

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

DESARROLLO DE SISTEMA CLÍNICO ORIENTADO A LA TELEMEDICINA

MARCELO ALEJANDRO FLORES VILLABLANCA

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

DICIEMBRE 2012

Pontificia Universidad Católica De Valparaíso
Facultad De Ingeniería
Escuela De Ingeniería Informática

DESARROLLO DE SISTEMA CLÍNICO ORIENTADO A LA TELEMEDICINA

MARCELO ALEJANDRO FLORES VILLABLANCA

Profesor Guía: **José Rubio León**

Profesor Co-referente: **Aldo Migliaro Osorio**

Carrera: **Ingeniería de Ejecución en Informática**

DICIEMBRE 2012

*Con mucho amor a mi familia, especialmente a mis padres que
creyeron ciegamente en mi proyecto de vida y me enseñaron a ser la
persona que hoy soy.
A la memoria de 2 grandes hombres, a mi abuelo Don Carlos Flores
Briones y a Don Oscar Lagos Covarrubias, que me enseñaron el
valor de la amistad incondicional.
A Jesucristo que siempre ha iluminando mi camino.*

Índice

Resumen	6
Lista de Figuras	7
Lista de Tablas	8
1. Capítulo primero: Descripción del proyecto	10
1.1. Introducción	10
1.2. Objetivos	10
1.2.1. Objetivo General	10
1.2.2. Objetivo Especifico.....	11
1.3. Descripción del Problema	11
1.4. Plan de trabajo.....	12
2. Capitulo Segundo: Investigación preliminar.....	13
2.1 Estado del Arte.....	13
2.2 Reseña Histórica.....	13
2.3 Bases de la Telemedicina (definición, objetivos).....	14
2.3.1 Objetivos de la Telemedicina.....	15
2.3.1.1 Asistencia Médica	15
2.3.1.2 Educación y Difusión	15
2.3.2 Aplicaciones de la Telemedicina.....	16
2.4 Telemedicina y su Aceptación	16
2.5 Manejo de la información clínica en telemedicina.....	17
2.5.1 Historia Clínica	17
2.5.2 Sistema de Salud	18
2.5.2.1 Principales Agentes del Sistema.....	19
2.5.2.1.1 Ministerio de Salud	19
2.5.2.1.2 Instituto de Salud Pública.....	20
3 Capítulo Tercero: Solución propuesta.....	22
3.1 Modelo de comunicación y conectividad.....	22
3.1.1 Arquitectura.....	22
3.1.1.1 Arquitectura Lógica.....	22
3.1.1.1.1 Nivel Superior Móvil - Presentación y Lógica.....	23
3.1.1.1.2 Nivel Superior Web - Presentación y Lógica.....	23

3.1.1.1.3	Nivel Inferior - Lógica y Datos	23
3.1.1.2	Comunicación.....	23
3.1.1.2.1	SOAP.....	24
3.1.1.2.1.1	Partes de un Mensaje SOAP.....	24
3.1.1.2.2	Web Services.....	25
3.1.1.2.2.1	Pila de protocolos para Servicios Web.....	25
3.1.1.2.2.2	Ventajas de los Web Services	26
3.2	Contexto Tecnológico	26
3.2.1	Android	26
3.2.1.1	Breve Historia	26
3.2.1.2	Características de Android	27
3.2.1.3	Arquitectura Android	27
3.2.2	SQL Lite (Base de Datos móvil)	28
3.2.2.1	Características SQL Lite	29
3.2.3	J2EE	29
3.2.3.1	Aplicaciones Distribuidas Multicapas.....	29
3.2.3.2	Componentes J2EE	30
3.2.3.3	Clientes J2EE	31
3.2.3.3.1	Cliente Web.....	31
3.2.3.3.2	Aplicaciones Clientes	31
3.2.3.4	Arquitectura de componentes JavaBeans	31
3.2.3.5	Comunicación con el Servidor J2EE.....	32
3.2.3.6	Componentes Web	32
3.2.3.7	Componentes de la parte Negocio.....	33
3.2.4	Contenedores J2EE	34
3.2.4.1	Tipos de Contenedores	34
3.2.4.1.1	Servidor J2EE.....	34
3.2.4.1.2	Contenedor de JavaBeans (EJB) empresariales	34
3.2.4.1.3	Contenedor Web.....	34
3.2.4.1.4	Contenedor de Aplicaciones cliente	35
3.2.4.2	Servicios de los Contenedores.....	35
3.2.5	J2EE APIs	35
3.2.5.1	Enterprise JavaBeans Technology.....	35

3.2.5.2	Java Servlet Technology	35
3.2.5.3	Java Server Pages Technology	36
3.2.5.4	Java Transaction API	36
3.2.5.5	JDBC API.....	36
3.2.5.6	Java Authentication and Authorization Service	36
3.2.5.7	Ventajas al utilizar las APIs	37
3.2.6	Oracle	37
3.2.6.1	Reseña histórica de Oracle	37
3.2.6.2	Características de Oracle	38
3.2.6.3	Ventajas.....	38
3.2.7	HTTPS.....	38
3.2.7.1	Funcionamiento de HTTPS	39
3.2.7.2	Desventajas.....	39
3.2.8	XML.....	39
3.2.8.1	Estructura de un documento XML	39
3.2.8.1.1	Cabecera.....	40
3.2.8.1.2	Cuerpo	40
3.2.8.1.3	Ventajas.....	40
3.2.8.1.4	Desventajas.....	40
3.2.8.1.5	Ejemplo de XML.....	41
3.2.9	Subversion-SVN.....	41
3.2.9.1	Versionado de directorios.....	41
3.2.9.2	Verdadero historial de versiones	41
3.2.9.3	Envíos atómicos	42
3.2.9.4	Versionado de metadatos	42
3.2.9.5	Manipulación consistente de datos.....	42
3.2.9.6	Herramientas	42
3.2.10	Google code.....	43
3.3	Metodologías de desarrollo	43
3.3.1	Modelo Evolutivo.....	44
3.3.1.1	Construcción de Prototipos	44
3.5.1.1	Problemas del Modelo Evolutivo.....	45
3.3.2	Ventajas del Modelo Evolutivo.....	45

3.3.3	Modelo en Cascada	45
3.3.3.1	Etapas del Modelo en Cascada.....	46
3.5.1.2	Problemas del Modelo en Cascada.....	47
3.3.4	Metodología para BD	47
3.3.5	Análisis.....	48
3.3.6	Especificación	48
3.5.1.3	Diseño	49
3.3.7	Implementación.....	49
4	Capítulo Cuarto: Desarrollo del proyecto	50
4.1	Alcances	50
4.2	Limitaciones	51
4.3	Objetivos del Sistema.....	51
4.4	Reglas de negocio.....	53
4.5	Requerimientos de información	53
4.6	Restricciones de información.....	56
4.7	Actores del sistema.....	57
4.8	Requerimientos funcionales	58
4.9	Requisitos no funcionales.....	62
4.10	Diagrama de Casos de uso	64
4.10.1	Diagrama Caso de Uso General	64
4.10.2	Caso de Uso Gestionar Usuario:	64
4.10.3	Caso de Uso Gestionar Visita:.....	65
4.10.4	Caso de Uso Gestionar Receta:	65
4.10.5	Caso de Uso Gestionar Examen:	66
4.10.6	Caso de Uso Gestionar Licencia:	66
4.11	Diagrama de clases.....	67
4.11.1	Sistema web.....	67
4.11.2	Sistema Móvil	69
4.12	Diagrama de Secuencias.....	70
4.12.1	Diagrama de Secuencia Inicio de Sesión Modulo Web	70
4.12.2	Diagrama de Secuencia Inicio de Sesión Modulo Móvil	71
4.12.3	Diagrama de Secuencia Gestionar usuarios	71
4.12.4	Desplegar agenda web.....	72

4.12.5	Diagramas de Actividad	73
4.12.6	Diagrama de Actividad Agendar Cita Médica	73
4.12.7	Diagrama de Actividad Guardar Archivos Adjuntos	74
4.13	Diseño de la base de datos.....	74
4.13.1	Diseño conceptual	74
4.13.2	Usuarios del sistema.....	74
4.13.3	Entidades	75
4.13.4	Diagramas relacional.....	76
5	Capítulo Quinto: Prototipo modulo Paciente	77
5.1	Web	77
5.1.1	Ingresar.....	77
5.1.2	Bienvenida Modulo Web	77
5.1.3	Reservar Hora.....	78
5.1.4	Listar Hora.....	79
5.2	Móvil.....	79
5.2.1	Ingresar.....	80
5.2.2	Bienvenida Modulo Móvil	81
5.2.3	Selección de Especialidad	82
5.2.4	Selección de Día.....	83
5.2.5	Resultado Final.....	84
6	Capítulo Sexto: Pruebas	85
6.1.1	Pruebas unitarias	85
6.1.2	Sistema web.....	85
6.1.3	Pruebas de integración	85
6.1.4	Sistema móvil.....	86
7	Capítulo Séptimo: Conclusiones y Trabajo Futuro	88
8	Capitulo Noveno: Bibliografía y Referencias	90
8.1	Bibliografía	90
8.2	Referencias electrónicas.....	90

Resumen

El Sistema Clínico orientado a la telemedicina propuesto como Proyecto de Título, busco la implementación de un estándar para mensajes que permitiera el intercambio de información médica entre las diferentes aplicaciones desarrolladas con distintas tecnologías.

En este proyecto se desarrollaron dos tipos de aplicaciones: un módulo web desarrollado en la plataforma J2EE con base de datos Oracle y 2 módulos para dispositivos móviles desarrollados en Android, utilizando Sqlite como respaldo de datos. La comunicación entre las distintas tecnologías se logró mediante el consumo de Web Services desarrollados en Java.

