructura de la base de datos, conocer sus tablas y los atributos de los campos que la conforman. Todo esto se muestra en la Tabla 7.

Doctor

Campo	Tipo	Nulo	Comentarios
id_medico	integer	No	id del medico
usuario	char(20)	No	Usuario del medico
contrasena	char(20)	No	Contraseña del medico
nombre	char(20)	No	Nombre del medico
apellidospadre	char(50)	No	Primer Apellido del medico
apellidosmadre	char(50)	Si	Segundo Apellido del medico
correo	char(50)	No	Correo del medico
estatut	char(20)	Si	Estatuto del medico
direccion	char(50)	Si	Dirección del medico
codigo_postal	char(5)	Si	Código Postal
colonia	char(50)	Si	Colonia
municipio	char(30)	Si	Municipio
estado	char(20)	Si	Estado

Patient

Campo	Tipo	Nulo	Comentarios
id_patient	integer	No	id del paciente
id_oficina	Integer	No	Id de la oficina del paciente
fecha_naci	char(10)	No	Fecha de nacimiento
contraindicacion	char(50)	Si	Contraindicación
nombre	char(20)	No	Nombre del paciente
apellidospadre	char(50)	No	Primer Apellido del paciente
apellidosmadre	char(50)	No	Segundo Apellido del paciente
sexo	char(1)	No	Sexo del paciente

direccion	char(50)	Si	Dirección del paciente
codigo_postal	char(5)	Si	Código Postal
colonia	char(50)	Si	Colonia
municipio	char(30)	Si	Municipio
estado	char(20)	Si	Estado

Historics

Campo	Tipo	Nulo	Comentarios
id_historico	integer	No	id del histórico
id_medico	integer	No	id del medico
id_patient	integer	No	id del paciente
fecha	Char(10)	No	Fecha del histórico
Resume	char(200)	Si	Resumen de un diagnostico

SMS

Campo	Tipo	Nulo	Comentarios
id_sms	integer	No	id del SMS
id_patient	integer	No	id del paciente
mensaje	char(100)	Si	Mensaje del SMS
fecha	char(10)	No	Fecha de recepción
leido	char(3)	Si	Si el SMS ha sido leído o no

Oficina

Campo	Tipo	Nulo	Comentarios
id_oficina	Integer	No	Id
nombre	Char(50)	No	Nombre de la oficina
numeroSMS	Char(10)	No	Numero de envío de SMS
numeroLlamada	Char(10)	No	Numero de llamada
paramedico	Char(50)	Si	Nombre del paramédico

Espera

Campo	Tipo	Nulo	Comentarios
id_espera	integer	No	Id
id_oficina	integer	No	Id de la oficina en espera
espera	char(3)	Si	Si el paciente ha sido atendido o no

Tabla 7: Diccionario de datos

3.6. Resumen del Capitulo

Este capítulo abordó el contexto del sistema y definió los tres sub-sistemas. Se ha hecho también un análisis del modelo de los datos necesario para el funcionamiento del sistema.

En el siguiente capítulo se especificara los tres sub-sistemas, abordando con detalle los aspectos tecnológicos y de diseño de software involucrados.

Capítulo 4. Implementación del Sistema

En este capítulo, se analiza la implementación del sistema tomando en cuenta cada sub-sistemas: el sub-sistema del médico, del paciente y el sub-sistema de recuperación de los datos.

4.1. Sub-Sistema del lado del Médico

En esta sección, se analiza el sub-sistema del lado del médico de forma detallada.

Se analiza el contexto del sistema, la arquitectura del sistema, los diagramas de secuencia de varios casos de uso del sistema y por fin se describen las funcionalidades implementadas en el sistema.

4.1.1. Contexto del Sub-Sistema

Un médico accede al sistema a utilizando un navegador Web. Puede identificarse y acceder de esa manera a su página personal para utilizar el sistema. En una utilización habitual, el médico espera la próxima llamada a efectuar. La página se actualiza automáticamente. Una vez elegida una solicitud de atención, el médico llama a la oficina correspondiente para comunicarse con el paciente. Se informa al médico sobre el estado de salud del paciente y de sus problemas. El médico puede empezar el diagnostico del paciente. A partir de la información recibida del paciente, el médico puede solicitarle datos vitales con la ayuda del paramédico o del responsable de la oficina. Por ejemplo, tomar su temperatura o su tensión arterial.

El médico espera la recepción de los datos vía un SMS para emitir su diagnóstico. Una vez concluido el proceso, el médico puede guardar el diagnostico del paciente en su histórico.

Además de su utilización habitual, el sistema permite buscar los datos de los pacientes, administrar los datos de los pacientes y los médicos.

4.1.2. Arquitectura del Sub-Sistema.

La arquitectura del sub-sistema del lado del médico se compone de tres capas (ver Figura 21). La primera es la interfaz del médico con un navegador Web y un teléfono para llamar a las oficinas. La segunda capa es la parte que construye la interfaz grafica con un conjunto de Servlets Java y de páginas JSP. Por fin la tercer y última capa es un sistema de gestión de base de datos que nos va a permitir manejar nuestros datos.

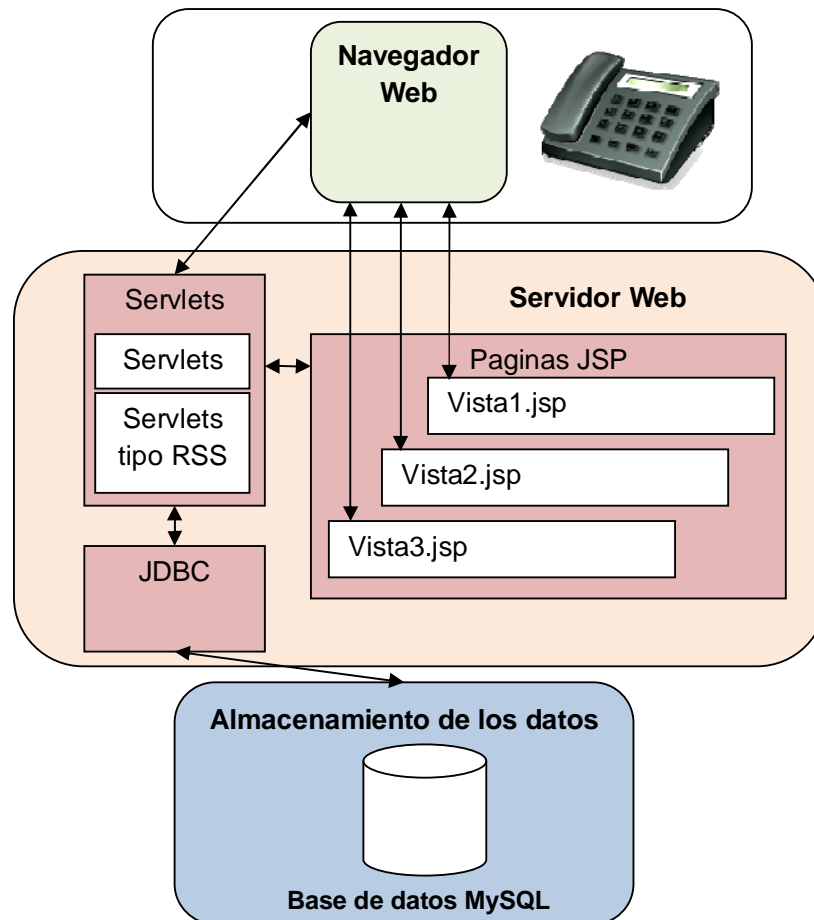


Figura 21: Arquitectura del sub-sistema del Médico

4.1.3. Los Servlets y las páginas JSP del Sub-Sistema

Esta parte proporciona información sobre los Servlets y las páginas JSP utilizados en el sistema. Vamos a ver su utilización en el sistema. Esa información será muy útil para entender la siguiente parte que se enfoca a los diagramas de secuencia que nos muestra la comunicación entre Servlets y páginas JSP.

4.1.3.1. Los Servlets del Sub-Sistema

Este sub-sistema utiliza un conjunto de Servlets. Cada uno tiene su propia utilidad. La siguiente tabla (Tabla 8) nos muestra los Servlets junto con su URL de acceso y su utilidad en el sub-sistema.

Servlet	URL	Utilidad
AccueilServlet	/accueil	Permite desplegar la página de login o página de inicio.
AnadirdocServlet	/anadirdoc	Permite desplegar la página de login la página para añadir un médico.
AnadirServlet	/anadir	Permite desplegar la página de login o la página para añadir un paciente.
BuscarPacienteServlet	/buscar-paciente	Permite desplegar la página de login o la página para buscar un paciente.
DiagnosticoServlet	/diagnostico	Permite desplegar la página de login, la página de inicio o la página de diagnóstico.
GuardarDatosDoctorServlet	/anadir_doctor	Permite desplegar la página de login o añadir un doctor en la base de datos e ir al Servlet MostrarDatosDoctor
GuardarDatosPacienteServlet	/anadir_paciente	Permite desplegar la página de login o añadir un paciente en la base de datos e ir al Servlet MostrarDatosPaciente
Modif_OficinaServlet	/modif_oficina	Permite modificar los datos de una oficina.
ModificarDatosDoctorServlet	/modificar_doctor	Permite desplegar la página de login o modificar los datos de un médico en la base de datos e ir al Servlet MostrarDatosDoctor
ModificarDatosPacienteServlet	/modificar	Permite desplegar la página de login o modificar los datos de un paciente en la base de datos e ir al Servlet MostrarDatosPaciente
MostrarDatosDoctorServlet	/mostrar_doctor	Permite desplegar la página de login o la página de modificación de un médico
MostrarDatosPacienteServlet	/mostrar	Permite desplegar la página de login o la página de modificación de un paciente
New_OficinaServlet	/new_oficina	Permite crear una oficina.
rssDoctor	/rssDoctor	Permite crear un archivo XML con la información de varios médicos
rssNumero	/rssNumero	Permite crear un archivo XML con las últimas llamadas
rssOficina	/rssOficina	Permite crear un archivo XML con la información de varias oficinas.
rssPaciente	/rssPaciente	Permite crear un archivo XML con la información de varios pacientes
rssSms	/rssSms	Permite crear un archivo XML con los últimos SMS no leído de un paciente
SauvegardeServlet	/sauvegarde	Permite guardar el histórico de un diagnóstico de un paciente
SessionServlet	/mexmed	Permite desplegar la página de login o ir al Servlet AccueilServlet

ShowOficinasServlet	/oficinas	Permite desplegar la página de login o la pagina de administración de las oficinas.
ShowSupprDocServlet	/suppr-doc	Permite desplegar la página de login o la página para suprimir un médico
ShowSupprPacienteServlet	/suppr	Permite desplegar la página de login o la página para suprimir un paciente
SuprimirDoctorServlet	/suprimirDoctor	Permite suprimir un médico a partir de su id
SuprimirNumeroServlet	/suprimir	Permite poner una llamada como atendida.
SuprimirOficinaServlet	/suprimirOfic	Permite suprimir una oficina a partir de su id.
SuprimirPacienteServlet	/suprimirPaciente	Permite suprimir un paciente a partir de su id
SuprimirSmsServlet	/suprimirSms	Permite guardar el mensaje de un SMS y suprimir el SMS a partir de su id

Tabla 8: URL y Utilidades de los Servlets del sub-sistema

4.1.3.2. Las páginas JSP del Sub-Sistema

El sub-sistema utiliza un conjunto de páginas JSP. Cada una tiene su propia utilidad. La siguiente tabla (Tabla 9) nos muestra las páginas JSP con su utilidad para el sub-sistema.

Paginas JSP	Utilidad
accueil.jsp	Es la página de inicio del sistema, una vez conectado.
anadir.jsp	Es la página para añadir un paciente
anadir-doc.jsp	Es la página para añadir un médico
buscar_paciente.jsp	Es la página para buscar un paciente si no se conoce su id
diagnostico.jsp	Es la página para diagnosticar un paciente
envoieEmail.jsp	Es la página para enviar un correo al administrador en caso de que no se acuerda de su contraseña.
erreur.jsp	Es la página que muestra los errores de ingreso de información.
modificar_datos.jsp	Es la página que permite modificar los datos de un paciente
modificar_datos_doctor.jsp	Es la página que permite modificar los datos de un médico
oficinas.jsp	Es la página que permite administrar las oficinas.
session.jsp	Es la página de login al sistema
suprimir.jsp	Es la página que permite suprimir un paciente
suprimir-doctor.jsp	Es la página que permite suprimir un médico

Tabla 9: Utilidad de las paginas JSP del sub-sistema

4.1.4. Desarrollo con páginas JSP y Servlets.

Para ejecutar los Servlets, se necesita un contenedor de Servlets por ejemplo Tomcat 6. En la carpeta de Tomcat 6.0, se puede encontrar varios elementos. La Figura 22 nos muestra la arborescencia de Tomcat, una aplicación Web se tiene que poner en la carpeta webapps.

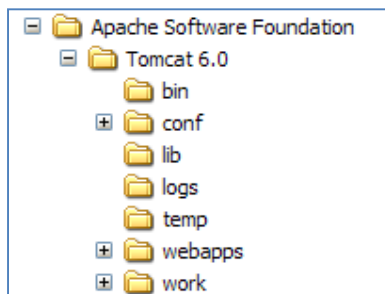


Figura 22: Arborescencia de Tomcat

Para desplegar una aplicación Web con Tomcat 6, se necesita seguir una arborescencia y poner los elementos en la carpeta correspondiente. La Figura 23 nos muestra la arborescencia de la carpeta Tesis que contiene toda la parte del médico. Los .class de los Servlets tienen que estar en WEB-INF/classes/servletsTesis. Los .java de los Servlets tienen que estar en WEB-INF/src/servletsTesis.