El módulo web permite realizar tareas como: Administración de perfiles que acceden al sistema, Afiliación de pacientes, Visualizador de documentos proporcionados por las aplicaciones móviles, Asignación de citas médicas.

Las aplicaciones para dispositivos móviles, permiten el acceso a usuarios registrados en el sistema que cuenten con el perfil de usuario "médico" o "paciente". La aplicación móvil para médico, permite fotografiar y registrar documentos que pueden ser del tipo: licencia, examen o receta y almacenarlos en la base de datos central del sistema, con respaldo en la base de datos del dispositivo móvil, para su posterior validación legal. La aplicación móvil para Pacientes, permite al usuario fijar una cita médica de acuerdo a la especialidad que este requiera. La cita médica estará condicionada a la disponibilidad de horarios del profesional de la salud requerido.

Abstract

The Clinical System oriented telemedicine project proposed as Degree University Title, seek to implement a messaging standard that allows the exchange medical information between different applications developed with different technologies.

In this project we developed two types of applications: a web module developed on J2EE platform with Oracle database and 2 modules for mobile devices developed on Android, using Sqlite as data backup. Communication between the various technologies was achieved by using Web Services developed in Java.

The web module to perform tasks such as: Profile Management accessing the system, Patients Affiliation, Document viewer provided by mobile applications, assigning medical appointments. On the other hand, applications for mobile devices, allow access to registered users in the system that have the user profile "doctor" or "patient".

The mobile application for medical, allows take a photograph and record documents that may be: license, exam or prescription and stored in the central database system with support for the database on the mobile device for later legal validation. Patients Mobile Application allows the user to set an appointment in accordance with the required specialty. The appointment will be subject to the availability of schedules required health professional.

Lista de Figuras

Ilustración 1 Arquitectura del sistema	22
Ilustración 2 Arquitectura del sistema	24
Ilustración 3 Arquitectura de WebServices	25
Ilustración 4 Arquitectura Android	28
Ilustración 5 Aplicaciones Multicapas	30
Ilustración 6 Comunicación con el Servidor	32
Ilustración 7 Parte Web y Aplicación J2EE	33
Ilustración 8 Parte Negocio y EIS	33
Ilustración 9 Servidor y contenedores de J2EE	34
Ilustración 10 Modelo Evolutivo.....	45
Ilustración 11 Modelo en Cascada.....	47
Ilustración 12 Diagrama Caso de Uso General	64
Ilustración 13 Caso de Uso Gestionar Usuario.....	64
Ilustración 14 Caso de Uso Gestionar Visita.....	65
Ilustración 15 Caso de Uso Gestionar Receta	65
Ilustración 16 Caso de Uso Gestionar Examen	66
Ilustración 17 Caso de Uso Gestionar Licencia.....	66
Ilustración 18 Diagrama de clases Conexión Sistema Móvil.....	69
Ilustración 19 Diagrama de Clases Móvil	70
Ilustración 20 Inicio de Sesión Modulo Web	70
Ilustración 21 Inicio de Sesión Modulo Móvil.....	71
Ilustración 22 Gestionar usuario.....	71
Ilustración 23 Agenda Web	72
Ilustración 24 Agendar Cita Médica.....	73
Ilustración 25 Diagrama de Actividad Guardar Archivos Adjuntos	74
Ilustración 26 Diagrama Relacional	76
Ilustración 27 Ingresar Modulo Web.....	77
Ilustración 28 Bienvenidos	78
Ilustración 29 Reservar horas	78
Ilustración 30 listar horas	79
Ilustración 31 ingresar	80
Ilustración 32 Bienvenida Móvil	81
Ilustración 33 Selección de Especialidad	82
Ilustración 34 Selección de día.....	83
Ilustración 35 Resultado Final.....	84
Ilustración 36 Respuesta web service	86

Lista de Tablas

Tabla 1 Plan de trabajo tentativo	12
Tabla 2 Objetivo del sistema 1	51
Tabla 3 Objetivo del sistema 2	51
Tabla 4 Objetivo del sistema 3	52
Tabla 5 Objetivo del sistema 4	52
Tabla 6 Objetivo del sistema 4	52
Tabla 7 Objetivo del sistema 6	52
Tabla 8 Objetivo del sistema 7	52
Tabla 9 reglas de negocio	53
Tabla 10 Información sobre el administrador general.....	53
Tabla 11 Información sobre el personal médico	54
Tabla 12 Información sobre pacientes.....	54
Tabla 13 Información sobre los empleados administrativos	55
Tabla 14 Información sobre las peticiones de exámenes	55
Tabla 15 Información sobre las recetas médicas asignadas a los pacientes	55
Tabla 16 Información sobre las citas médicas.....	56
Tabla 17 Información sobre los antecedentes del paciente	56
Tabla 18 Unicidad de usuarios	56
Tabla 19 Unicidad de las citas.....	56
Tabla 20 Administrador General	57
Tabla 21 Administrador de citas.....	57
Tabla 22 Personal Médico	57
Tabla 23 Médico General	57
Tabla 24 Paciente.....	57
Tabla 25 Consultar agenda médico desde el móvil	58
Tabla 26 Sincronizar Agenda con el servidor	58
Tabla 27 Ver detalles de cita en la agenda Médica móvil.....	58
Tabla 28 Ver detalles de cita en la agenda Médica Web.....	58
Tabla 29 Consultar agenda desde Web.....	59
Tabla 30 Realizar consulta Médica General desde el móvil	59
Tabla 31 Realizar petición de exámenes paraclínicos desde el móvil.....	59
Tabla 32 Realizar consulta de Historia clínica de un paciente desde el móvil.....	59
Tabla 33 Autenticar usuarios.....	59
Tabla 34 Registrar personal médico en el sistema	60
Tabla 35 Cambiar contraseña	60
Tabla 36 Anexar multimedia	60
Tabla 37 Administrar citas médicas	60
Tabla 38 Generar formula médica desde el móvil.....	61
Tabla 39 Generar orden de remisión a especialista desde el móvil.....	61
Tabla 40 Registrar Evento médico desde el móvil.....	61
Tabla 41 Generar Log de cada suceso en el servidor	61
Tabla 42 Actualizar Historia clínica del paciente desde el móvil	62
Tabla 43 Usar SOAP como protocolo para comunicación.....	62
Tabla 44 Usar Android	62

Tabla 45 La información del sistema se debe mantener en un servidor.....	62
Tabla 46 La información del sistema se debe manejar con el DBMS ORACLE.....	62
Tabla 47 La comunicación debe hacerse por medio de web services	63
Tabla 48 El sistema web se debe hacer usando J2EE.....	63
Tabla 49 El sistema deberá ser probado sobre equipos reales.....	63
Tabla 50 El sistema deberá ser sometido a pruebas	63
Tabla 51 Caso de prueba sistema web.....	85
Tabla 52 Caso de prueba sistema móvil 1	86
Tabla 53 Caso de prueba sistema móvil 2	87

1. Capítulo primero: Descripción del proyecto

1.1. Introducción

En Chile, para Febrero del 2012, existían 22 millones de usuarios de tecnología móvil, lo que indica una cobertura cerca del 129% de la población.

No es raro el hecho que la tecnología móvil haya tenido gran acogida dentro de la población chilena, debido a que se encuentra al alcance económico de casi cualquier persona, además, estos dispositivos móviles llegan a tener una velocidad de procesamiento comparable a los computadores de hasta hace algunos años atrás[2].

Los dispositivos móviles de hoy en día cuentan con la gran ventaja de la portabilidad, pero además, entre sus características, encontramos aplicaciones para la georeferenciación, e integración de dispositivos como cámara, reloj, agenda y reproductores de audio y video en el mismo aparato.

Teniendo en cuenta las numerosas ventajas de la tecnología móvil actual, se pretendió aprovechar algunas de estas características a favor del proyecto de telemedicina. El proyecto ofrece al usuario una solución innovadora para acortar distancias y reducir tiempos de espera al solicitar citas médicas en algún centro de atención hospitalario, mediante el uso de aplicaciones móviles que resulten ser muy intuitivas y fáciles de usar.

Los centros médicos se encuentran en las grandes ciudades y algunas cabeceras municipales, por tanto no toda la población tiene acceso a la medicina especializada, y muchos usuarios que si tienen acceso no logran acceder a estos centros médicos por la naturaleza de su enfermedad.

En vista de la problemática planteada se buscó crear un prototipo de telemedicina móvil para asistencia médica domiciliaria y remota a través de servicios web y tecnologías móviles, que abarque desde un acceso a historias clínicas y antecedentes vía web y móvil hasta la prestación de una asesoría remota entre médicos.

1.2. Objetivos

1.2.1. Objetivo General

Analizar, diseñar y desarrollar un prototipo de sistema de información en telemedicina con tecnologías móviles para la asistencia médica domiciliaria y remota, usando servicios web para la comunicación con un módulo web central.

1.2.2. Objetivo Especifico

- Establecer los requerimientos funcionales, no funcionales y de información de médicos necesarios en el ámbito del desarrollo del prototipo de sistema de información.
- Diseñar la arquitectura del prototipo con base a los requerimientos anteriormente establecidos y el modelo de base de datos acorde con la información recopilada.
- Diseñar la arquitectura de la red de comunicación del prototipo.
- Diseñar la interfaz de la aplicación acorde al análisis previo.
- Desarrollar los módulos web y móvil del prototipo de sistema de información.

1.3. Descripción del Problema

Teniendo en cuenta la masividad con la que cuentan las tecnologías móviles por estos días y la real posibilidad de implementar soluciones prácticas e innovadoras en áreas como la salud, las aplicaciones de software orientadas a la telemedicina podrían llegar a ser una indispensable herramienta tecnológica en favor del bienestar humano a nivel mundial; se hace necesario expandir este tipo de tecnologías y servicios para la comunidad.

El uso de dispositivos móviles está creciendo de manera exponencial en el mundo, favorecidos por el proceso de globalización en el que nos encontramos insertos, ofreciendo la oportunidad a desarrolladores de software y a profesionales del área informática, de trabajar en el área de los servicios y la salud y con ello brindar un aporte a la humanidad. Como enfatiza el presidente de las Naciones Unidas Timothy E. Wirth, el poder de estas tecnologías para mejorar la condición humana no puede ser desestimado: *“Modern telecommunications, and the creative use of it, has the power to change lives and help.... solve some of the world’s biggest challenges.”*[1]

Sin embargo, en Chile la telemedicina no es muy difundida, desaprovechando todos los beneficios que puede prestar a la población. Desde el punto de vista de la ingeniería informática se podría aportar conocimiento y herramientas para que estas tecnologías sean más difundidas en el país, actualmente existen aplicaciones web destinadas a prestar servicios en el área de la salud, pero carecen de la personalización y portabilidad que el proyecto de telemedicina pretende ofrecer con los módulos desarrollados para dispositivos móviles. Esta carencia de implementación de este tipo de servicios, ofrece a la comunidad y a los desarrolladores de software una oportunidad única para llevar a cabo ideas novedosas en el campo de la salud en pos del beneficio de la atención de personalizada y de calidad para los pacientes.

En Chile la atención médica especializada y la mayoría de centros hospitalarios se concentran en las zonas urbanas, generando falta de cobertura en servicios de salud y dificultando el acceso de las personas residentes en áreas rurales y regiones extremas. Asumiendo que más del 20% de la población se encuentra ubicada en dichas áreas, se podría utilizar la telemedicina como herramienta para permitir el acceso de servicios de salud, a este porcentaje de la población [3].

De igual manera, en muchas ocasiones los procesos de salud que implican atención inmediata, se ven limitados por la falta de eficiencia, información y comunicación entre el personal médico. Ejemplos claros de esta problemática se evidencian en situaciones de emergencias que implican el uso de ambulancias, en casos de personas cuya enfermedad impide su traslado físico al centro de atención.

Debido a la posibilidad de llevar la tecnología en un bolsillo, y teniendo la posibilidad de estar interconectado en cualquier momento es que nace la idea de digitalizar estos procesos y lograr un intercambio de información de manera eficiente y duradera.

1.4. Plan de trabajo

Se investigaron distintas propuestas referentes a las tecnologías asociadas a los dispositivos móviles y se tomó la decisión de usar aquella que permitiera entregar al usuario una aplicación fácil e intuitiva de usar.

El plan de trabajo fue el siguiente:

Fase	Actividades	Fecha inicio	Fecha Fin
Proyecto 2	Planteamiento del problema	07-07-2012	08-07-2012
	Justificación	08-07-2012	09-07-2012
	Limitaciones	09-07-2012	10-07-2012
	Objetivo General y Especifico	10-07-2012	11-07-2012
	Contexto del Sistema	11-04-2012	19-04-2012
	Manejo de la Información Clínica en Telemedicina	13-07-2012	13-07-2012
	Modelo de Comunicación y Conectividad	14-07-2012	14-07-2012
	Contexto tecnológico y metodologías de desarrollo	15-07-2012	15-07-2012
	Definición de requerimientos funcionales y no funcionales	16-07-2012	16-07-2012
	Casos de Uso	16-07-2012	19-07-2012
	Modelo de Clases	16-07-2012	19-07-2012
	Modelos de Secuencia	20-07-2012	21-07-2012
	Modelo Relacional y diccionario de Datos	20-07-2012	22-07-2012
	Desarrollo del Primer Prototipo Web	22-07-2012	25-07-2012
	Desarrollo del Primer Prototipo Móvil	22-07-2012	25-07-2012
Construcción del Informe	25-07-2012	29-07-2012	

Tabla 1 Plan de trabajo tentativo

2. Capitulo Segundo: Investigación preliminar

2.1 Estado del Arte

La Telemedicina se refiere a la medicina basada en la ubicuidad, en usar la movilidad a favor de la medicina y llevarla a donde esta es requerida.

La integración de las ciencias médicas con el desarrollo de las telecomunicaciones y la informática y su aplicación en las diferentes actividades del sector de la salud, hace posible conceptualizar el término de Telemedicina como la distribución de servicios de salud, en el que la distancia es un factor crítico, donde los profesionales de la salud usan información y tecnología de comunicaciones para el intercambio de información válida para el diagnóstico, tratamiento y prevención de enfermedades o daños, investigación y evaluación; y para la educación continuada de los proveedores de salud pública, todo ello en interés del desarrollo de la salud del individuo y su comunidad [5].

La estrepitosa evolución digital hace que Profesionales Médicos, Ingenieros, personal de enfermería, pacientes, usuarios y asistentes de la Salud incorporen en sus cotidianos quehaceres la Telemedicina y la Informática Médica.

La telemedicina acompaña la entrega de servicios de salud, incluyendo diagnósticos clínicos, cuidados directos, educación del paciente y el movimiento de la información médica en formato electrónico, en tiempo real o no, mediante procesos interactivos entre profesionales, pacientes y equipamiento.

La monitorización, capacidades de diagnóstico remoto y otros elementos de la telemedicina requieren de herramientas y tecnologías especialmente diseñadas para tales fines. Las Telecomunicaciones entran en acción para proveer el sostén, evolución y futuro de la herramienta. La telemedicina ofrece la oportunidad de incrementar la eficacia clínica de los profesionales y mejorar los tiempos de diagnóstico [10].

2.2 Reseña Histórica

Desde que a principios del presente siglo se pusieron en marcha las modernas vías de la telecomunicación, cuyos representantes genuinos fueron el telégrafo, el teléfono y más tarde la radio y la televisión, la Medicina fue consciente de que una nueva era de investigación y progreso quedaba abierto.

Fue Norman Jefferis Holter quién a mitad de este siglo, junto a Gengerelli y Glasscock, inició la exploración de los parámetros biológicos "sin tocar al hombre", y cuyas experiencias culminaron felizmente, cuando consiguieron recibir vía radio el electrocardiograma de personas que deambulaban por la calle a considerable distancia de la estación receptora.

En 1959, los avances tecnológicos, permitieron por vez primera la transmisión de imágenes radiológicas a través de las líneas telefónicas. Se inauguraba así, una nueva etapa de avances en el área de la salud, en la que un medio audiovisual, acercaba a profesionales alejados físicamente, pero permitiéndoles una discusión interactiva que mejorara el enfoque diagnóstico y coordinándose entre ellos la actitud terapéutica a seguir sobre pacientes igualmente distantes.

En los últimos años, la Telemedicina ha estado adquiriendo un valor singular. Ha permitido que delicadas intervenciones médico-quirúrgicas hayan podido ser dirigidas por especialistas desde puntos tan alejados como Alemania o Estados Unidos con resultados plenamente satisfactorios.

Es precisamente en Estados Unidos donde 13 agencias federales y más de 40 estados, han desarrollado esta infraestructura tecnológica, y trabajan con esta metodología médica a pleno rendimiento y satisfacción. Universidades y grandes centros médicos, como la Clínica Mayo, el Hospital General de Massachusetts, el Emory University ó el Southern California, por citar sólo algunos, invierten sumas importantes de sus presupuestos para el avance de la Telemedicina. De igual modo, grandes empresas privadas como AT&T y V-TEL, han invertido en los últimos años más de 1000 millones de dólares en el desarrollo de estas tecnologías de comunicación.

La medicina militar de campaña, fue pionera en el uso de la Telemedicina, y hoy en día, los modernos ejércitos disponen de estos sistemas como parte básica de su infraestructura militar y sanitaria.

Uno de los aspectos más interesantes y en los que la Telemedicina está cobrando una mayor preponderancia, radica en la obtención de una "segunda opinión" a cargo de un experto en cada materia concreta. En tal sentido, la medicina rural, la medicina deportiva, la medicina de emergencia (catástrofes, terremotos, inundaciones, etc.) o simplemente la medicina habitual que requiere de expertos en casos concretos, están encontrado a través de este sistema, una excelente vía de comunicación y de trabajo en equipo [4].

2.3 Bases de la Telemedicina (definición, objetivos)

Telemedicina es el empleo de las tecnologías de información y comunicaciones para brindar información y asistencia médica a quien lo requiera en sitios distantes. Básicamente consiste en la transferencia de información médica a través de redes de comunicación.

Esta información o datos médicos son trasladados, recibidos y tratados en una central adecuada para tal fin.

En otra definición más descriptiva se pueden mencionar los tres principios que forman la columna vertebral de esta modalidad y que debemos tener presente en todo momento:

- *Telemedicina es medicina*: dejando de lado la tecnología, estamos ante la presencia de un servicio de atención médica hacia la ciudadanía cubriendo prevención, curación y rehabilitación. Sumando todo lo referente a la formación médica [12]
- *Telemedicina es servicio a la sociedad*: esto magnifica la importancia de la tecnología y sus adelantos para ponerla a disposición de la gente, para lograr un equilibrio equitativo y eficaz en los servicios que le competen al área de la salud [4].
- *Telemedicina es práctica a distancia*: esta es su esencia y su distintiva cualidad. Es importante conocer que la calidad y seguridad de la atención médica están garantizadas con las nuevas tecnologías, es obvio que se modifican los escenarios y la percepción de la realidad, pero una vez que los profesionales y usuarios se habitúen los resultados pueden igualarse o superarse en comparación con la medicina clásica.