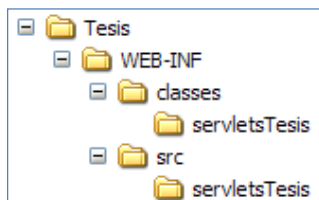


Figura 23: Arborescencia de una aplicación web

Antes de arrancar Tomcat, se necesita compilar los Servlets y poner los .class en su lugar. El archivo web.xml permite hacer el mapeo entre los URL y los Servlets (es decir, los archivos .class); se ubica en WEB-INF y se compone de varios elementos. La Figura 24 nos muestra un ejemplo de archivo web.xml que se compone de un conjunto de etiquetas servlet para caracterizar cada Servlet y de etiquetas servlet mapping para mapear cada Servlet a un URL de acceso.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  version="2.5">

  <servlet>
    <servlet-name>nombreServlet</servlet-name>
    <servlet-class>paquete.nombredelServlet</servlet-class>
    <init-param>
      <param-name>urlAccueil</param-name>
      <param-value>/accueil.jsp</param-value>
    </init-param>
    <init-param>
      ...
    </init-param>
  </servlet>

  <servlet-mapping>
    <servlet-name>nombreServlet</servlet-name>
    <url-pattern>/urlServlet</url-pattern>
  </servlet-mapping>
</web-app>

```

Figura 24: Ejemplo de fichero web.xml

Este archivo es muy importante porque nos permite definir los URL de la aplicación. Por ejemplo con el Servlet que tenemos en la Figura 24, el URL del Servlet sería en `http://localhost:8080/Tesis/urlServlet`.

4.1.4. Diagramas de Secuencia

Esta parte nos permite detallar la comunicación entre varios Servlets y varias páginas JSP para implementar un caso de uso del médico. Se describe el acceso de un médico al sistema, un caso de diagnóstico de un médico para un paciente, la búsqueda de un paciente y por fin la administración de los pacientes o de los médicos con su creación, su modificación y su supresión.

4.1.4.1. Acceso a usuario

La Figura 25 nos permite entender las interacciones entre varios Servlets y páginas JSP para permitir o no el acceso a un usuario. El acceso de un médico se hace gracias al Servlet `SessionServlet` que al inicio despliega la página `session.jsp`. A partir de la información ingresada en la página `session.jsp`, el Servlet `SessionServlet` verifica si la información es válida y si existe un usuario compatible con esa información. Si existe el usuario, se despliega el Servlet `AccueilServlet` que nos va permitir mostrar la página de inicio del sistema.

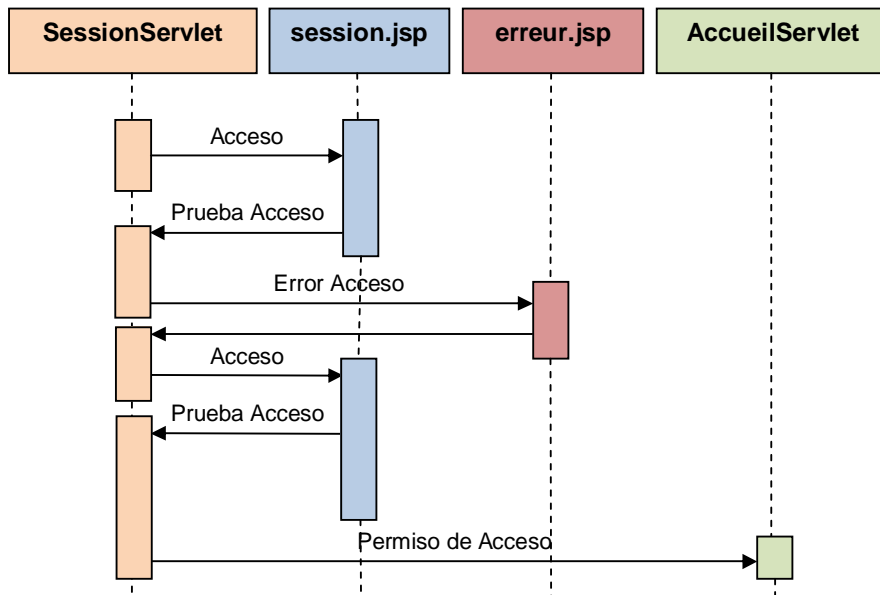


Figura 25: Diagrama de secuencia - Acceso a usuario

4.1.4.2. Creación de pacientes

La Figura 26 nos permite entender las interacciones entre varios Servlets y páginas JSP para permitir o no la creación de un nuevo paciente en la base de datos. Añadir un paciente se hace gracias al Servlet AnadirServlet que permite desplegar la página JSP anadir.jsp. A partir de la información ingresada por el usuario, el Servlet va a verificar si la información es válida. Si no lo es va a desplegar una página de fallo, la pagina JSP erreur.jsp. En caso de que la información sea válida, va a añadir los datos del paciente a la base de datos y dirigirse al Servlet MostrarDatosPacienteServlet que nos va a permitir mostrar la información del nuevo paciente y así tener la posibilidad de modificarlo en caso de error en la información guardada.

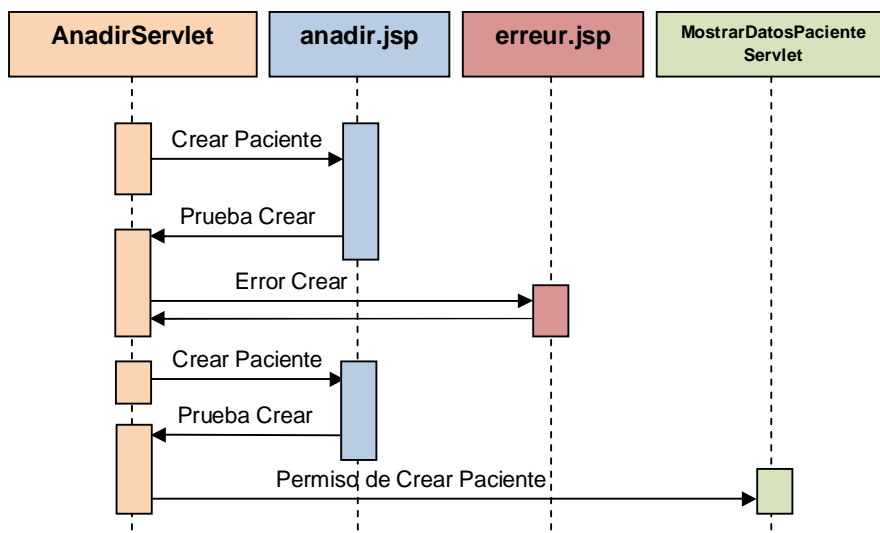


Figura 26: Diagrama de secuencia – Creación de paciente

4.1.4.3. Modificación de paciente

La Figura 27 nos permite entender las interacciones entre varios Servlets y páginas JSP para modificar los datos de un paciente. Modificar un paciente se hace gracias al Servlet `MostrarDatosPacienteServlet` que permite desplegar la página JSP `Modificar_datos.jsp` con la información del paciente. A partir de la información ingresada por el usuario, el Servlet `ModificarDatosPacienteServlet` va a modificar la información en la base de datos. El Servlet `MostrarDatosPacienteServlet` nos permitirá finalmente desplegar de nuevo la información del paciente pero con la modificación realizada.

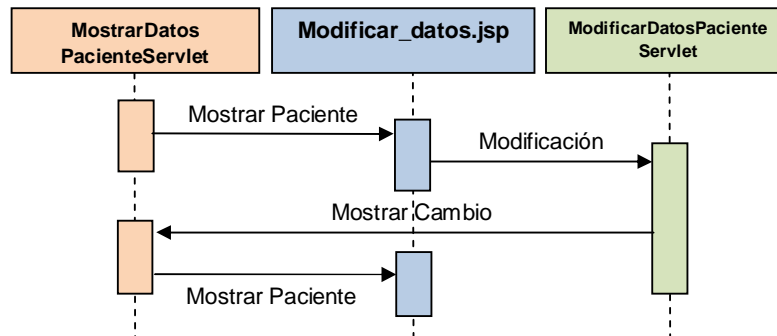


Figura 27: Diagrama de secuencia – Modificación de paciente

4.1.4.4. Supresión de paciente

La Figura 28 nos permite entender las interacciones entre varios Servlets y páginas JSP para suprimir un paciente de la base de datos. Suprimir un paciente se hace gracias al Servlet `ShowSupprPacienteServlet` que permite desplegar la página JSP `suprimir.jsp`. Al momento de cargar de la página, se hace una llamada asíncrona al servidor para buscar los pacientes en la base de datos mediante el Servlet `rssPaciente`. Se puede hacer una búsqueda sobre los datos del paciente y cada vez que se ingresa una letra se hace la misma llamada asíncrona al Servlet `rssPaciente` para buscar los datos de pacientes de la base de datos que son coherentes con la búsqueda. Una vez que se han encontrado los datos del paciente se puede suprimirlo gracias al Servlet `SuprimirPacienteServlet`. Para mostrar el cambio se llama de nuevo al Servlet `ShowSupprPacienteServlet`.

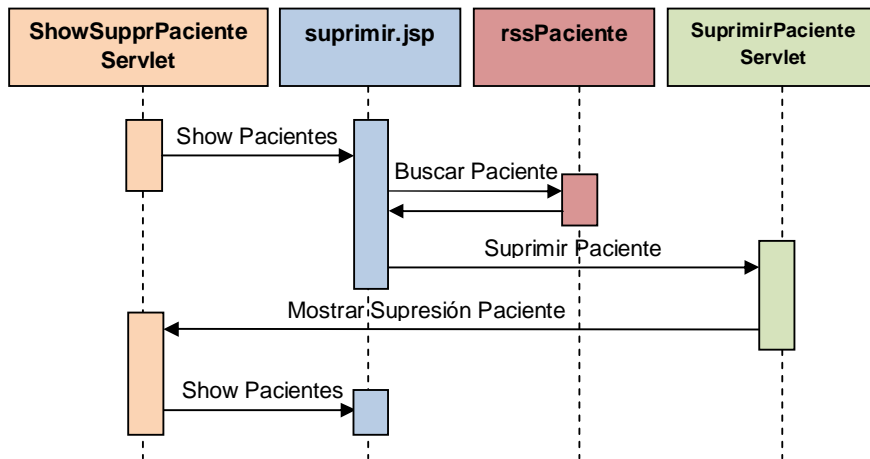


Figura 28: Diagrama de Secuencia – Supresión de paciente

4.1.4.5. Creación de Médico