2.3.1 Objetivos de la Telemedicina

2.3.1.1 Asistencia Médica

Brindar servicio de Asistencia Médica a quien lo requiera, independientemente del momento y ubicación geográfica del paciente y el centro asistencial.

Aquí se puede dividir la telemedicina en 3 áreas:

- *Como ayuda en el proceso de tomar decisiones*: la telemedicina incluye áreas tales como los sistemas expertos a distancia, que contribuyen al diagnóstico del paciente o el uso de bases de datos on-line. Este es el uso más antiguo de la telemedicina.
- *Control a distancia*: consiste en la transmisión de información del paciente (ECG, radiografías, datos clínicos, bioquímicos, etc.). Este es el uso más actual y en el que se centra el proyecto.
- *Colaboración en tiempo real para el manejo del paciente a distancia*: permite que un médico pueda observar y discutir los síntomas de un paciente que está siendo asistido por otro médico a la distancia. Este es el uso del futuro, cuando se normalicen ciertos aspectos tales como responsabilidad, acreditación, formas de pago, etc.

2.3.1.2 Educación y Difusión

Educar y difundir información médica a público en general, estudiantes y profesionales. La Telemedicina, es su faz educativa, constituye un recurso formidable para la enseñanza y el aprendizaje a distancia.

- Permite que varios médicos y/o estudiantes adquieran simultáneamente los conocimientos.
- Elimina los costos que involucra el traslado físico de los educandos a los centros de capacitación de nivel internacional.
- Adecuadamente programada, no interfiere con la actividad habitual de los participantes.

2.3.2 Aplicaciones de la Telemedicina

A continuación se mencionan algunas de las aplicaciones clasificándolas de acuerdo a su orientación fundamental y el tipo de tecnología asociada.

- **Diagnóstico por imagen:** tele-radiología, tele-dermatología, tele-oftalmología, tele-patología, tele-citología, tele-endoscopía.
- **Telemetría:** tele-cardiología, tele-oftalmología, tele-neurología, tele-emergencia.
- **Tele robótica:** tele-cirugía, tele-endoscopía, endoscopía virtual.
- **Control de enfermedad:** tele-oncología, tele-emergencia, tele-ortopedia, tele-pediatría, tele-psiquiatría.
- **Ingeniería médica:** tele-bioingeniería, tele-ingeniería clínica
- **Docencia:** tele-supervisión, tele-soporte domiciliario, tele-rehabilitación.
- **Gestión:** tele-epidemiología, tele-salud pública, tele-radio seguridad, tele-bioseguridad.

2.4 Telemedicina y su Aceptación

La disponibilidad de una infraestructura eficiente y moderna de telecomunicaciones acentúa el crecimiento real de las aplicaciones de telemedicina, como la tele-educación entre otros. Debido a los límites de infraestructura, los países en desarrollo optan por almacenar el concepto de telemedicina. Sin embargo, algunos programas de telemedicina en países en desarrollo con conectividad son bastante comparables con algunos del mundo desarrollado. La telemedicina está ganando campo en los países en desarrollo e incluso muchas clínicas en estos países han empezado a practicar la telemedicina fuera de sus necesidades profesionales. En una forma muy simple se están logrando los objetivos de la telemedicina al intercambiar información clínica y comentarios a través de emails. Algunos países en desarrollo como India, Nepal y Bangladesh, etc. se han abierto a la telemedicina para lograr algunos objetivos propuestos por sus sistemas de salud.

En un estudio hecho en 1999 en Australia, cuatro centros médicos escogieron un grupo de 31 pacientes a quienes les fue preguntado que midieran su presión sanguínea dos veces por semana y que enviaran el resultado de sus mediciones, también si efectuaban algún cambio en la medicación y los posibles efectos secundarios. La información recopilada fue transferida a un computador clínico, y los reportes fueron generados para cada médico de cada paciente. Aquellos que no enviaban las mediciones a tiempo eran contactados telefónicamente para que efectuaran el procedimiento. Los 31 pacientes tuvieron en promedio 1.5 visitas al médico durante el año de estudio lo que significa una reducción de lo usual en el tipo de pacientes escogidos (2.7 visitas al año). De acuerdo con este estudio la prescripción de los doctores pueden también ser influenciada por la evidencia de la toma de presión por cada paciente.

Con base en los anteriores casos de éxito, y la importancia que a nivel tecnológico y académico puede traer a la sociedad chilena un desarrollo de telemedicina móvil, se consideró factible e importante llevar a cabo este proyecto usando tecnologías móviles y de código abierto que son más accesibles económicamente.

2.5 Manejo de la información clínica en telemedicina

2.5.1 Historia Clínica

Es un documento privado con características: legales, éticas, docentes, estadísticas, médicas o clínicas. Los datos deben ser consignados en términos adecuados y en forma lógica y ordenada o secuencial.

Uno de los módulos desarrollados, permite registrar documentos como: Exámenes, Licencias y Recetas y almacenarlos en la Base de Datos central, para su posterior validación legal, además el modulo web, permitirá visualizar e imprimir dichos archivos adjuntos.

La historia clínica es una narración escrita, clara, precisa, detallada y ordenada de todos los datos y conocimientos, remotas y actuales, personales y familiares, relativos al paciente, que sirven como base para el conocimiento de la enfermedad.

Es el conjunto de datos o la información que aporta el interrogatorio. Es la forma en que se inicia la relación profesional – enfermo. Consta de tres partes:

- **Identificación del paciente:** En esta parte se identifica al paciente en cuanto a su nombre y edad. Cabe la posibilidad de agregar más información como teléfono de su casa, a quién contactar en caso de necesidad, qué previsión tiene, o qué actividad desarrolla.

Al momento de comenzar a escribir la historia clínica, se anota:

- Fecha y hora.
- Nombre completo del paciente.
- Edad.
- Eventualmente, se agrega:
 - Teléfono o dirección.
 - A quién avisar en caso de necesidad.
 - Actividad que desempeña.

En pacientes que no son capaces de aportar su historia, conviene señalar la fuente de dónde provino la información (p.ej.: la mamá, algún familiar con el que vive, un testigo).

- **Problema principal o motivo de consulta:** Es una mención muy corta del motivo por el que consulta el paciente.

- **Enfermedad Actual:** Se precisa la enfermedad que padece el paciente al momento de efectuar consultar.

Se preguntan los aspectos individuales y familiares pasados del paciente. Se divide en tres antecedentes personales, antecedentes familiares y revisión por sistemas.

• **Antecedentes:** En esta parte se mencionan distintos antecedentes ordenados según su naturaleza. Estas secciones son:

- Antecedentes personales (médicos, quirúrgicos, traumatismos).
- Hábitos.
- Antecedentes sobre uso de medicamentos.
- Alergias.
- Antecedentes sociales y personales.
- Antecedentes familiares.
- Inmunizaciones.

2.5.2 Sistema de Salud

El sistema de salud chileno es un sistema mixto, liderado por el Ministerio de Salud, el que tiene por tarea el diseño de políticas y programas, la coordinación de las entidades del área, la supervisión, la evaluación y el control de las políticas de salud.

La base productiva del sistema de salud pública está conformada por el SNSS. Para llevar a cabo su tarea, cada Servicio de Salud posee y opera varios hospitales de distintos niveles de complejidad y centros de atención abierta, pudiendo establecer contratos con proveedores privados para servir a ciertas zonas o para tipos específicos de prestación. La atención de salud primaria está a cargo de Centros de Atención Primaria.

En el sistema público existe un seguro social de salud administrado por FONASA. El seguro opera sobre la base de un esquema de reparto (se financia con la prima única de 7% de la renta imponible de sus asegurados y con recursos provenientes de impuestos generales de la nación). Los beneficios que este esquema entrega son los mismos para todos los afiliados, independientemente del monto de la prima cancelada y del tamaño del grupo familiar cubierto.

El sistema de salud privado está conformado por las ISAPRE y por productores de salud particulares. Las ISAPRE operan como un sistema de seguros de salud basado en contratos individuales pactados con los asegurados, en el que los beneficios otorgados dependen directamente del monto de la prima cancelada. Los proveedores privados de salud son los hospitales, clínicas y profesionales independientes que atienden tanto a los asegurados de las ISAPRE como a los cotizantes del sistema público.

Los trabajadores activos y pasivos tienen la obligación de cotizar el 7% de su renta imponible al sistema de salud, con un tope de UF 4.2 mensual. Este pago puede ser realizado a FONASA o a una ISAPRE. La afiliación a una ISAPRE requiere de una prima determinada por cada ISAPRE, la cual depende del tipo de seguro que se adquiera y de las características del afiliado. El trabajador puede cancelar primas adicionales al 7% con el fin de obtener beneficios adicionales.

Los indigentes y no cotizantes forman parte del FONASA, aunque están sujetos a un trato especial. Los cotizantes adscritos a FONASA pueden elegir entre dos modalidades de atención: la modalidad institucional (atención cerrada) y la de libre elección (atención abierta).

En la primera, los cotizantes reciben las prestaciones en hospitales o centros de atención primaria. En la segunda, los cotizantes reciben sus atenciones de prestadores privados adscritos a esta modalidad. La modalidad institucional requiere un copago que varía de acuerdo al nivel de ingreso de la persona, quedando exentas de este copago las personas cuyo ingreso es inferior a un nivel mínimo establecido. La modalidad de libre elección requiere un copago que depende del nivel en el cual se ha inscrito el prestador de salud. Los beneficiarios indigentes y no cotizantes del sistema público están excluidos de esta modalidad.