La Figura 29 nos permite entender las interacciones entre varios Servlets y páginas JSP para permitir o no la creación de un nuevo médico en la base de datos. Añadir los datos de un médico se hace utilizando el Servlet AnadirDocServlet que permite desplegar la página JSP anadir-doc.jsp. A partir de la información ingresada por el usuario, el Servlet va a verificar si la información es válida. Si no lo es va a desplegar una página de fallo, la página JSP erreur.jsp. En caso de que la información es válida, va a añadir los datos del médico en la base de datos y dirigirse al Servlet MostrarDatosDoctorServlet que nos va a permitir mostrar la información del nuevo médico y así tener la posibilidad de modificarlo en caso de error en la información guardada.

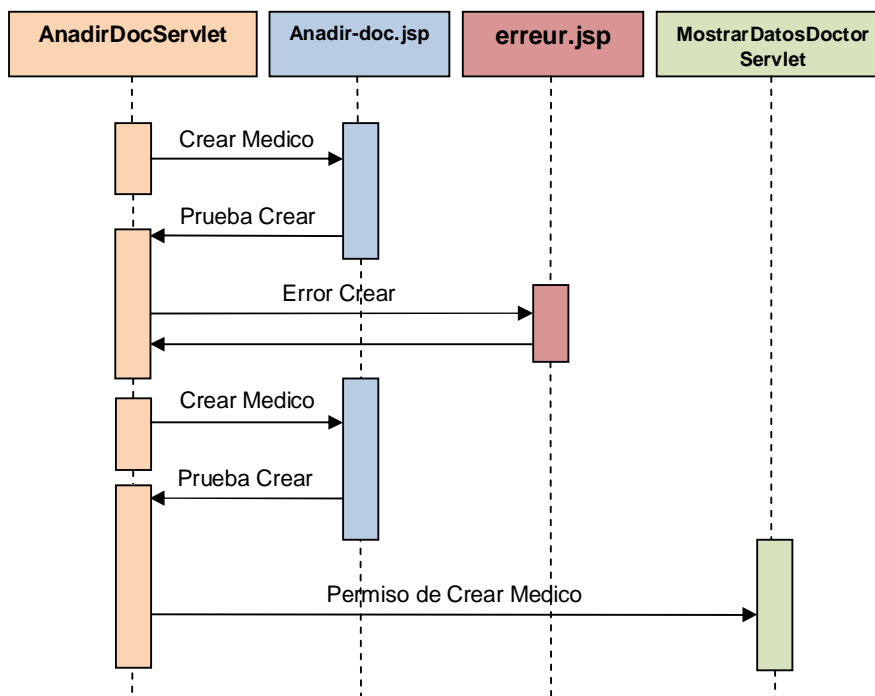


Figura 29: Diagrama de Secuencia – Creación de Medico

4.1.4.6. Modificación de Médico

La Figura 30 nos permite entender las interacciones entre varios Servlets y páginas JSP para modificar los datos de un médico. Modificar los datos de un médico se hace gracias al Servlet `MostrarDatosDoctorServlet` que permite desplegar la página JSP `modificar_datos_doctor.jsp` con la información del médico. A partir de la información ingresada por el usuario, el Servlet `ModificarDatosDoctorServlet` va a modificar la información en la base de datos. El Servlet `MostrarDatosDoctorServlet` nos permitirá finalmente desplegar de nuevo la información del médico pero con la modificación realizada.

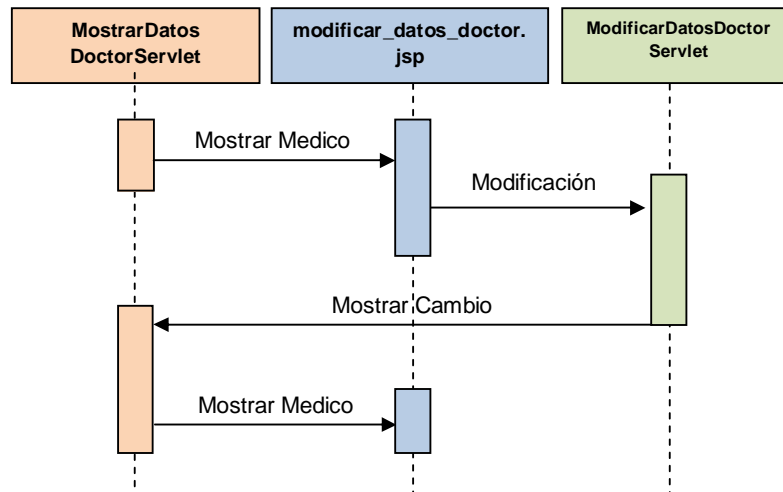


Figura 30: Diagrama de Secuencia – Modificación de Médico

4.1.4.7. Supresión de Médico

La Figura 31 nos permite entender las interacciones entre varios Servlets y páginas JSP para suprimir los datos de un médico de la base de datos. Suprimir un médico se hace al inicio gracias al Servlet `ShowSupprDocServlet` que permite desplegar la página JSP `suprimir-doctor.jsp`. Al momento de cargar la página, se hace una llamada asíncrona al servidor para buscar los datos de los médicos en la base de datos gracias al Servlet `rssDoctor`. Se puede hacer una búsqueda sobre el médico y cada vez que se ingresa una letra se hace la misma llamada asíncrona al Servlet `rssDoctor` para buscar los datos de médicos de la base de datos que son coherentes con la búsqueda. Una vez que se han encontrado los datos del médico buscado se puede suprimir gracias al Servlet `SuprimirDoctorServlet`. Para mostrar el cambio se llama de nuevo el Servlet `ShowSupprDocServlet`.

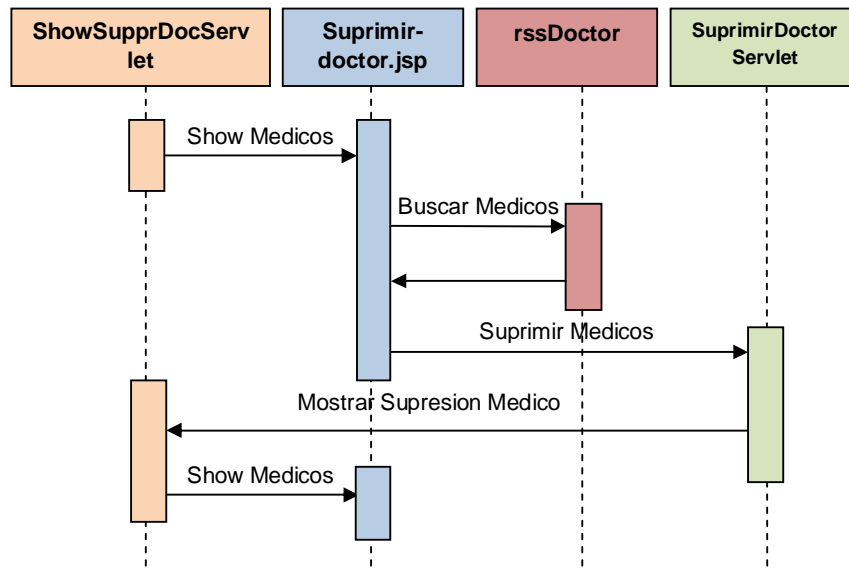


Figura 31: Diagrama de Secuencia – Supresión de Médico

4.1.4.8. Diagnosticar Paciente

La Figura 32 nos permite entender las interacciones entre varios Servlets y páginas JSP para diagnosticar un paciente. Diagnosticar un paciente se hace al inicio gracias al Servlet `AccueilServlet` que permite desplegar la página JSP `accueil.jsp` que es la página de inicio del sistema. A la carga de la página, se hace una llamada asíncrona al servidor para buscar las oficinas en espera y así conocer el próximo número telefónico a llamar. Es una llamada asíncrona al Servlet `rssNumero`. Una vez que se ha encontrado el próximo número telefónico a llamar se borra de la base de datos gracias al Servlet `SuprimirNumeroServlet`. Se puede a partir de este momento realmente empezar el diagnóstico llamando el Servlet `DiagnosticoServlet` gracias al número de paciente recibido del paciente en comunicación. El Servlet `DiagnosticoServlet` nos permite desplegar la página JSP `diagnostico.jsp` con la información del paciente. A la carga de la página, se hace una llamada asíncrona al servidor para buscar los SMS del paciente todavía no leídos. Posiblemente al inicio no hubiera ningún SMS a leer y se tendría que esperar el SMS que va enviar el paciente con sus datos. Una vez que tenemos el SMS a leer se puede suprimir de la base de datos gracias al Servlet `SuprimirSmsServlet` y añadir la información del SMS al diagnóstico. Si el médico tiene suficiente información para completar el diagnóstico de la enfermedad del paciente puede guardar los datos del diagnóstico en la base de datos utilizando el Servlet `SauvegardeServlet`. Por fin se llama de nuevo el Servlet `DiagnosticoServlet` para mostrar el cambio. Para elegir el próximo número telefónico a llamar se tiene que regresar al inicio con el Servlet `AccueilServlet`.

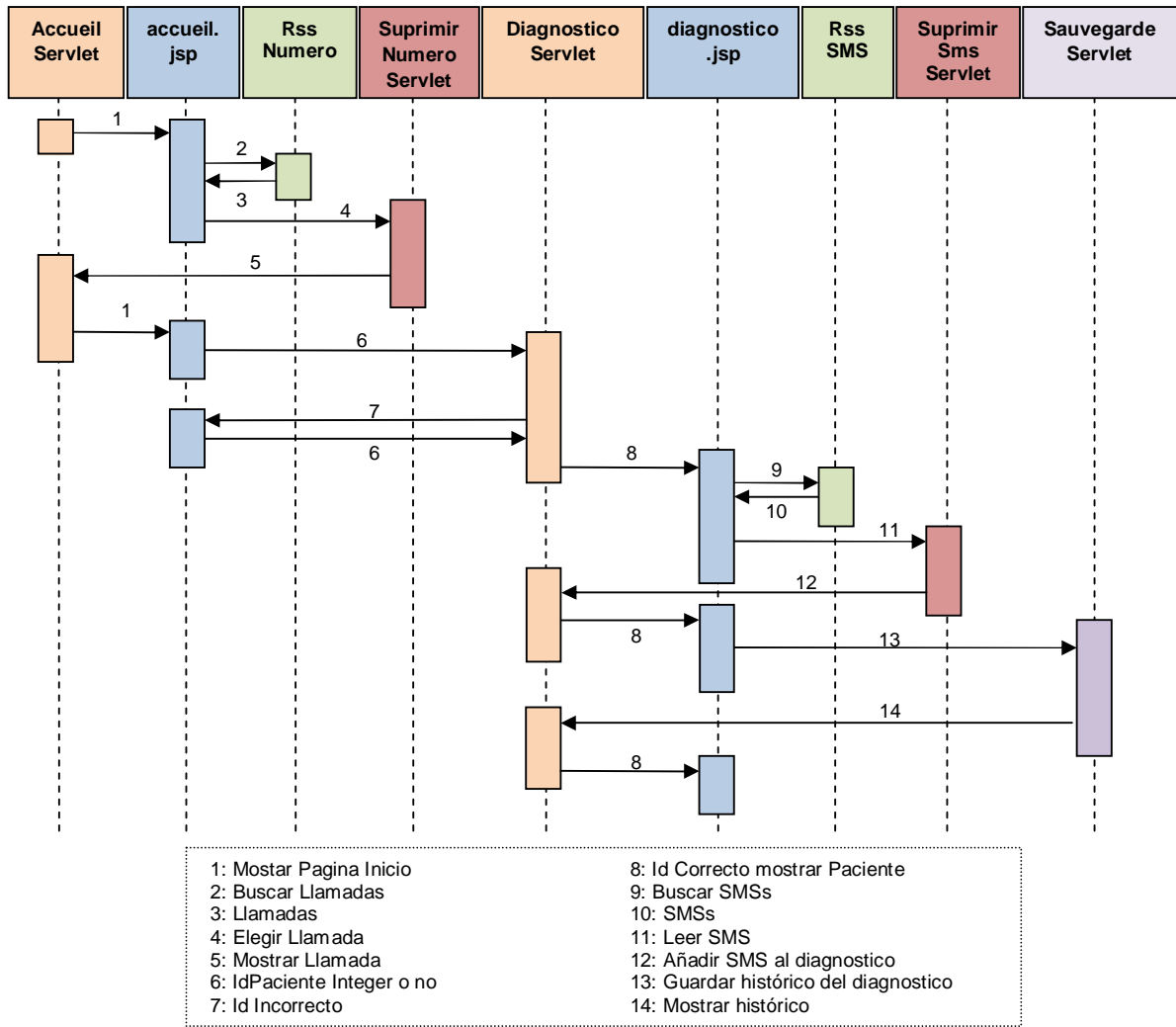


Figura 32: Diagrama de Secuencia – Diagnosticar un Paciente

4.1.6. Funcionalidades implementadas en el Sub-Sistema

En esta sección se explica el funcionamiento del sub-sistema resumiendo en acciones concretas de utilización e ilustrando con figuras como es posible operarlo para diagnosticar cualquier paciente.

4.1.6.1. Identificación del médico

En la Figura 33, se puede observar que el acceso al sistema de Médico a distancia se da mediante la introducción de una URL al navegador Web. Se despliega la pantalla de login, donde es necesario identificarse como un usuario valido en el sistema.

Un medico ingresa su Id de usuario y su clave de acceso en la página de login. Si uno de los dos campos es falso, se despliega una página de información del fallo (Figura 34). A partir de la información, puede regresar a la página de login para ingresar de nuevo la información y tratar de conectarse al sistema.

En caso de que no se acuerda de su Id de usuario o de su clave (Figura 35), un médico puede enviar un email al administrador del sistema para recibir, en su correo de respalda, la información.

Si el médico ingresa su usuario y su contraseña, puede entrar en el sistema, con lo cual se desplegara la página de inicio del sistema.



Figura 33: Página de Inicio



Figura 34: Página de fallo al inicio

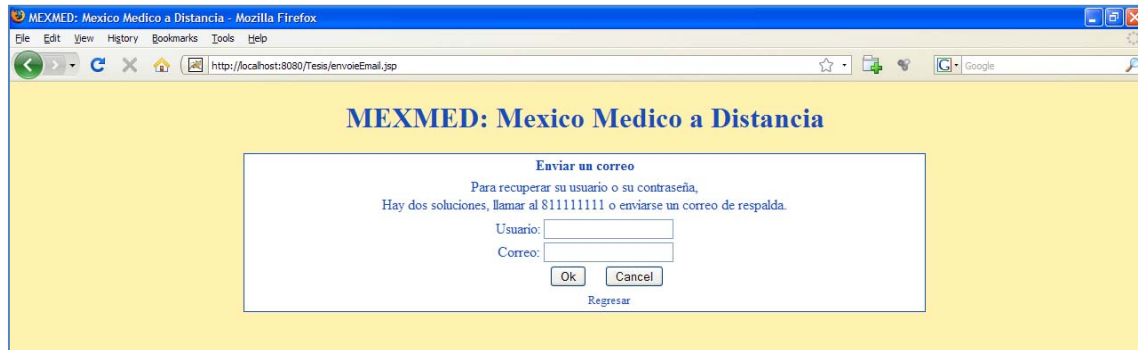


Figura 35: Si se olvide su contraseña, se puede enviar un correo de respalda.

4.1.6.2. Esperar una llamada

Una vez conectado al sistema, se despliegue la página de inicio identificando al médico. En la parte superior izquierda, se encuentra el nombre del médico. Abajo se encuentra un menú con las diferentes acciones que puede hacer el médico, hacer un diagnóstico de paciente, buscar un paciente, administrar los usuarios o desconectarse del sistema.

Al inicio, si no hay nadie a quien atender, se tiene que esperar una llamada de un paciente (Figura 36). En caso de que haya varias llamadas en espera se puede elegir la siguiente. De manera automática, la página se actualiza con las nuevas llamadas (Figura 37). Utilizando una llamada asíncrona al servidor gracias a javascript y al objeto XMLHttpRequest. Cuando se elige una llamada, se pone la llamada como atendida en la base de datos, se guarda el número a llamar en la sesión y a continuación de "Diagnostico-Llamar" en el título de la página (Figura 38).

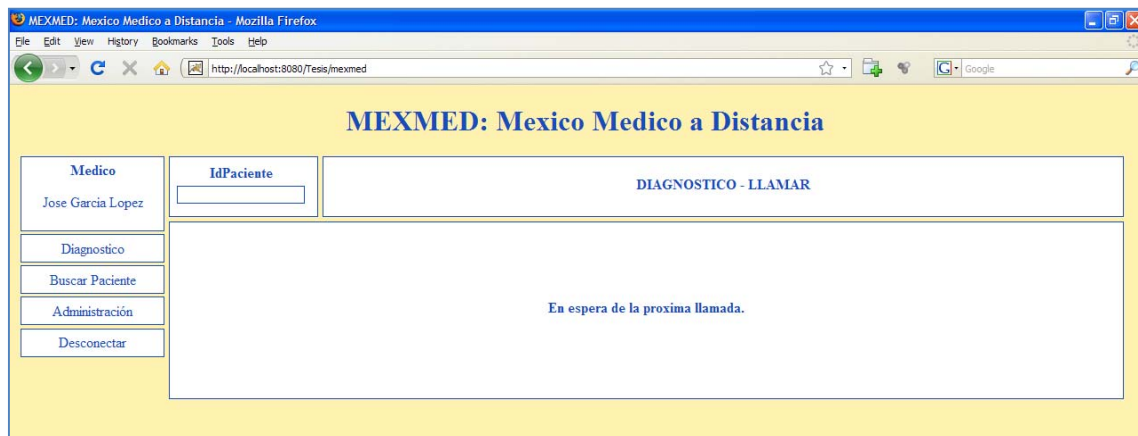


Figura 36: En espera de llamada

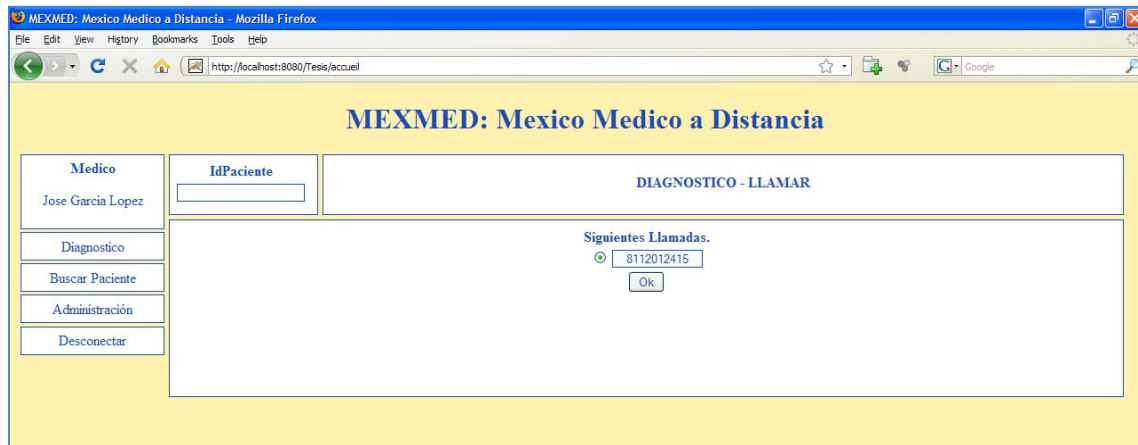


Figura 37: Elegir una llamada



Figura 38: Llamada elegida

4.1.6.3. Diagnosticar un paciente

El diagnóstico de un paciente se hace de la siguiente manera. El paciente ha hecho una llamada para inscribirse en el sistema. El médico elige el número de paciente en la lista de llamadas en espera y llama al paciente.

El médico pregunta el id de paciente para recuperar su información y seguir los pasos de diagnóstico (Figura 39). A partir de ese momento, el médico pregunta los síntomas del paciente. Puede ingresarlos en la parte “diagnóstico” de la pantalla (Figura 40). Para tener mayor información, el médico pregunta al paciente hacer una toma de datos como temperatura o tensión arterial.

El paciente va hacer los diferentes exámenes que ha solicitado el médico con la ayuda del paramédico que está con él. Los datos son enviados vía un SMS al servidor (Figura 41). El médico espera el SMS para concluir su diagnóstico; ello se hace mediante una llamada asíncrona al servidor gracias a Javascript y al objeto XMLHttpRequest. Una vez que se ha leído el SMS, la información del SMS se agrega al resto del diagnóstico (Figura 42). Para

seguir la información de los pacientes, hay un histórico de los diagnósticos de cada paciente (Figura 43).

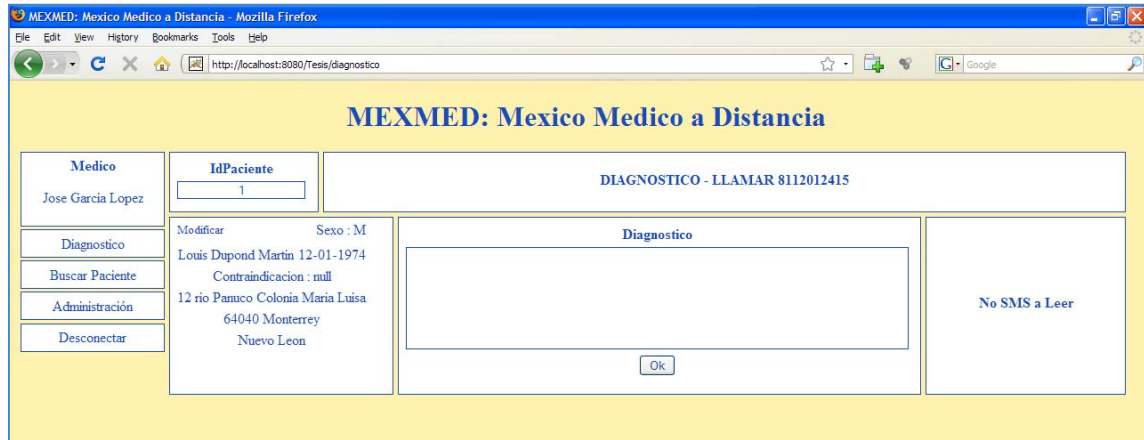


Figura 39: Diagnóstico paciente



Figura 40: En espera de SMS, empezando el diagnóstico

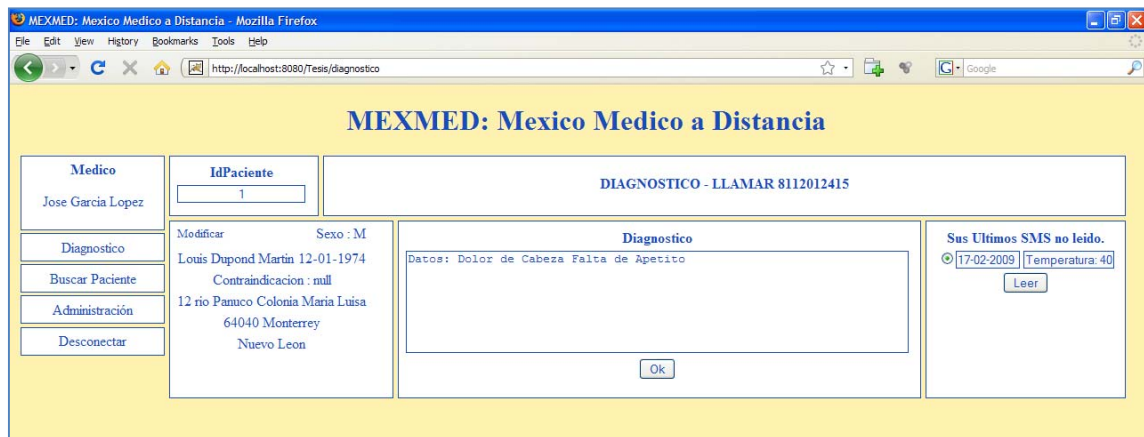


Figura 41: SMS recibido

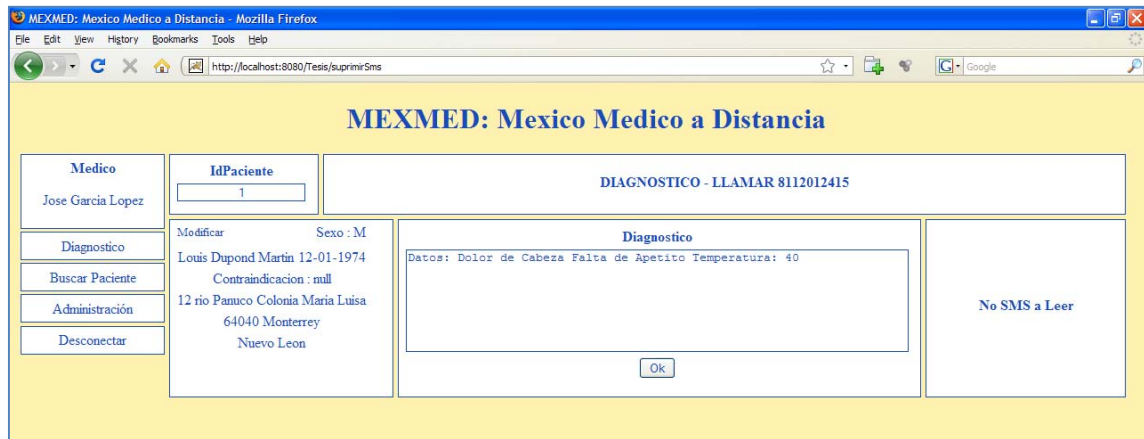


Figura 42: SMS leído y añadido al diagnóstico

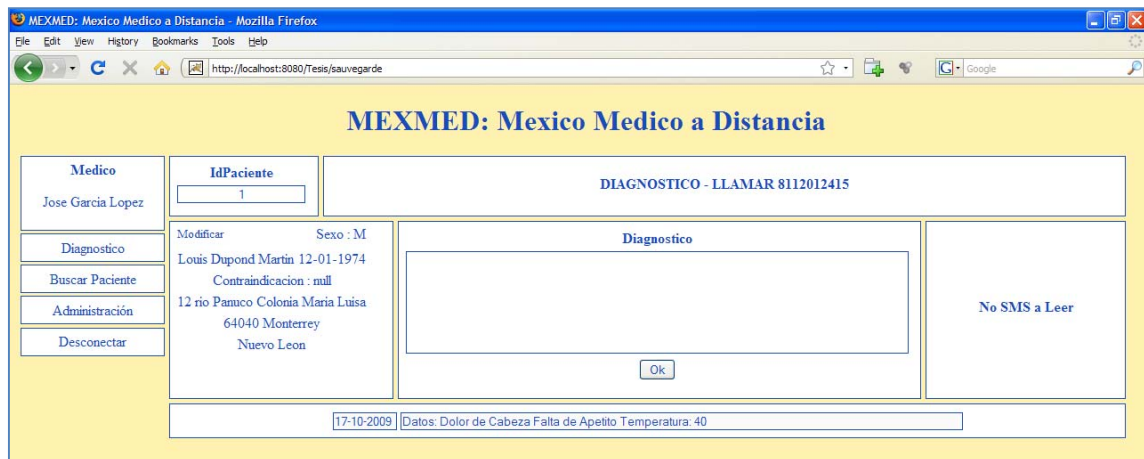


Figura 43: Diagnóstico guardado

4.1.6.4. Buscar un paciente

Al inicio del diagnóstico de un paciente, puede dar el caso de que el paciente no se acuerda de su Id. Se puede buscar los datos del paciente en la base de datos gracias a la opción Buscar Paciente en el menú.

En la parte superior se despliega los tres elementos de búsqueda que tiene para un paciente: nombre, primer apellido y segundo apellido. Al inicio se despliega en la parte inferior los datos de todos los pacientes (Figura 44). Agregando letras al nombre o a los apellidos, se puede afinar la búsqueda hasta encontrarlo (Figura 45). Utilizando una llamada asíncrona al servidor gracias a Javascript y al objeto XMLHttpRequest, el resultado de la búsqueda se ve automáticamente.

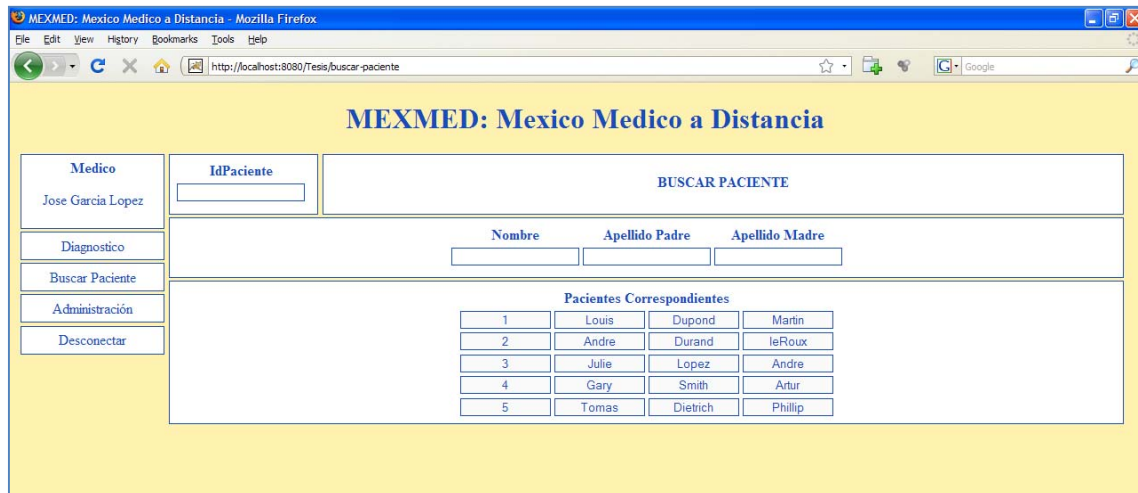


Figura 44: Búsqueda de un paciente



Figura 45: Búsqueda afinada de un paciente

4.1.6.5. Crear un paciente

En el caso de que tengamos un nuevo paciente, se puede añadir los datos del paciente en la base de datos seleccionando Administración en el menú. Al seleccionar, se despliega la página de creación de un paciente.

En esa página (Figura 47), los inputs llegan vacío y se necesita ingresar la información. Si al seleccionar el botón Ok, hay errores en los campos, se despliega la página de errores (Figura 48).

A partir de la información de los errores, se puede regresar a la página, para ingresar de nuevo los campos y tratar de crear el paciente. Una vez que el paciente es creado (Figura 50), se despliega la página de modificación en caso de errores no bloqueante para la creación que se tiene que modificar. Se muestra también el número del nuevo paciente, lo cual es importante informar para una llamada ulterior.

Vamos a necesitar crear también un paciente, en caso de que el paciente nos da un número desconocido en el sistema. Puede llegar ese caso, si el paciente ha sido suprimido del

sistema de manera errónea. En este caso, al ingresar el id del paciente, va a recibirse la pagina de diagnóstico con el sitio de la información del paciente una frase del tipo “Id Paciente desconocido Añadir Paciente” (Figura 46), haciendo clic sobre “Añadir Paciente” se puede acceder a la misma página que seleccionando “Administración” en el menú.

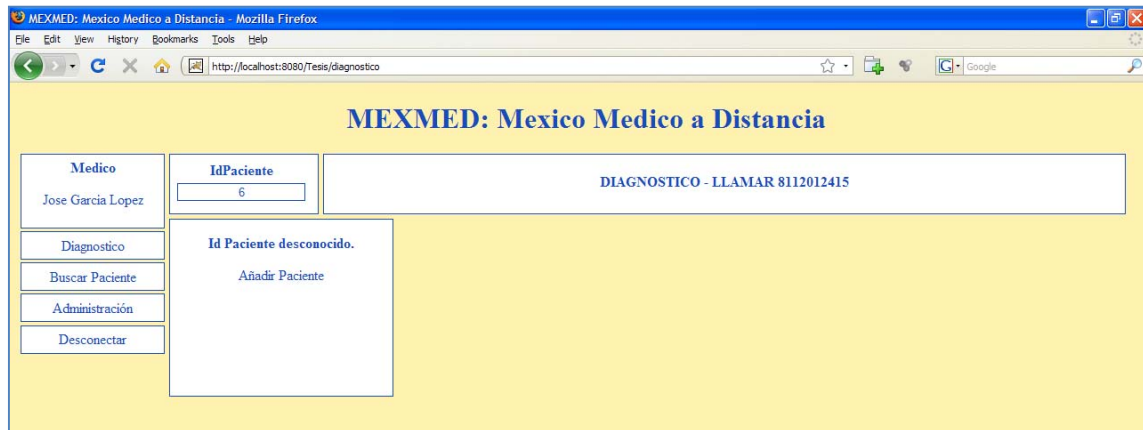


Figura 46: IdPaciente desconocido

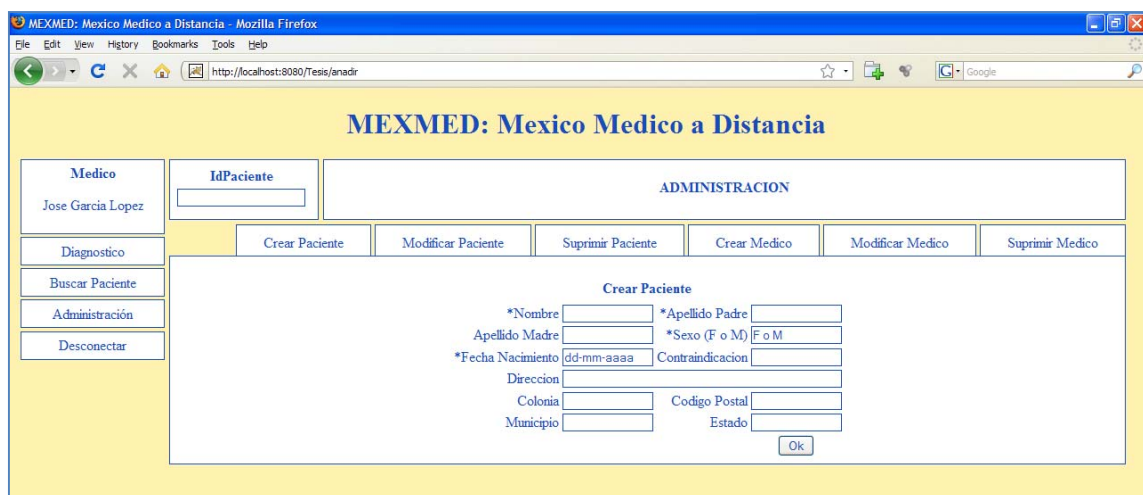


Figura 47: Administración – Crear Paciente

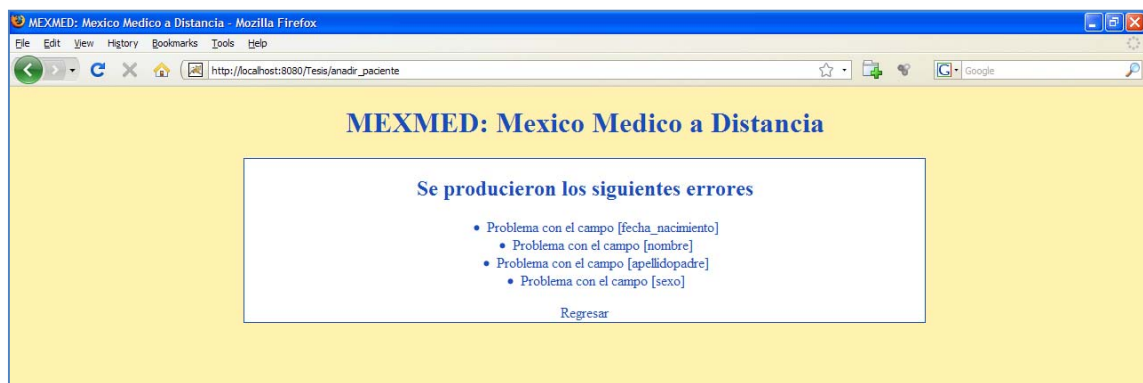


Figura 48: Administración – Pagina de fallo de creación de un paciente