2.5.2.1 Principales Agentes del Sistema

El sistema de salud funciona a través de cuatro agentes: el Ministerio de Salud, el Instituto de Salud Pública (Ispch), el Fondo Nacional de Salud (Fonasa) y las Empresas de Administración de Salud (ISAPRE).

2.5.2.1.1 Ministerio de Salud

Al Ministerio de Salud le corresponde formular y fijar, de acuerdo con las directivas que señale el Supremo Gobierno, las políticas de salud y dictar las normas y planes generales para el Sistema.

Las funciones del Ministerio de Salud:

- Dirigir y orientar todas las actividades del Estado relativas al Sistema, en conformidad con las políticas fijadas.
- Dictar normas generales sobre materias técnicas, administrativas y financieras a las que deberán ceñirse los organismos y entidades del Sistema para ejecutar actividades de promoción o fomento, protección y recuperación de la salud y de rehabilitación de las personas enfermas.
- Formular los planes y programas generales del Sistema, en concordancia con la política general del Gobierno.
- Coordinar y controlar la actividad de los organismos del Sistema y propender, en la forma autorizada por la ley, al desarrollo de las actividades de salud por organismos y personas que no integran ese Sistema.
- Supervisar, controlar y evaluar el cumplimiento de las políticas, planes y normas de salud.
- Relacionarse con personas, organismos y entidades públicas y privadas que no pertenezcan al Sistema o al Sector Salud.
- Fijar las políticas, dictar las normas, aprobar los planes y programas generales y evaluar las acciones respecto de las construcciones, transformaciones y reparaciones de edificios destinados a establecimientos hospitalarios de servicios de salud.
- Cumplir las demás funciones que le asignen las leyes y reglamentos.
- El Reglamento del Ministerio de Salud establece que, para el cumplimiento de las políticas, planes y normas que imparta esa Secretaría de Estado, las personas naturales o jurídicas, públicas o privadas que laboren en salud

coordinadamente y dentro de los marcos fijados por el Ministerio constituyen el Sistema Nacional de Salud.

- Los servicios públicos que dependen del Ministerio y forman parte del Sistema son: los Servicios de Salud, el Fondo Nacional del Sistema de Salud, el Instituto de Salud Pública, la Central de Abastecimiento del Sistema Nacional de Servicios de Salud y la Superintendencia de Salud.
- Las personas, instituciones y demás entidades privadas gozarán de libre iniciativa para realizar acciones de salud, en la forma y condiciones que determine la ley.

2.5.2.1.2 Instituto de Salud Pública

El Instituto de Salud Pública de Chile (ISP) es un servicio público funcionalmente descentralizado, que posee autonomía de gestión y está dotado de personalidad jurídica y de patrimonio propio, dependiendo del Ministerio de Salud para la aprobación de sus políticas, normas y planes generales de actividades, así como en la supervisión de su ejecución.

La misión del ISP es contribuir al mejoramiento de la salud de la población, garantizando la calidad de bienes y servicios, a través del fortalecimiento de la referencia, la fiscalización y la normalización.

2.5.2.1.3 Fondo Nacional de Salud

El Fondo Nacional de Salud, FONASA, es el organismo público encargado de otorgar cobertura de atención, tanto a las personas que cotizan el 7% de sus ingresos mensuales para la salud en FONASA, como a aquellas que, por carecer de recursos propios, financia el Estado a través de un aporte fiscal directo.

Asimismo FONASA da cobertura de salud a más de 10 millones de beneficiarios, sin exclusión alguna de edad, sexo, nivel de ingresos, número de cargas familiares legales y enfermedades preexistentes, bonificando total o parcialmente las prestaciones de salud que les son otorgadas por profesionales e instituciones del sector público y privado.

2.5.2.1.4 Empresas de Administración de Salud

Las ISAPRE son instituciones privadas que captan la cotización obligatoria de los trabajadores que libre e individualmente han optado.

Las prestaciones de salud se entregan directamente o a través del financiamiento de las mismas mediante la contratación de servicios médicos financiados por las ISAPRE.

El Sistema ISAPRE ha contribuido al desarrollo global del sector salud en Chile, descongestionando al sector público y permitiéndole a éste, por tanto, centrar sus esfuerzos en las personas de más bajos recursos.

Con la responsabilidad de otorgar atención a casi cuatro millones de beneficiarios y apoyados en la competencia de libre de mercado las ISAPRE han logrado perfeccionar sus servicios y otorgar más y mejores prestaciones de salud. En este contexto, han desarrollado productos de bajo costo como planes colectivos, enfermedades catastróficas y tercera edad. Actualmente, las ISAPRE reúnen a más de 3,4 millones de beneficiarios y, entre ellos, se encuentran más de 140 mil personas con más de 60 años de edad y con ingresos promedios que hoy van desde los 200 mil pesos hacia arriba.

2.5.2.1.5 Superintendencia de Salud

La Superintendencia de Salud garantiza el cumplimiento de la ley y genera una regulación y fiscalización efectiva, que contribuya a mejorar el desempeño del Sistema de Salud chileno, velando por el cumplimiento de los derechos y garantías de las personas, mediante el desarrollo y la entrega de servicios de excelencia. Colabora en la satisfacción de las necesidades de salud de todos los chilenos, aportando y promoviendo soluciones a los problemas sanitarios de modo ágil, creativo e innovador.

Está orientada a los usuarios y entes regulados, a quienes brinda servicios con altos estándares de calidad. Creada en 1990, a través de la Ley N° 18.933, la Superintendencia de Instituciones de Salud Previsional (SISP) es el organismo del Estado que se encarga de fiscalizar y controlar a las ISAPRE.

3 Capítulo Tercero: Solución propuesta

3.1 Modelo de comunicación y conectividad

3.1.1 Arquitectura

La aplicación de Telemedicina, es un prototipo de Sistema de información para el manejo de información médica a nivel de consultas, citas, alarmas, remisiones e historias clínicas, desde dispositivos móviles (Celulares / Tablets) haciendo uso de los estándares internacionales para el intercambio de información médica, además de servir para el manejo de pacientes, personal médico y los usuarios administradores del sistema.

3.1.1.1 Arquitectura Lógica

La Arquitectura Lógica es la manera en que se organizan e integran los componentes lógicos de una solución, la elegida para el desarrollo de este proyecto es la programación por capas la cual es un estilo de programación en el que el objetivo principal es la separación de la lógica de negocios, de la lógica de diseño. Como ventaja principal de este estilo, es el poder llevar a cabo el desarrollo en varios niveles, en caso de que se presente algún cambio, sólo se ataca al nivel requerido sin tener que revisar la totalidad del código.

En el diseño se suelen usar las arquitecturas multinivel o programación por capas por su alta escalabilidad debido a que pueden ampliarse con facilidad, en caso de que las necesidades aumenten o cambien.

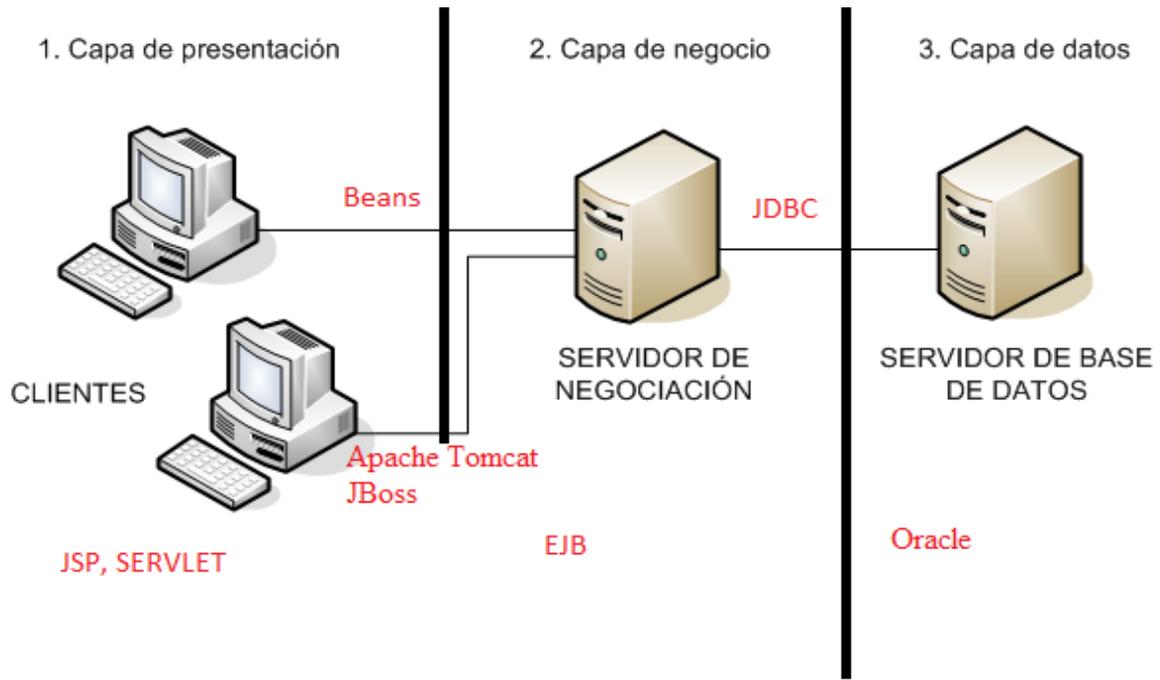


Ilustración 1 Arquitectura del sistema

3.1.1.1.1 Nivel Superior Móvil - Presentación y Lógica

Esta capa está desarrollada en Android, para ser usada en dispositivos móviles. Tiene los servicios de:

- Consultar la agenda medica
- Consultar la información básica del paciente
- Consultar e insertar información pertinente a la historia clínica de un paciente, esta información es desde texto plano hasta fotografías y archivos de audio y video.
- Enviar toda la información médica a través de mensajes SOAP, pudiéndose comunicar con la Base de Datos Central.