Figura 49: Administración – Ingresando Datos del Paciente



Figura 50: Administración – Paciente Creado

4.1.6.6. Modificar los datos del paciente.

Al momento de diagnosticar un paciente, se puede verificar la información del paciente (Figura 51). En caso de que haya cambios que hacer, se puede modificar la información haciendo clic sobre la palabra “Modificar” del perfil del paciente.

Se desplegará la página de modificación (Figura 52), donde podemos ver los diferentes campos del paciente. Al seleccionar el botón Ok, se modificará la información del paciente. Haciendo clic sobre la palabra “Regresar”, se desplegará la página de diagnóstico con la información del paciente modificado.

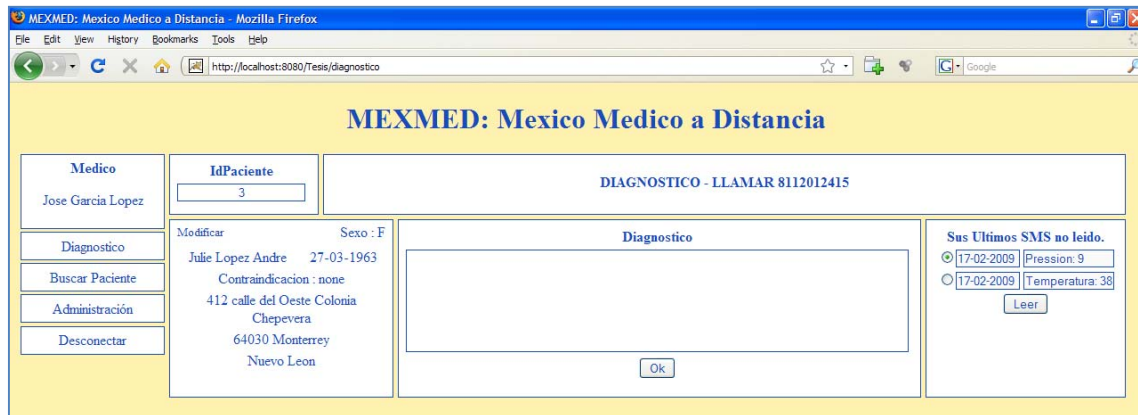


Figura 51: Administración – Verificar datos paciente



Figura 52: Administración – Página de modificación de los datos del paciente

4.1.6.7. Suprimir Paciente

Si un paciente ha decidido de dejar de utilizar el sistema, se puede suprimir su cuenta. No es una obligación pero es una posibilidad. Para acceder a la página de supresión, se tiene que seleccionar “Administración” en el menú general y “Suprimir Paciente” en el menú de administración.

Seleccionando eso se va a desplegar todos los pacientes (Figura 53). Para encontrar el paciente a suprimir, se puede hacer una búsqueda y afinarla sobre su Id, su nombre, su primer apellido y su segundo apellido (Figura 54). Utilizando una llamada asíncrona al servidor gracias a javascript y al objeto XMLHttpRequest, el resultado de la búsqueda se ve automáticamente.

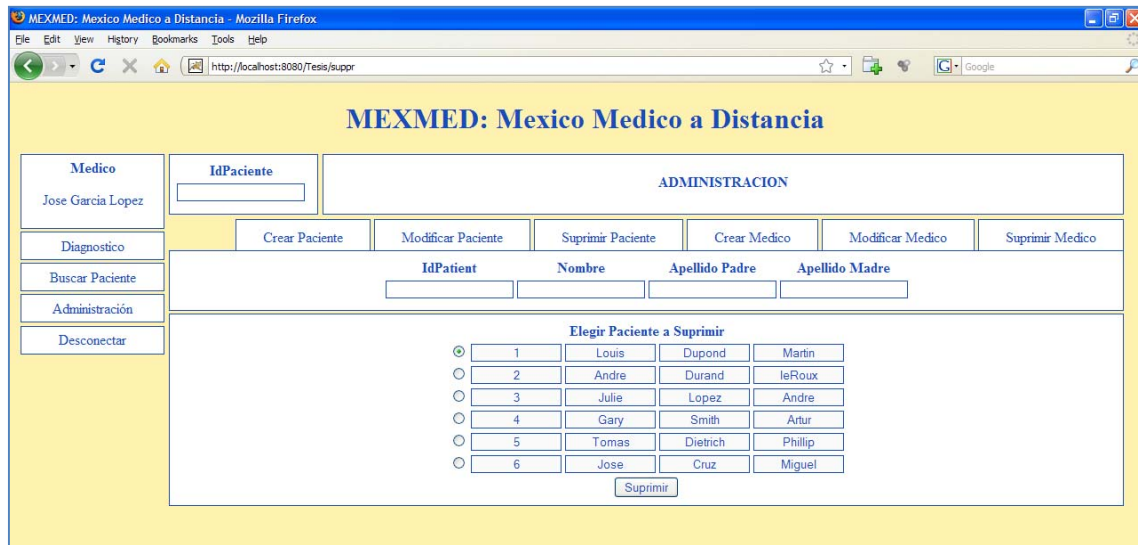


Figura 53: Administración – Suprimir Paciente

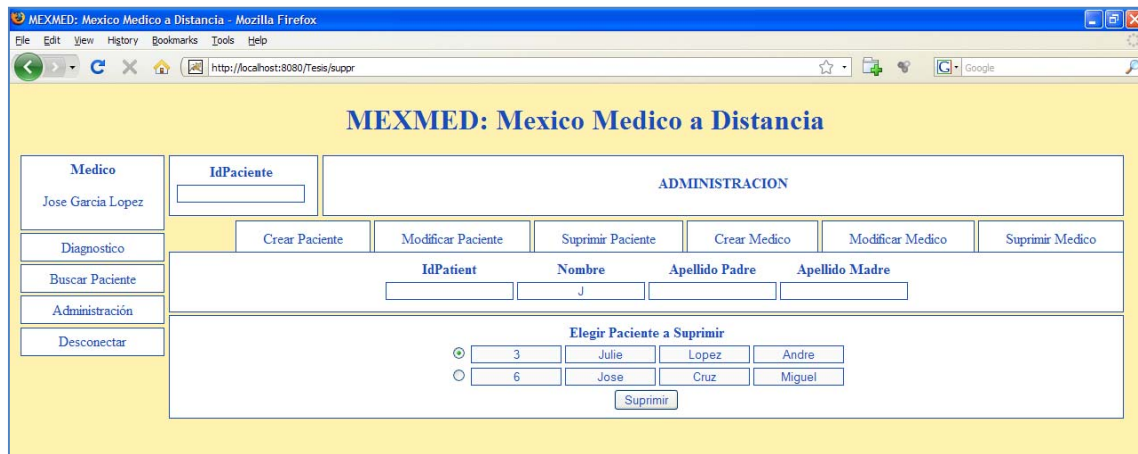


Figura 54: Administración - Búsqueda afinada del paciente a suprimir

4.1.6.8. Creación de un médico

En el sistema, cada médico tiene su propia cuenta para tener un histórico del médico. Así que cuando hay un nuevo médico se tiene que crear su cuenta en la base de datos.

Se puede acceder a esa página, seleccionando “Administración” en el menú general y “Crear médico” en el menú de administración. Se desplegará la página de creación de un médico (Figura 55).

Los inputs del formulario están vacío al inicio y se tiene que ingresar la información. Si al momento de crear el médico, hay errores en el formulario, se despliega la página de errores que nos da la información de los fallos (Figura 56). Al regresar, podemos tratar de nuevo de crear la cuenta del nuevo médico.

Una vez el médico creado (Figura 58), se despliega la página de modificación en caso de que hay errores no bloqueante, la creación que se tiene que modificar. Se tiene que acordar del id de usuario y de la contraseña para que el médico pueda conectarse al sistema con su cuenta.

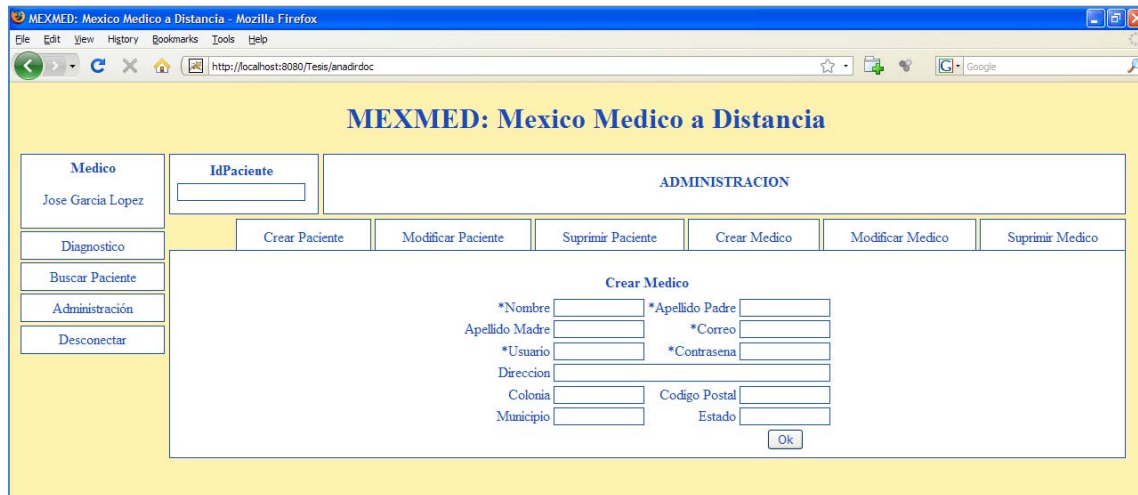


Figura 55: Administración - Creación de un médico



Figura 56: Administración - Pagina de fallo de creación de un médico



Figura 57: Administración - Ingresando los datos del médico a crear



Figura 58: Administración - Médico creado

4.1.6.9. Modificar datos de un médico

En caso de que un médico tiene que cambiar su información, se puede acceder a la página de modificación de dos formas. La primera es seleccionando "Administración" en el menú general y seleccionando "Modificar Médico" en el menú de Administración. La otra manera es más rápida, se necesita nada más seleccionar el nombre del médico en su perfil en la parte superior izquierda de la página (Figura 59).

Se desplegará la página de modificación (Figura 60), donde podemos ver los diferentes campos del médico. Al seleccionar el botón Ok, se modificará la información del médico. Haciendo clic sobre la palabra Regresar, se desplegará la página de inicio.

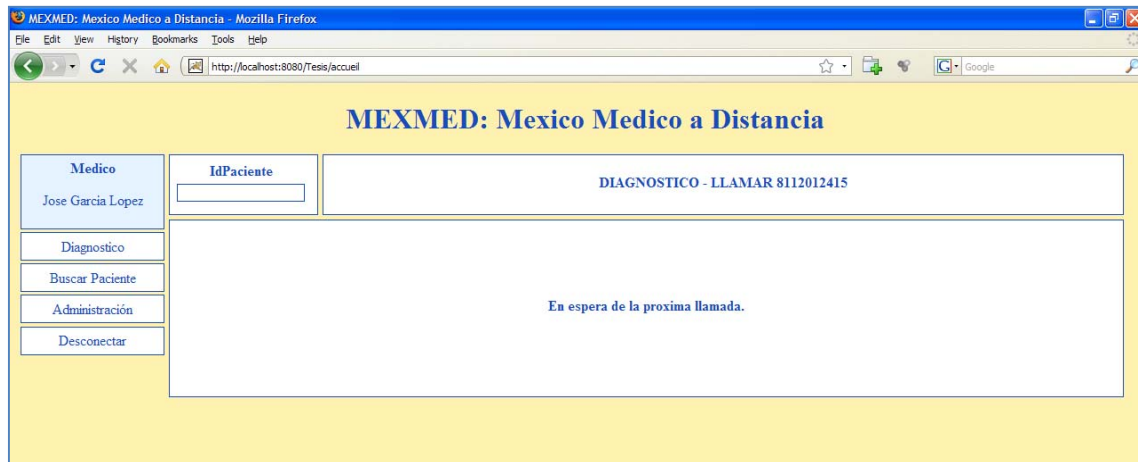


Figura 59: Administración – Acceso para modificar los datos del médico



Figura 60: Administración – Modificar los datos del médico

4.1.6.10. Supresión de un médico

En caso de que un médico deja el sistema y para no dejar abiertas puertas al sistema, se puede suprimir la cuenta del médico. No es una obligación pero es una posibilidad. Para acceder a la página de supresión, se tiene que seleccionar Administración en el menú general y Suprimir Médico en el menú de administración.

Seleccionando eso se va a desplegar los datos de todos los médicos (Figura 61). Para encontrar el médico a suprimir, se puede hacer una búsqueda y afinarla sobre su nombre, su primer apellido y su segundo apellido (Figura 62). Utilizando una llamada asíncrona al servidor gracias a Javascript y al objeto XMLHttpRequest, el resultado de la búsqueda se muestra automáticamente.

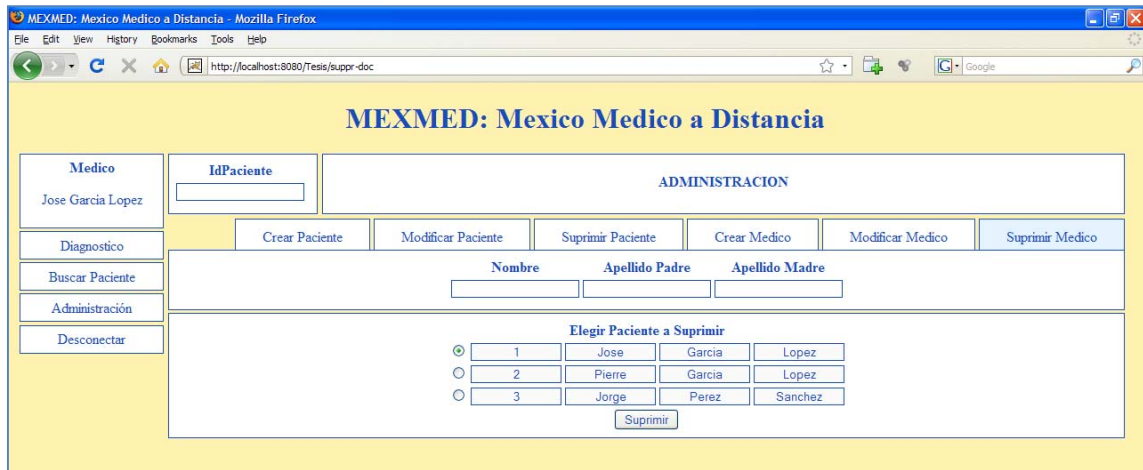


Figura 61: Administración – Supresión de un médico

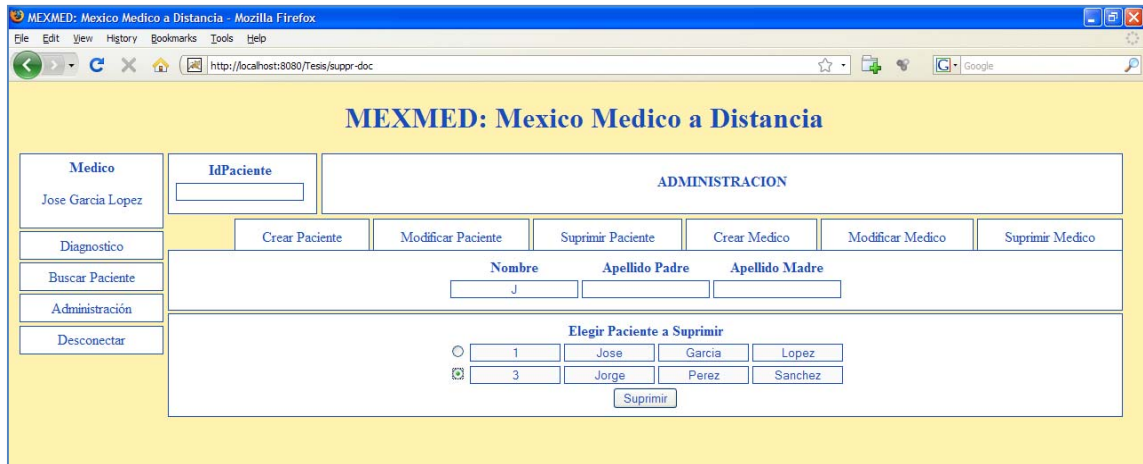


Figura 62: Administración – Búsqueda afinada del médico a suprimir

4.1.7. RSS con Ajax.

Para tener la impresión de un cambio automático de los datos en la página, la misma impresión de una actualización RSS (en inglés Really Simple Syndication), una opción es utilizar un servidor “push” que continuamente podría agregar datos en la página. Después de una investigación sobre este tema y encontrando varias técnicas que hace posible esa funcionalidad, se determinó que la mejor posibilidad es utilizar Ajax y especialmente el objeto XMLHttpRequest. Permite construir una petición al servidor de manera asíncrona. La página es cargada pero faltan algunos datos. Gracias a una llamada a una función Javascript, se puede crear un objeto XMLHttpRequest, para construir una petición a través de un Servlet del lado servidor y presentar el resultado de esa petición para añadir datos en la página. Esa funcionalidad fue agregada al sistema en varias páginas del sub-sistema del lado del médico. En esta sección se detallan las llamadas y las respuestas.

4.1.7.1. RSS de llamadas en la fila de espera.

En la página accueil.jsp, al final de la carga se llama a la función Javascript rssNumero() que construye un objeto XMLHttpRequest. A partir de ese objeto, se hace una petición al Servlet rssNumero del lado servidor. Es una petición de tipo GET que no tiene parámetros, entonces se necesita nada más invocar el url del Servlet rssNumero sin agregar parámetros al URL. El resultado de ese Servlet es un archivo XML que podría contener los números a llamar de las consultas en la fila de espera. La Figura 63 muestra el resultado si no hay consultas en espera. Hay una etiqueta respuesta que contiene la palabra invalid. La función Javascript va a tratar esa respuesta, cambiando el parámetro innerHTML del div de id igual a numero_accueil, escribiendo una frase diciendo que se tiene que esperar la próxima llamada (Figura 64).

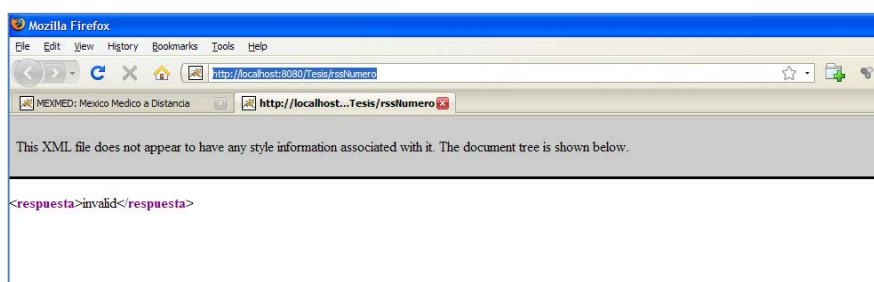


Figura 63: Archivo XML de respuesta al Servlet rssNumero con fila vacía.

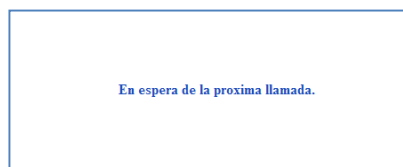


Figura 64: El archivo XML transformado en contenido XML

La Figura 65 muestra el resultado de esa llamada cuando hay varias consultas en espera. Hay una etiqueta raíz “respuesta” que contiene varias etiquetas “next”. Las cuales contienen dos elementos: el id de la consulta y el número de teléfono. A partir de ese archivo XML se puede construir, como lo muestra la Figura 66, un div con un grupo de botón radio que nos permite elegir la próxima llamada.

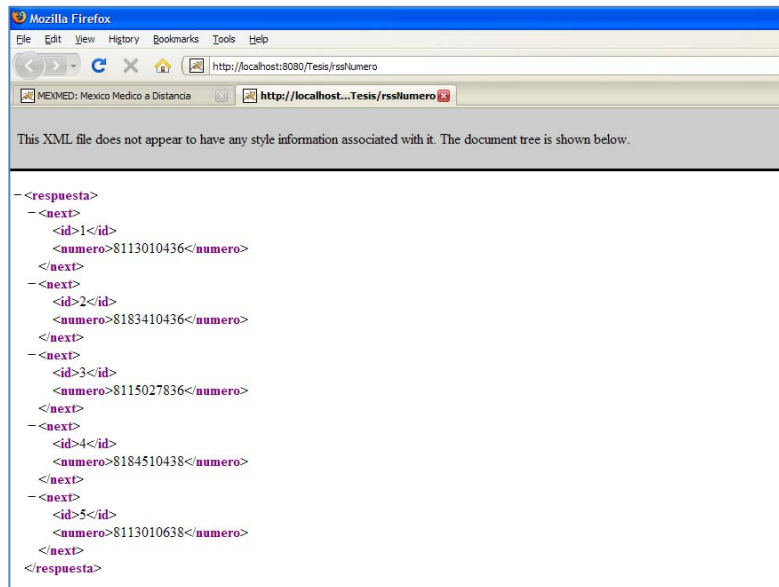


Figura 65: Archivo XML de respuesta al Servlet rssNumero

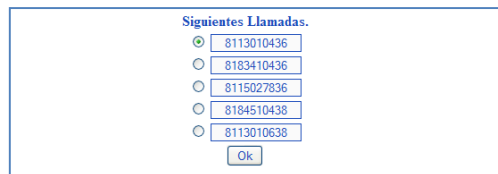


Figura 66: El archivo XML transformado en contenido HTML.

4.1.7.2. RSS SMS no leídos del paciente

En la página diagnostico.jsp, al final de la carga se llama a la función Javascript rssSMS() que construye un objeto XMLHttpRequest. Utilizando ese objeto, se hace una petición al Servlet rssSMS del lado servidor. Es una petición de tipo GET que tiene como parámetro el id del paciente en diagnostico, entonces se necesita llamar el url del Servlet rssSMS agregando el valor del parámetro al URL. Por ejemplo si se busca los SMS no leídos del paciente de id igual a 1, el URL será rssSMS?id=1. El resultado de ese Servlet es un archivo XML que potencialmente contendrá los SMS a leer del paciente en proceso de diagnóstico. La Figura 67 muestra el resultado si no hay SMSs a leer. Hay una etiqueta “respuesta” que contiene la palabra invalid. La función Javascript va a procesar esa respuesta, cambiando el parámetro innerHTML del div de id igual a sms, escribiendo una frase diciendo que no hay sms a leer por ese paciente (Figura 68).

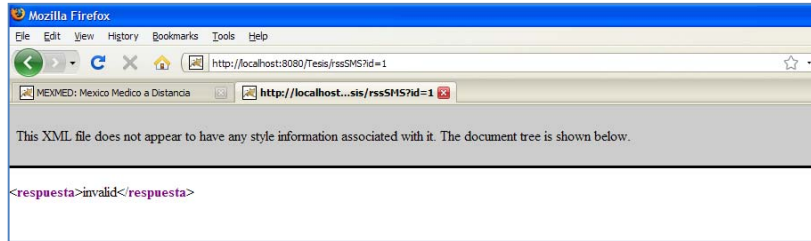


Figura 67: Archivo XML de respuesta al Servlet rssSMS si SMS a leer.

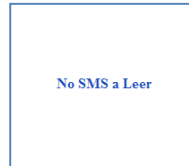


Figura 68: El archivo XML transformado en contenido HTML.

La Figura 69 muestra el resultado de esa llamada cuando hay varios SMS a leer de un paciente. Hay una etiqueta raíz “respuesta” que contiene varias etiquetas “sms”. Las cuales contienen tres elementos: el id del sms, su fecha y su mensaje. A partir de ese archivo XML se puede construir, como lo muestra la Figura 70, un div con un grupo de botón radio que nos permite leer un SMS.

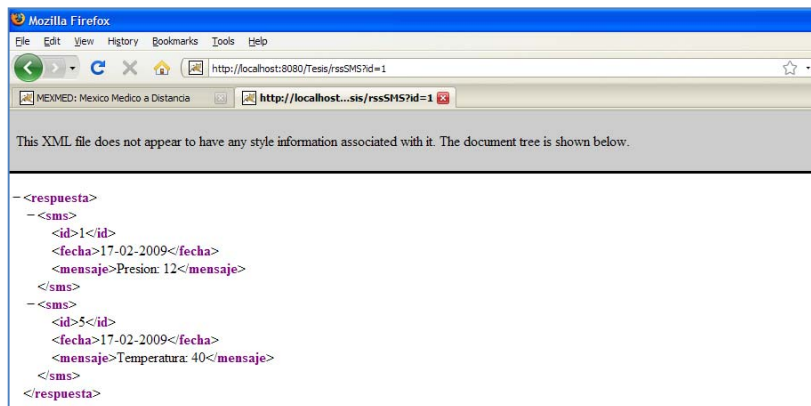


Figura 69: Archivo XML de respuesta al Servlet rssSMS

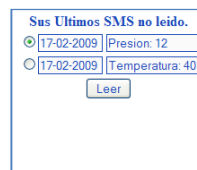


Figura 70: El archivo XML transformado en contenido HTML.

4.1.7.3. Buscar el id de un paciente

En la página buscar_paciente.jsp, al final de la carga se llama la función Javascript rssPaciente() que construye un objeto XMLHttpRequest. A partir de ese objeto, se hace una petición al Servlet rssPaciente del lado servidor. Es una petición de tipo GET que tiene como parámetro el nombre du paciente a buscar, su primer apellido y su segundo apellido; al inicio como no se sabe nada del paciente, entonces se necesita llamar al URL del Servlet rssSMS, sin agregar el valor de las parámetros al URL. Por ejemplo, si se busca un paciente con

primera letra del nombre la letra A, el URL será rssPaciente?nombre=A&padre=&madre=. El resultado de ese Servlet es un archivo XML que contiene todos los pacientes compatibles con la búsqueda. La Figura 71 muestra el resultado de esa llamada cuando hay varios pacientes compatibles con la búsqueda. Hay una etiqueta raíz “respuesta” que contiene varias etiquetas “paciente”. Las cuales contienen cuatros elementos: el id del paciente, su nombre, su primer apellido y su segundo apellido. A partir de ese archivo XML se puede construir, como lo muestra la Figura 72, un div con una tabla de pacientes. Se utiliza la misma función en la página suprimir.jsp donde se tiene que elegir el paciente a suprimir. Además una función equivalente se encuentra en la página suprimir-doctor.jsp donde se tiene que elegir el médico a suprimir.