3.1.1.1.2 Nivel Superior Web - Presentación y Lógica

Esta capa está desarrollada en JSP/XHTML + JavaScript para la parte de presentación y en SERVLET para la parte de la lógica.

Tiene los servicios de:

- Administrar usuarios (creación / eliminación /cambio de datos)
- Administrar pacientes (Afiliar/desactivar)
- Administrar Agendas (Horarios de citas para el Personal Médico)
- Crear entidades Administradoras y entidades prestadoras además de sus respectivas sedes.

3.1.1.1.3 Nivel Inferior - Lógica y Datos

Esta parte está desarrollada en EJB para la parte de la lógica y ORACLE para la parte de los datos. En esta capa no hay funciones a nivel de interacción con el usuario, sin embargo es la encargada de la consistencia de los datos, según los modelos planteados.

3.1.1.2 Comunicación

La comunicación entre el nivel superior tanto Móvil como Web y el nivel inferior se hace a través de servicios web SOAP, para algunos de los servicios en especial desde el dispositivo móvil.

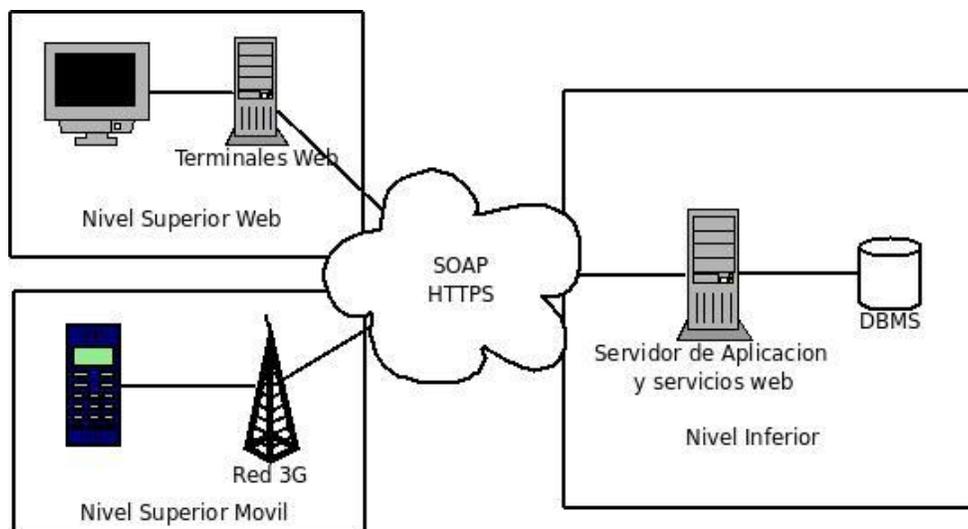


Ilustración 2 Arquitectura del sistema

3.1.1.2.1 SOAP

SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por David Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

SOAP es importante para el desarrollo de aplicaciones por permitir la comunicación a través de internet de diversos programas.

La mejor forma de comunicarse entre aplicaciones es a través de HTTP, porque HTTP esta soportada por todos los browsers y servidores de Internet. SOAP fue creado para esto.

SOAP provee una forma de comunicación entre aplicaciones corriendo en diferentes sistemas operativos con diferentes tecnologías y diferentes lenguajes de programación[5].

3.1.1.2.1.1 Partes de un Mensaje SOAP

Un mensaje SOAP es un documento XML común y corriente que contiene los siguientes elementos:

- Una envoltura (<soap: Envelope>) que identifica el documento XML como un mensaje SOAP
- Una cabecera (<soap: Header>) que contiene información de cabecera
- Una cuerpo (<soap: Body>) que contiene información de llamada y respuesta
- Una infracción (<soap: Fault>) que contiene los errores y la información de estado

3.1.1.2.2 Web Services

Un servicio web (en inglés Web Service) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes como Internet.

La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web [6].

3.1.1.2.2.1 Pila de protocolos para Servicios Web

La Pila de protocolos para Servicios Web es una colección de protocolos para redes de Computadores que son utilizados para definir, localizar, implementar y hacer que un Servicio Web interactúe con otro. La Pila de Protocolos para servicios está comprendida principalmente por cuatro áreas [7]:

- **Servicio de Transporte:** responsable del transporte de mensajes entre las Aplicaciones de red y los protocolos en los cuales se incluyen protocolos tales como HTTP, SMTP, FTP, así como también el más reciente Blocks Extensible Exchange Protocol (BEEP).
- **Mensajería XML:** responsable por la codificación de mensajes en un formato común XML así que ellos puedan ser entendidos en cualquier extremo de una conexión de red. Actualmente, esta área incluye protocolos tales como XML-RPC, SOAP y REST.
- **Descripción del Servicio:** usado para describir la interfaz pública de un Servicio Web específico. El formato de interfaz Web Services Description Language - WSDL es típicamente usado para este propósito.
- **Descubrimiento de servicios:** centraliza servicios en un registro común tal que los servicios Web de la red puedan publicar su localización y descripción, y hace que sea fácil descubrir que servicios están disponibles en la red.

Actualmente, la API Universal Description Discovery and Integration - UDDI se utiliza normalmente para el descubrimiento de servicios.

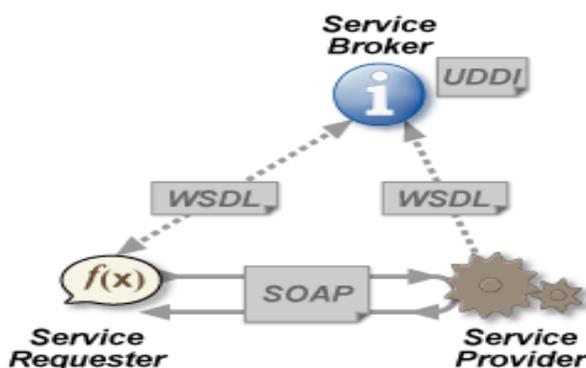


Ilustración 3 Arquitectura de WebServices

3.1.1.2.2 Ventajas de los Web Services

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C, por tanto no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones [7].

3.2 Contexto Tecnológico

3.2.1 Android

Android es una plataforma de software y un sistema operativo para dispositivos móviles basada en un kernel Linux, desarrollada por Google y más tarde por la Open Handset Alliance. Esta plataforma permite a los desarrolladores escribir código en Java que se ejecuten en móviles mediante las librerías Java desarrolladas por Google. También se pueden escribir aplicaciones en otros lenguajes, como por ejemplo C, para posteriormente ser compiladas en código nativo ARM y ejecutarlas, aunque este proceso de desarrollo no está soportado oficialmente por Google. La mayor parte de la plataforma de Android está disponible bajo licencia de software libre de Apache y otras licencias de código abierto.

3.2.1.1 Breve Historia

En Julio de 2005, Google adquirió Android Inc, una pequeña startup de California. En esos momentos, la compañía se dedicaba a la creación de software para teléfonos móviles. Una vez en Google, el equipo desarrolló un S.O. basado en Linux para dispositivos móviles. Más adelante, Google adaptó su buscador y sus aplicaciones para su uso en móviles.

En septiembre del 2007, Google tenía varias patentes de aplicaciones sobre el área de la telefonía móvil. El 5 de noviembre del mismo año, se anunció la fundación de la Open Handset Alliance al mismo tiempo que la creación de la plataforma Android. La Open Handset Alliance está formada por un consorcio de 34 compañías de hardware, software y telecomunicaciones, entre las cuales se incluyen Google, HTC, Intel y Motorola entre otras, dedicadas a investigar estándares abiertos para dispositivos móviles.

El primer teléfono en el mercado que posee Android es el T-Mobile G1 (también conocido como Dream), lanzado el día 22 de octubre de 2008 que viene con la versión Android 1.0 preinstalada. Este móvil es el resultado conjunto de T-Mobile, HTC y Google.

Por último, desde el 21 de octubre de 2008, Android está disponible como código abierto. Gracias a esto, cualquiera puede añadir extensiones, nuevas aplicaciones o reemplazar las existentes por otras dentro del dispositivo móvil [8]

3.2.1.2 Características de Android

- Amplia variedad de diseños (VGA, librerías de gráficos 2D y 3D...)
- Almacenamiento de datos en Base de Datos SQLite
- Conectividad (GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth y Wi-Fi)
- Mensajería (SMS y MMS)
- Navegador Web
- Máquina virtual de Java
- Las aplicaciones escritas en Java pueden ser compiladas y ejecutadas en la máquina virtual de Dalvik, la cual es una especializada máquina virtual diseñada para uso en dispositivos móviles.
- Soporte de formatos (MPEG-4, H.264, MP3, AAC, OGG, AMR, JPEG, PNG, GIF)
- Soporte para hardware adicional (cámaras de video, pantallas táctiles, GPS, acelerómetros...)
- Entorno de desarrollo (emulador, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para Eclipse IDE)

3.2.1.3 Arquitectura Android

Los componentes principales del sistema operativo de Android son:

- Aplicaciones: las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- Marco de trabajo de aplicaciones: los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- Bibliotecas: Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- Runtime de Android: Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La

Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".