```

<respuesta>
  <paciente>
    <id>1</id>
    <nombre>Louis</nombre>
    <padre>Dupond</padre>
    <madre>Martin</madre>
  </paciente>
  <paciente>
    <id>2</id>
    <nombre>Andre</nombre>
    <padre>Durand</padre>
    <madre>leRoux</madre>
  </paciente>
  <paciente>
    <id>3</id>
    <nombre>Julie</nombre>
    <padre>Lopez</padre>
    <madre>Andre</madre>
  </paciente>
  <paciente>
    <id>4</id>
    <nombre>Gary</nombre>
    <padre>Smith</padre>
    <madre>Artur</madre>
  </paciente>
</respuesta>

```

Figura 71: Archivo XML de respuesta al Servlet rssPaciente

Pacientes Correspondientes			
1	Louis	Dupond	Martin
2	Andre	Durand	leRoux
3	Julie	Lopez	Andre
4	Gary	Smith	Artur

Figura 72: El archivo XML transformado en contenido HTML.

4.2. Sub-Sistema del lado del Paciente

En esta sección, se analizará el sub-sistema del lado del paciente de forma detallada.

Se describe el contexto del sistema, la arquitectura, los diagramas de secuencia de varios casos de uso del sistema y por fin se muestran las funcionalidades implementadas en el sub-sistema.

4.2.1. Contexto del Sub-Sistema.

Una paciente llega a solicitar una consulta. Espera su turno en la sala de espera. Una vez llamado por el paramédico o la persona encargada de la oficina, puede registrarse en el sistema. Este registro se hace vía el envío de un SMS de tipo 1 a la central. Después de eso, tiene que esperar la llamada de un médico. Su espera depende de la cantidad de médicos y del número de oficinas en espera.

En la oficina, hay dos aparatos importantes. El primero permite enviar SMSs y proporcionar los datos del paciente tal como su temperatura o su tensión arterial, el otro permite recibir llamadas de un médico.

El paciente recibe la llamada del médico y empieza una conversación entre los dos. El médico le pregunta su número de paciente para visualizar su información tal como su nombre o su historial en el sistema. El paciente describe sus síntomas; el médico puede necesitar información adicional para verificar los síntomas y confirmar su diagnóstico. Esa información adicional se obtiene mediante varios aparatos y se envía mediante un SMS de tipo 2 al médico para que pueda analizar la nueva información. Esa etapa puede repetirse varias veces hasta que el médico puede terminar su diagnóstico.

4.2.2. Arquitectura del Sub-Sistema.

El sub-sistema del lado del paciente (Figura 73) se compone de un teléfono estacionario para recibir las llamadas del médico, un teléfono celular ordinario para enviar SMS, el cual tiene instalada una aplicación JME (auxiliándose de varios aparatos de toma de datos). Por el momento, los aparatos de toma de datos están desconectados del teléfono de envío de SMS pero una posibilidad sería conectar todos los aparatos al celular para construir el SMS de datos automáticamente, sin la necesidad de introducir manualmente los datos en la aplicación.



Figura 73: Organización del parte del paciente.

4.2.3. Diagrama de secuencia

Ese diagrama de secuencia (Figura 74) muestra la construcción de un SMS de tipo 2 gracias a la aplicación JME. Al inicio se muestra una lista para elegir los datos. Una vez elegidos los datos, se muestra un formulario para ingresar el resultado por cada dato. A continuación de eso, se permite agregar comentarios, por fin se muestra el SMS antes de enviarlo. Las pantallas en JME son como un juego de carta nada más se elige cual pantalla se muestra. Por eso con el botón se puede regresar a las pantallas anteriores y modificar los datos.

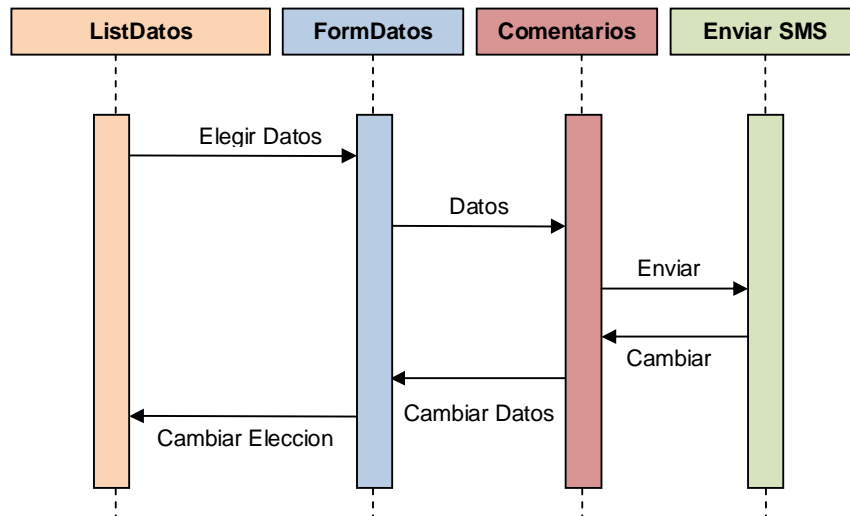


Figura 74: Diagrama de Secuencia Paciente

4.2.4. Desarrollar con JME

En JME, como ya fue explicado en el capítulo 2, una aplicación es un conjunto de pantalla del cual se tiene que elegir la cuál va a ser mostrado. Se tiene que pensar que una pantalla es de un sub-tipo Displayable y no se puede poner en la misma pantalla un Textarea y una lista.

Lo interesante de desarrollar con JME es la capacidad de probar la aplicación con un emulador de celular. No se necesita poner la aplicación en el celular para probarla.

4.2.5. Funcionalidades del Sub-Sistema

La aplicación permite enviar un SMS de tipo 1 para registrarse en el sistema. La Figura 75 muestra la primera pantalla de la aplicación que confirma el envío del SMS y el registro en el sistema.

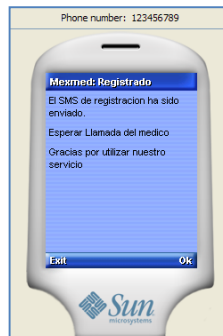


Figura 75: Pantalla de confirmación de registración

Las pantallas siguientes permiten construir un SMS de tipo 2 para enviar datos al médico. La Figura 76 muestra la pantalla para la elección de los datos. A partir de esa decisión, se construye la siguiente pantalla. Puede tener varios textfield para ingresar los datos correspondientes o ninguno si ningún dato ha sido elegido.

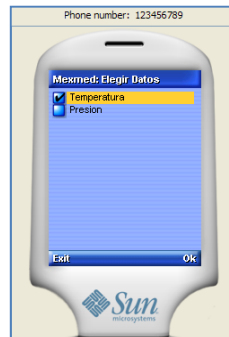


Figura 76: Pantalla de selección de los datos

La Figura 77 muestra diferentes casos de pantalla para ingresar los datos. Al final a partir de los datos ingresados se construye un SMS de tipo 2.

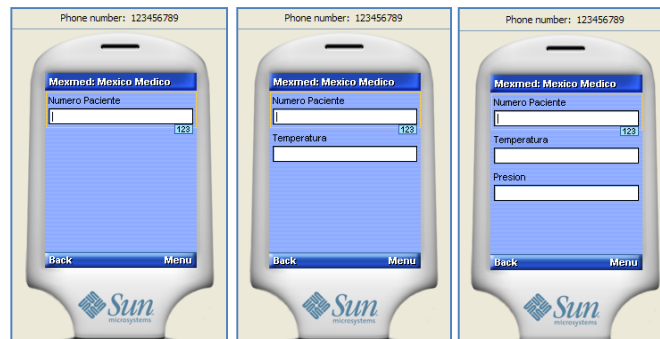


Figura 77: Diferentes pantallas para ingresar datos

La Figura 78 muestra la pantalla para ingresar comentarios en caso de ser necesario. Al final, tal como lo muestra la Figura 79 se presenta la pantalla de confirmación del SMS antes del envío. Para construir los dos tipos de SMS, se tienen que seguir varias reglas. La letra “;” sirve de separador entre dos elementos diferentes del SMS. Por ejemplo, la primera parte del SMS es el tipo de SMS, contiene un entero que puede ser 1 o 2 según el tipo del SMS. Al final del tipo de mensaje se tiene que agregar la letra “;”. Un SMS de tipo 1 será un mensaje igual a “1;”. Un SMS de tipo 2 es más complejo porque contiene el número del paciente y sus datos. Su presentación será de la forma siguiente: “2;3556;Temperatura: 39,6” donde 2 es el tipo del SMS y 3556 es el numero del paciente.



Figura 78: Pantalla en caso de comentarios adicional



Figura 79: Pantalla de confirmación del SMS

4.3. Sub-Sistema de Recuperación de los datos

En esta sección, se describe el sub-sistema de recuperación de los datos.

Se expone el contexto del sub-sistema, la arquitectura del sub-sistema, los diagramas de secuencia de varios casos de uso del sub-sistema y al final se muestra las funcionalidades implementadas en el sub-sistema.

4.3.1. Contexto del Sub-Sistema.

El servidor de recuperación de los datos permite recuperar los SMS (solicitudes de consulta, por ejemplo) pendientes de ser atendidos. Distingue si es un SMS es de registro al sistema (tipo 1) o de envió de datos (tipo 2). Puede conectarse a la base de datos y añadir una petición de consulta en la fila de espera o añadir un SMS no leído de un paciente. Si el SMS recibido no es de los tipos del sistema (por ejemplo un falso número), no se conecta a la base de datos. Por fin, suprime el SMS del celular para recuperar más SMS.

4.3.2. Arquitectura del Sub-Sistema.

La arquitectura del sub-sistema de recuperación de los datos se compone de tres partes (Figura 80). La primera es un celular conectado al servidor por un cable USB. La segunda parte es la aplicación del servidor que permite conectarse con el celular y navegar a en la base de datos para recuperar por ejemplo los SMS, además puede conectarse con la base de datos y modificarla. Por fin hay un DBMS que se encarga de manejar la base de datos.

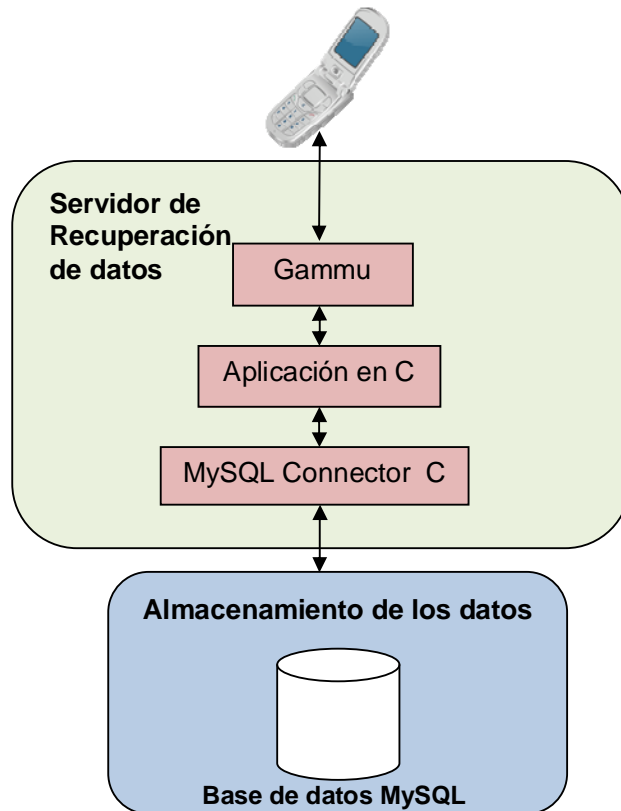


Figura 80: Organización de la recuperación de los datos.

4.3.3. Diagrama de secuencia

Este diagrama de secuencia (Figura 81) permite entender la aplicación de recuperación de datos.

Primero se verifica si hay SMSs en la memoria. Se recupera el primer SMS y si no hay problema durante la recuperación se analiza el mensaje para dividirlo en sus componentes, dependiendo del tipo de mensaje. Con un tipo 1, se busca el número de donde se envió el SMS para encontrar de cual consultorio procede y añadir la petición de consulta en la fila de espera (en la base de datos). Con un tipo 2, se busca el número del paciente y el mensaje para añadir un SMS no leído de un paciente.

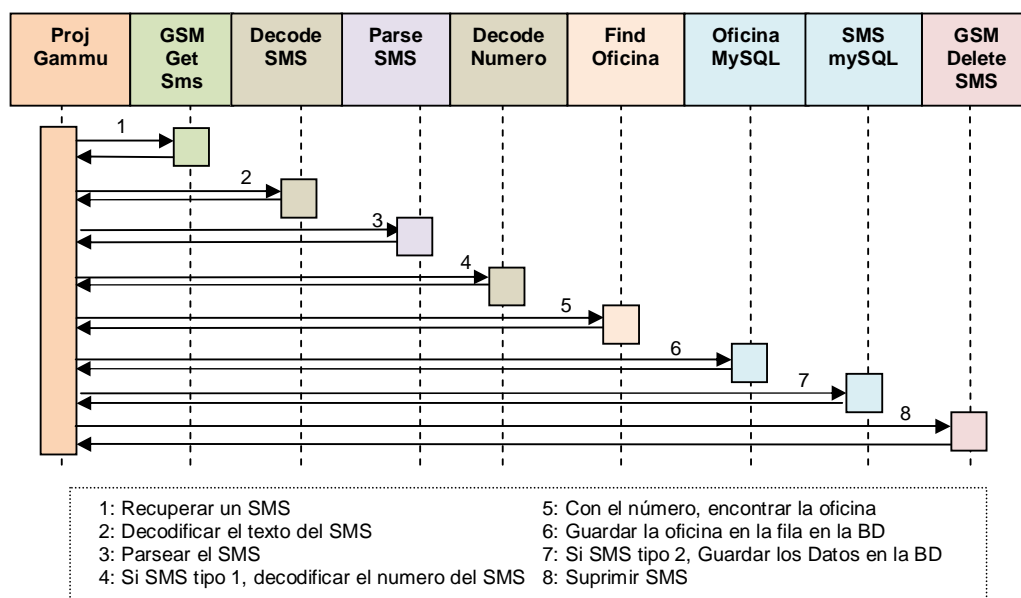


Figura 81: Diagrama de secuencia de recuperación de los SMS.

4.3.4. Funcionalidad del Sub-Sistema.

El sub-sistema de recuperación de datos permite conectarse a un celular conectado al servidor con un cable USB. Puede recuperar los SMS recibidos, tratarlos de la manera que se necesita según el tipo del SMS y agregar la información recibida en la base de datos; también permite suprimir los SMS del celular.

Esa aplicación tiene una salida en la consola que podemos analizar para ver el funcionamiento del programa. Esto se presenta en varias de las figuras siguientes.

Primero, las figuras 82 y 83 permiten ver el manejo de la conexión al celular. La primera muestra el fallo en la conexión debido a que el teléfono celular no está presente. Una vez conectado se puede realizar los pasos descritos en el diagrama de secuencia explicado en la sección anterior.

```
[Gammu - 1.26.1 built 09:57:29 Sep 8 2009 using GCC 4.3, MinGW 3.15]
[Connection - "at19200"]
[Connection index - 0]
[Model type - "at"]
[Device - "com10:"]
[Running on - Windows XP SP3]
[Module - "A2D:iPAQ:at:IM20:IS25:IMC35:IC35:IS65:IS300:IS110:IS130:IS190:
5210:IS6110:IS6130:IS6150:IS6190:IS6210:IS6250:IS6310:IS6310i:IS6510:IS7110:IS8210:IS8250:IS8290:IS8310:IS8390:
IS8850:IS8855:IS8890:IS8910:IS9110:IS9210"]
Device is \\.\COM10
[System error - CreateFile in serial_open, 2, "Le fichier spécifié est intro
uvable. "]
Init:GSM_OpenConnection failed with error DEVICENOTEXIST[4]: Error opening device, it doesn't exist.
FALLO CONNEXION
-
```

Figura 82: Fallo de conexión

```
311:12C,1300:1311:13A:1355:1344:13A:1311:1344:12B+1300:1300:122:10D:10D 1,01:54:14+00".
0A:14F0:14BK:10D:10A .OK..
Getting firmware versions
SENDING frametype 0x00/length 0x08/8
41A:154T:12B+143C:147G:14DM:152R:10D AT+CGMR.
1 "AT+CGMR"
2 "R9A035 R9A035 prgCXC1123260_GENERIC_JE"
3 "OK"
RECEIVED frametype 0x00/length 0x3E/62
41A:154T:12B+143C:147G:14DM:152R:10D:10D:10A:152R:1399:141A:1300:1333:1355 AT+CGMR...R9A035
20:120:152R:1399:141A:1300:1333:1355:120:120:120:120:120:170p:172r R9A035 pr
67g:143C:158X:143C:1311:1311:1322:1333:1322:1366:1300:15F_:147G:145E:14EN:145E gCXC1123260_GENE
52R:149I:143C:15F_:14AJ:145E:10D:10A:10D:10A:14F0:14BK:10D:10A RIC_JE...OK..
Received firmware version: "R9A035 R9A035 prgCXC1123260_GENERIC_JE"
Number version is "903590351123260_000000"
[Firmware version - "R9A035 R9A035 prgCXC1123260_GENERIC_JE"]
[Connected]
EXITO CONNEXION
```

Figura 83: Éxito de conexión

Las figuras 84 y 85 muestran la diferencia entre la ausencia de SMS en el celular y la presencia de dos SMS. En caso de ausencia de SMS, se reinicia el ciclo para esperar la llegada de otros mensajes. En el otro caso, se trata los SMS.

```
22":10D "
1 "AT+CPMS="ME","ME""
2 "+CPMS: 0,200,0,200,0,200"
3 "OK"
RECEIVED frametype 0x00/length 0x34/52
41A:154T:12B+143C:150P:14DM:153S:13D=122":14DM:145E:122":12C,122":14DM:145E AT+CPMS="ME","ME
22":10D:10D:10A:12B+143C:150P:14DM:153S:13A:120:1300:12C,1322:1300:1300 "...+CPMS: 0,200
2C,1300:12C,1322:1300:1300:12C,1300:12C,1322:1300:1300:10D:10A:10D:10A .0,200,0,200....
4F0:14BK:10D:10A OK..
SMS status received
Parsing +CPMS: 0,200,0,200,0,200 with +CPMS: Ei, Ei, 00
Parsed int 0
Parsed int 200
Used : 0
Size : 200
Leaving GSM_GetSMSStatus
Status Used phone: 0
Status Used SIM: 0
```

Figura 84: Ausencia de SMS

```

22"10D
1 "AT+CPMS="ME","ME""
2 "+CPMS: 2,200,2,200,2,200"
3 "OK"
RECEIVED frametype 0x00/length 0x34/52
41A154T12B+143C150P14DM153S13D=122"14DM145E122"12C,122"14DM145E AT+CPMS="ME", "ME
22"10D 10D 10A 12B+143C150P14DM153S13A:120 132212C,132213001300 "...+CPMS: 2,200
2C,132212C,13221300130012C,132212C,13221300130010D 10A 10D 10A ,2,200,2,200...
4FO14BK10D 10A OK..

SMS status received
Parsing +CPMS: 2,200,2,200,2,200 with +CPMS: Ei, Ei, E0
Parsed int 2
Parsed int 200
Used : 2
Size : 200
Leaving GSM_GetSMSStatus
Status Used phone: 2
Status Used SIM: 0

```

Figura 85: Presencia de dos SMS

Las figuras 86 y 87 muestran la diferencia de tratamiento entre un SMS de tipo 1 y de tipo 2. Para un SMS de tipo 1, se necesita recuperar el número de teléfono del emisor del SMS para buscar a cual oficina corresponde y poder registrar la oficina en espera en la base de datos. En el caso de un SMS de tipo 2, se recupera del SMS el ID del paciente y sus datos. Con esa información se puede agregar los datos del paciente en la base de datos como un SMS no leído del paciente.

```

SMS tipo 1
Leaving parse_sms
tipo:1 id:0 text:ytR
Decode Numero
2
8113010436
Leaving decode_Numero
Initialisation reussie de Mysql
Connexion reussie au serveur Mysql
Selection de la base de donnee reussie
prepare, INSERT successful
total parameters in INSERT: 1
total columns in SELECT statement: 1
Fetching results ...
row 1
column1 (integer) : 1(4)

total rows fetched: 1
la oficina del numero 8113010436 es la oficina 1
Initialisation reussie de Mysql
Connexion reussie au serveur Mysql
Selection de la base de donnee reussie
prepare, INSERT successful
total parameters in INSERT: 1
Oficina agregado a la BD

```

Figura 86: SMS de tipo 1, Oficina agregado a la base de datos

```

SMS PID: 0x00
SMS DCS: 0x11
SMS class: 1
Decoding date & time: 11/11/2009 01:44:40 -0600 (mer.)
UDH header available (length 1)
7 bit SMS, length 21
2;4;Temperatura: 37.5
ATGEN folder 2 & location 1 -> SMS folder 0 & location 100001
Leaving GSM_GetSMS
Recuperacion Exitosa
Decode SMS
21
2;4;Temperatura: 37.5
Leaving decode_SMS
Parse SMS
SMS tipo 2
Leaving parse_sms
tipo:2 id:4 text:Temperatura: 37.5
Temperatura: 37.5 4
Initialisation reussie de Mysql
Connexion reussie au serveur Mysql
Selection de la base de donnee reussie
prepare, INSERT successful
total parameters in INSERT: 3
SMS de datos agregado en la BD

```

Figura 87: SMS de tipo 2, Datos agregados a la base de datos

La figura 88 muestra la supresión de un SMS. La supresión de los SMS después de su tratamiento permite tener siempre el celular disponible para la recepción de nuevos SMS.

```
SMS type: Deliver, UDH included
L=10
Default number 81 <1 0 0 0;0 0 0 1>
Len 5
Remote number : "8113010436"
SMS PID: 0x00
SMS DCS: 0x11
SMS class: 1
Decoding date & time: 11/11/2009 01:12:08 -0600 <mer.>
UDH header available <length 1>
7 bit SMS, length 2
1;
ATGEN folder 2 & location 1 -> SMS folder 0 & location 100001
SMS folder 3 & location 1 -> ATGEN folder 2 & location 1
Deleting SMS
SENDING frametype 0x00/length 0x0A/10
41A:54T:2B+:43C:4DM:47G:44D:3D=:311:0D AT+CMGD=1.

1 "AT+CMGD=1"
2 "OK"
RECEIVED frametype 0x00/length 0x10/16
41A:54T:2B+:43C:4DM:47G:44D:3D=:311:0D :0D :0A :4FO:4BK:0D :0A AT+CMGD=1...OK..

SMS deleted OK
Leaving GSM_DeleteSMS
```

Figura 88: Supresión del SMS

4.4. Resumen del Capitulo

En este capítulo se describió en detalle la implementación del sistema, tomando en cuenta cada sub-sistema que lo compone.

El sub-sistema del lado del médico desarrollado con la tecnología Java Servlet y JSP permite construir una interfaz gráfica fácil de uso para los médicos. El sub-sistema del lado del paciente desarrollado con la tecnología Java ME permite tener una arquitectura de bajo costo sin perder la funcionalidad de enviar los datos del paciente al médico. Por fin el sub-sistema de recuperación de los datos desarrollado en C utilizando el API Gammu para la comunicación con el celular y el API MySQL Connector C para la conexión con la base de datos permite agregar automáticamente los SMS recibidos en la base de datos para que el médico les pueda utilizar lo más rápidamente posible.

El capítulo siguiente es el capítulo de conclusión en el cual vamos a tratar también el trabajo futuro.

Capítulo 5. Conclusiones y Trabajo Futuro

En este capítulo se resumen los resultados obtenidos de la propuesta presentada en el capítulo 3 y ejemplificada en el capítulo 4. A su vez, se detalla las experiencias aprendidas y el trabajo futuro relacionado.

5.1. Resumen

Este documento de tesis se enfocaba a ofrecer una solución al tema de asistencia médica a personas que residen en áreas rurales de la Republica Mexicana.

Se ha analizado y propuesto el diseño de un sistema para el problema de atención médica a personas que residen en áreas aisladas, utilizando la red telefónica inalámbrica de frecuencias bajas de la banda UHF (CDMA450) para permitir la comunicación entre un médico y un paciente.

Para lograr lo anterior, fue necesario realizar una investigación acerca del estado de la telemedicina en el mundo y especialmente en México. Además de eso fue importante investigar sobre las tecnologías que se podrían utilizar para implementar un prototipo de la solución. Las tecnologías investigadas fueron Java Servlet, JSP, AJAX, JDBC, MySQL para la interfaz y la lógica del médico; JME para la interfaz y la construcción de los SMS del paciente; Gammu y MySQL Connector C por la recuperación de los SMS y guardarlos en la base de datos.

Se demostró un prototipo de un sistema que permite la comunicación entre médicos y pacientes, y la transmisión de datos del paciente para que el médico los pueda analizar. El prototipo fue desarrollado para mostrar cómo podría funcionar el sistema utilizando las tecnologías investigadas.

5.2. Experiencias Aprendidas

Gracias al proceso de investigación desarrollado para la elaboración de este trabajo de tesis, se pueden resaltar las siguientes experiencias:

- Después de la investigación sobre Java Servlet y páginas JSP, se podría pensar que es lo mismo utilizar uno que el otro. Utilizando el patrón de diseño MVC, se ha podido diferenciar las acciones de los Java Servlet y de las páginas JSP. Se utiliza nada más las páginas JSP para la construcción de la interfaz del médico, utilizando los datos recibidos del Servlet. Los datos son recuperados por el Servlet y a partir de los datos recibidos puede decidir cuál será la siguiente página JSP a desplegar. Las páginas JSP son al final un tipo de Servlet pero como el enfoque de la programación es diferente de los Servlet, es muy fácil construir contenidos Web.

- Las Rich Internet Applications no solo ofrecen una nueva manera de desplegar contenidos en el navegador Web, sino que también cambian la perspectiva del desarrollador en cuanto a la manera de estructurar una aplicación. Ya que la correcta y oportuna identificación de las capas que conforman al sistema determinan los límites y operaciones de cada componente. Permitiendo la interoperabilidad con otros sistemas que pueden complementar a la aplicación original, esto se traduce en mayor complejidad en el desarrollo.

- La plataforma JME fue muy interesante de utilizar. Poder emular la aplicación para ver su funcionamiento en un celular sin tener que instalarlo es muy útil porque permite probar directamente, sin tener que instalar la aplicación en el celular para probarla.

- Fue un poco complejo de utilizar el API Gammu. Principalmente debido a falta de claridad de los documentos y ausencia de ejemplos para apoyarse. Por ejemplo en la parte de la recuperación de los datos no se especifica en ningún lugar que para recuperar cualquier dato se tiene que hacer una decodificación del contenido. Sin esa decodificación, no aparece ningún dato y se puede pensar que hay un problema en su código, sin que realmente lo haya. Una vez encontrada la solución a este problema, el API Gammu es bastante interesante para operar sobre los SMS.

5.3. Conclusiones

En conclusión, podemos decir que esta tesis logra una conjunción interesante de muchas tecnologías para enfrentar un problema y proponer una solución. Por ejemplo, con las tecnologías de Java con la parte de aplicación Web (Java Servlet y JSP) y la parte de aplicación móvil (JME), es realmente interesante ver como se combinan para soportar la funcionalidad requerida en varios de los sub-sistemas. Además fue muy útil poder utilizar el lenguaje C para el sub-sistema de recuperación de los datos.

Un tema que motivó el trabajo de tesis fue la posibilidad de actuar a nivel social gracias a las tecnologías de información y de comunicación. Ese tema fue determinante para que el autor decidiera adoptarlo como tema de tesis. Por eso la investigación del estado del arte en Telemedicina en México ha sido muy interesante permitiendo destacar la necesidad de un cambio del enfoque de la Tele-Salud en México. No solo ofrecer servicios de especialistas a las comunidades rurales sino adicionalmente ofrecer servicios de medicina general.

Es muy interesante ver el aumento de las publicaciones desde el inicio del año 2009, lo cual muestra un interés global por este tema e indica que es un tema de actualidad.

5.4. Trabajo Futuro

Este trabajo propone una solución al problema a la asistencia médica a personas aisladas. Existe muchas posibles mejoras aplicables a este trabajo, entre las cuales estarían:

- Aumentar los posibles datos que pueda enviar el paciente al médico. Por el momento, nada más se transmitirían la temperatura y la tensión arterial, pero se puede enviar todo lo que pudiera necesitar el médico para efectuar el diagnóstico del paciente.
- Diversificar el contenido y el tipo de datos enviado al médico. Es decir, por el momento se utiliza nada más los SMS para enviar los datos del paciente. Pero se podría enviar MMS para tener imágenes que pueda utilizar el médico. Todo depende de lo que se pueda hacer con la tecnología CDMA450. Otra mejora sería utilizar un posible CDMA450 3G para hacer video conferencia, pero esto también depende de la tecnología.
- Automatizar la creación del SMS de tipo 2, gracias a una conexión de los aparatos de toma (termómetro electrónico, etc.) al celular de envío de SMS. Al final de cada examen, se podría enviar los datos directamente sin que el paciente o el paramédico tenga que introducir el resultado al celular. Eso permitiría dar mayor confianza al médico, que recibe los datos sin que nadie ya lo ha alimentado manualmente.
- Presentación del proyecto a la Secretaría de Salud para poder implementarlo al nivel nacional, como apoyo a algunos de los programas actuales (Seguro Popular, etc.)

Referencias bibliográficas

- Augustinos, N., & Shehata, A. (2009). *Cisco HealthPresence™ - Transforming Access to Healthcare*. Cisco IBSG.
- Chauve, J.-Y. (2002). *Le guide de la médecine à distance Volume 1, Consulter un médecin à distance*. Distance Assistance.
- Crawford, W., & Hunter, J. (2002). *Servlets Java*. O'Reilly.
- Dubois, P. (2000). *MySQL*. CampusPress Reference.
- Fitzmaurice, M. J. (1998). *Telehealth Research and Evaluation: Implications for Decision Makers*. IEEE.
- Gantenbein, R. E., Cadez, F., & Stamm, B. H. (2004). Establishing a Rural Telehealth Project: the Wyoming Network for Telehealth. *Conference of the IEEE EMBS* (págs. 3089-3092). San Francisco: IEEE.
- Garavaglia, A., Murray, K. P., Baker, K. R., Chevalier, C., & Brunner, C. (2006). *WDCMA (UMTS) Deployment Handbook*. Wiley.
- Gass, V., Reichert, S., Gass, R., & Andrès, E. (2007). *Le stéthoscope à l'ère de la médecine factuelle*. ALSAMED.
- Hamphill, D., & White, J. (2002). *Java 2 Micro Edition*. Manning Publications.
- Harnett, B. (2006). Telemedecine Systems and Telecommunications. *Journal of Telemedicine and Telecare* , 4-15.
- Henni, J. (2003). Le CDMA progresse vite et bien. *Les Echos* , pág. 102.
- International Telecommunication Union, & Pan American Health Organization. (2003). *TeleHealth in the Americas*. Inter-American Telecommunication Commission.
- Johnson, C. (2009). Docs Say Goodbye to Bureaucracy, Hello Health. *Health Leader Media* .
- Legmann, M., & Lucas, J. (2009). *Télémédecine - Les préconisations du Conseil National des Médecins*. Conseil national de l'ordre des médecins.
- Murray, G. (2006). *Asynchronous JavaScript Technology and XML (Ajax) with the Java Platform*.
- Perry, B. W. (2004). *Java Servlet and JSP Cookbook*. O'Reilly.
- Reenskaug, T. (2003). *The Model-View-Controller (MVC) Its Past and Present*.
- Reese, G. (2001). *JDBC and Java*. O'Reilly.
- Scott, M., & Braithwaite, C. (2003). *UMTS Network Planning and Development*. Newnes.
- Tisal, J. (2003). *Le Réseau GSM (4Ed) Evolution GPRS, I-mode, Wap*. Dunod. 01 informatique.

WHO Global Observatory for eHealth. (2006). *Building Foundations of e-Health - Region of the Americas*. World Health Organization.

Williams, D. E. (2009). Podcast interview with Hello Health co-founder Dr. Jay Parkinson. *Health Business Blog* .

Zheng, L. (2007). *Dr. Microsoft is now ready to see you*. Started Something.