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

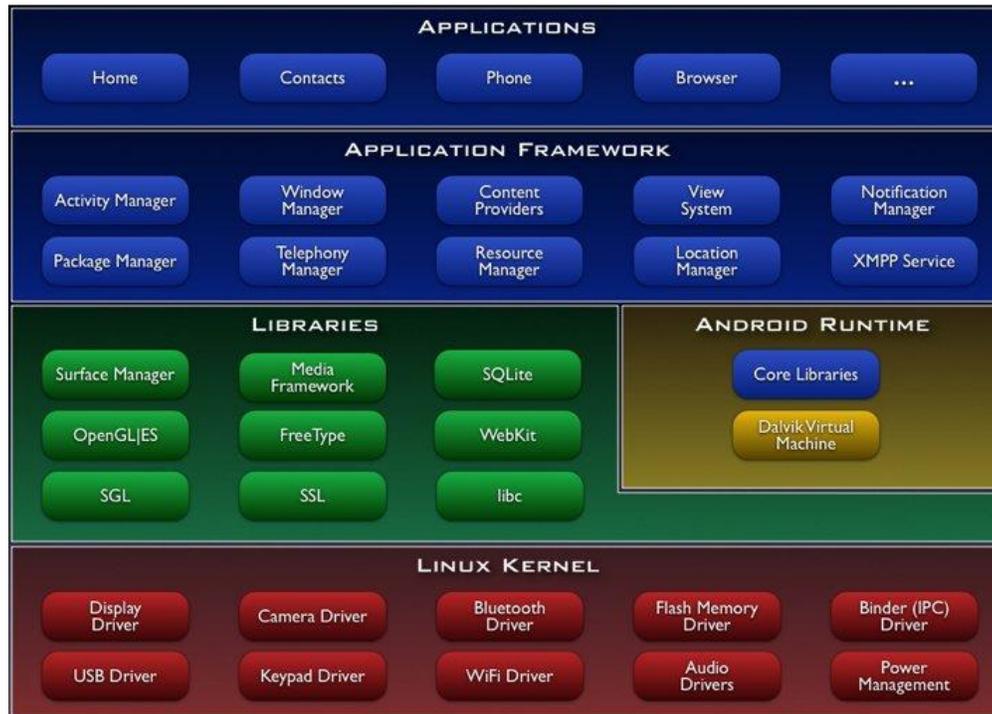


Ilustración 4 Arquitectura Android

3.2.2 SQL Lite (Base de Datos móvil)

SQL Lite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña ~275 kiB biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia del sistema de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB[8].

3.2.2.1 Características SQL Lite

La biblioteca implementa la mayor parte del estándar SQL-92, incluyendo transacciones de base de datos atómicas, consistencia de base de datos, aislamiento, y durabilidad (ACID), triggers y la mayor parte de las consultas complejas.

SQLite usa un sistema de tipos inusual. En lugar de asignar un tipo a una columna como en la mayor parte de los sistemas de bases de datos SQL, los tipos se asignan a los valores individuales. Por ejemplo, se puede insertar un string en una columna de tipo entero (a pesar de que SQLite tratará en primera instancia de convertir la cadena en un entero). Algunos usuarios consideran esto como una innovación que hace que la base de datos sea mucho más útil, sobre todo al ser utilizada desde un lenguaje de scripting de tipos dinámicos. Otros usuarios lo ven como un gran inconveniente, ya que la técnica no es portable a otras bases de datos SQL. SQLite no trataba de transformar los datos al tipo de la columna hasta la versión 3.

Varios procesos o hilos pueden acceder a la misma base de datos sin problemas. Varios accesos de lectura pueden ser servidos en paralelo. Un acceso de escritura sólo puede ser servido si no se está sirviendo ningún otro acceso concurrentemente. En caso contrario, el acceso de escritura falla devolviendo un código de error (o puede automáticamente reintentarse hasta que expira un timeout configurable). Esta situación de acceso concurrente podría cambiar cuando se está trabajando con tablas temporales. Sin embargo, podría producirse un deadlock debido al multithread [9].

3.2.3 J2EE

J2EE permite la creación de aplicaciones empresariales multicapa, basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchas de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja.

Para reducir costos, y aumentar la velocidad en el diseño y desarrollo de aplicaciones, J2EE provee componentes base, enfocados al diseño, al desarrollo, al montaje, y al despliegue de aplicaciones empresariales. La plataforma de J2EE ofrece un modelo distribuido de aplicaciones multicapas, componentes reutilizables, un modelo unificado de seguridad, control flexible de las transacciones, y soporte de servicios Web a través de un intercambio de datos integrados en Lenguaje de Marcas Extendible (XML) basados en estándares y protocolos abiertos.

3.2.3.1 Aplicaciones Distribuidas Multicapas

La plataforma de J2EE utiliza un modelo de aplicación distribuida multicapas, para las aplicaciones empresariales. La lógica de la aplicación, se divide en varios componentes de acuerdo a su función, y estos componentes se encuentran instalados en distintas máquinas, dependiendo de la magnitud del ambiente de multicapas, usado en la aplicación. La ilustración

5, muestra dos aplicaciones multicapas J2EE, dividida en una serie de partes, descritas en la siguiente lista:

- Los componentes del Cliente, funcionan en la máquina del cliente.
- Los componentes del Web funcionan en el servidor J2EE.
- Los componentes de Negocio, funcionan en el servidor J2EE.
- El sistema de información de la empresa (EIS) funciona en el servidor de EIS.

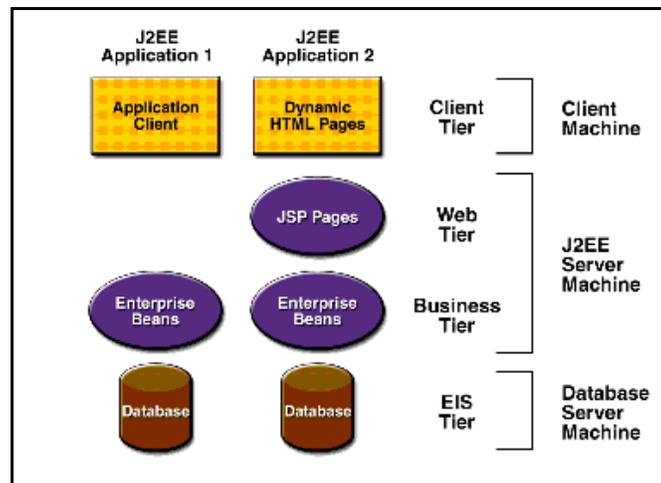


Ilustración 5 Aplicaciones Multicapas

Aunque una aplicación J2EE puede consistir en tres o cuatro partes, las aplicaciones multicapa son generalmente consideradas aplicaciones de tres partes, porque ellas se encuentran distribuidas sobre tres localizaciones: máquinas del cliente, la máquina del servidor J2EE, y las máquinas de la base de datos. Las aplicaciones de tres partes que funcionan de esta manera extienden el modelo estándar de dos partes, del cliente y del servidor, colocando una aplicación servidor multithreaded, entre la aplicación del cliente y el donde están almacenados los datos.

3.2.3.2 Componentes J2EE

Las aplicaciones J2EE, se forman de componentes. Un componente J2EE, es una unidad funcional autónoma de software, que está integrada en una aplicación J2EE, con sus relacionadas clases y archivos, y que se comunica con otros componentes. La especificación J2EE define los siguientes componentes J2EE:

- Aplicaciones clientes y applets, son los componentes que funcionan en el cliente.
- Java Servlet y JavaServer Pages (JSP), son los componentes Web que funcionan en el servidor.
- Los componentes JavaBeans de la empresa (EJB), son los componentes del negocio que funcionan en el servidor.

Los componentes de J2EE, se escriben en el lenguaje de programación Java y se compilan de la misma manera que cualquier programa del lenguaje. La diferencia entre los

componentes de J2EE y las clases "estándar" de Java; es que los componentes J2EE, están integrados en una aplicación J2EE, son verificados para que se integren bien y en conformidad con la especificación J2EE, y son desplegados en la fabricación de la aplicación, donde ellos funcionan.

3.2.3.3 Clientes J2EE

Un cliente J2EE puede ser un cliente Web o una aplicación cliente.

3.2.3.3.1 Cliente Web

Un cliente Web consta de dos partes: páginas Web dinámicas, que contienen varios tipos de lenguaje de marcas (HTML, XML, etcétera), y que son creadas por los componentes Web, que funcionan en un browser, el cual muestra las páginas recibidas del servidor.

Un cliente Web, a veces, es llamado un cliente delgado. Los clientes delgados no hacen requerimientos a las bases de datos, no ejecutan reglas de negocio complejas, y no se conectan con aplicaciones heredadas. Cuando se utiliza un cliente delgado, las llamadas operaciones duras, son ejecutadas por los beans de empresa, que funcionan en el servidor J2EE, donde ellos pueden aplicar la seguridad, la velocidad, los servicios, y la confiabilidad de las tecnologías del servidor J2EE.

3.2.3.3.2 Aplicaciones Clientes

Las aplicaciones clientes funcionan en la máquina del cliente, y proporcionan una interfaz a los usuarios para que realicen tareas. Esta interfaz, puede ser provista por un lenguaje de marcas, como HTML, XML, u otro. Tienen típicamente una interfaz gráfica de usuario (GUI), creada a partir de Swing o AWT, pero una interfaz de línea de comandos, también es posible.

Las aplicaciones clientes tienen acceso directo a los beans de empresa, que funcionan en la parte Negocio. Sin embargo, si los requisitos de la aplicación la autorizan, una aplicación cliente puede abrir una conexión HTTP, para establecer comunicación con un servlet, que funciona en la parte web.

3.2.3.4 Arquitectura de componentes JavaBeans

Las partes del servidor y del cliente, también pueden incluir componentes, basados en la arquitectura de componentes de JavaBeans, para manejar los flujos de datos entre una aplicación cliente o applet, y componentes que funcionan en el servidor J2EE, o entre componentes del servidor y una base de datos. Es importante mencionar que los componentes de JavaBeans, no son considerados componentes J2EE, por la especificación J2EE.

Los componentes JavaBeans, poseen propiedades, y tienen métodos set y get, para acceder y manejar estas propiedades. Los componentes JavaBeans usados de ésta manera son

típicamente simples en su diseño, y puesta en funcionamiento, pero deben estar conforme a las convenciones de nombramiento y diseño bosquejadas en la arquitectura de componente de JavaBeans.

3.2.3.5 Comunicación con el Servidor J2EE

En la ilustración 6, se muestran los distintos elementos que pueden estructurar la parte del cliente. El cliente se comunica con la parte del negocio, que funciona en el servidor J2EE directamente o, como en el caso de un cliente que funciona en un browser, pasando a través de página JSP o, servlets que funcionan en la parte del Web.

La aplicación J2EE puede utilizar un delgado cliente browser, o una densa aplicación cliente. En la decisión de cual usar, se debe estar consciente de las ventajas y desventajas, entre conservar funcionalidad en el cliente y cerca del usuario (cliente denso), o dejar tanta funcionalidad como sea posible al servidor (cliente fino). Cuanta más funcionalidad se saca al servidor, cuanto más fácil es distribuir, desplegar, y manejar la aplicación; sin embargo, manteniendo más funcionalidad en el servidor, se puede lograr un mejor rendimiento, percibido por el usuario[8].

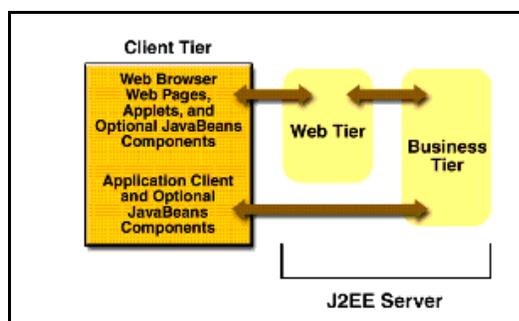


Ilustración 6 Comunicación con el Servidor

3.2.3.6 Componentes Web

Los componentes Web de J2EE, son servlets o, páginas Web creadas usando la tecnología JSP. Los servlets, son clases del lenguaje de programación Java, que dinámicamente procesan peticiones y construyen las respuestas a estas peticiones. Las páginas JSP, son documentos de texto, que se ejecutan como servlets pero permiten una forma más fácil de crear el contenido estático.

Las páginas estáticas HTML y los applet, son enlazados con los componentes Web, durante el ensamble de la aplicación, pero no son considerados componentes Web por la especificación J2EE. Clases utilitarias del Servidor, también se pueden enlazar con los componentes Web y, como las páginas HTML, no se consideran componentes Web.

Como se muestra en la ilustración 7, la parte Web, como la parte cliente, pueden incluir un componente de JavaBeans para manejar las entradas del usuario, y enviar esa entrada a los beans de empresa que funcionan en la parte del negocio, para que la procesen.

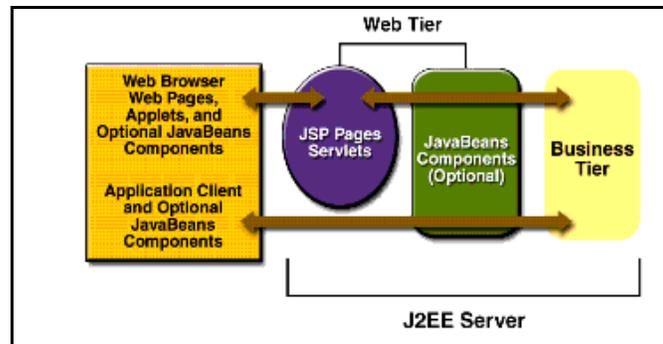


Ilustración 7 Parte Web y Aplicación J2EE

3.2.3.7 Componentes de la parte Negocio

El código del negocio, que es el método que resuelve las necesidades de un área particular de negocio, tales como, actividades bancarias, ventas al por menor, o finanzas, es manejado por los beans de empresa, que funcionan en la parte del negocio. La ilustración 8, muestra cómo un bean de empresa, recibe datos de los programas del cliente, procesa estos (si es necesario), y los envía a la parte del sistema de información de la empresa, para su almacenamiento. Un bean de empresa también recupera los datos almacenados, procesa estos (si es necesario), y los envía de nuevo al programa del cliente.

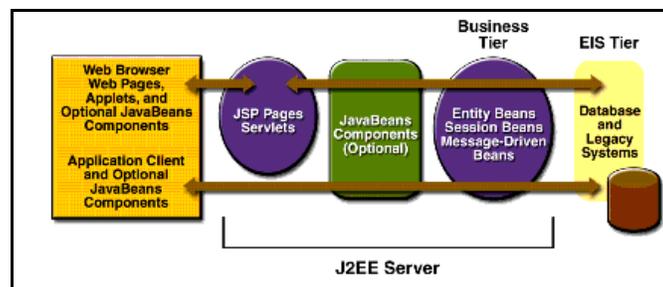


Ilustración 8 Parte Negocio y EIS

Hay tres tipos de beans de empresa: beans de sesión, beans de entidad, y beans manejadores de mensaje. Un bean de sesión, representa una conversación transitoria con un cliente. Cuando el cliente acaba su ejecución, se van el bean de sesión y sus datos. En contraste, un bean de entidad representa los datos persistentes almacenados en la fila de una tabla de una base de datos. Si el cliente termina o si el servidor se cierra, los servicios subyacentes, se aseguran de que los datos del bean de entidad, sean guardados. Un bean manejador de mensajes, combina características de un bean de sesión, y de un oyente de

mensajes, del servicio de mensajes de Java (JMS), permitiendo que un componente de negocio, reciba mensajes asincrónicos de JMS.

3.2.4 Contenedores J2EE

Los Contenedores son la interfaz entre un componente y la funcionalidad de bajo nivel de la específica plataforma que respalda al componente. Antes de crear un servicio Web, beans de empresa, o antes de que pueda ser ejecutado un componente de una aplicación cliente, este debe ser montado en un módulo de J2EE y ser desplegado en su contenedor.

3.2.4.1 Tipos de Contenedores

El proceso del despliegue instala componentes de la aplicación J2EE en los contenedores J2EE ilustrados en la figura 9.

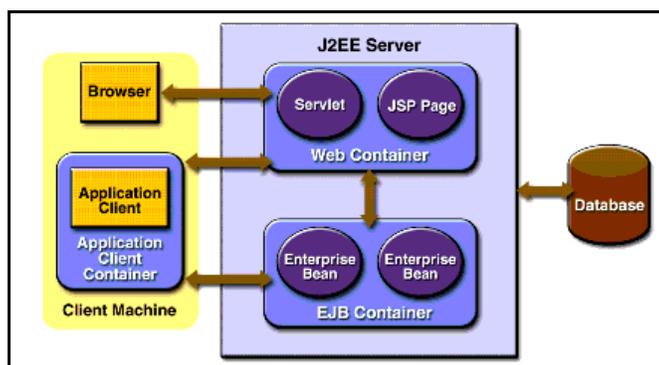


Ilustración 9 Servidor y contenedores de J2EE

3.2.4.1.1 Servidor J2EE

En él se encuentra la porción runtime de un producto J2EE. Un servidor J2EE proporciona los contenedores de los EJB y del Web.

3.2.4.1.2 Contenedor de JavaBeans (EJB) empresariales

Administra la ejecución de los beans de empresa, de las aplicaciones J2EE. Las beans de empresa y su contenedor corren en el servidor J2EE.

3.2.4.1.3 Contenedor Web

Administra la ejecución de los componentes JSP y los servlet de las aplicaciones J2EE. Los componentes Web y su contenedor corren en el servidor J2EE.

3.2.4.1.4 Contenedor de Aplicaciones cliente

Administra la ejecución de los componentes de las aplicaciones cliente. Las aplicaciones cliente y su contenedor corren en el servidor J2EE.

3.2.4.2 Servicios de los Contenedores

Como la arquitectura de J2EE proporciona servicios configurables, a través del manejo de estos por medio de los contenedores, los componentes de una aplicación, dentro de la misma aplicación, pueden funcionar en forma diferente, dependiendo en donde ellos son desplegados. Por ejemplo, un beans de empresa puede tener ajustes de seguridad que le permitan un cierto nivel de acceso a los datos de la base de datos, en un ambiente de producción, y otro nivel de acceso a la base de datos en otro ambiente de producción.

El contenedor también maneja servicios no configurables, tales como ciclos de vida de los beans de empresa, y de los servlet, manejo del conjunto de recursos de la conexión a la base de datos, persistencia de los datos, y acceso a la plataforma API de J2EE. Aunque la persistencia de los datos es un servicio no configurable, la arquitectura de J2EE permite que se sobrescriba el manejador del contenedor de persistencia, incluyendo el código apropiado en la implementación del bean de empresa cuando se desea tener más control que la que provee el manejador del contenedor de persistencia. Por ejemplo, puede ser que se utilice un bean que maneje la persistencia, para implementar un propio buscador de métodos, o crear un cache construido según las especificaciones de la base de datos

3.2.5 J2EE APIs

Las API, o interfaces para la programación de Aplicaciones, son la parte más importante que debe conocer el interesado en desarrollar aplicaciones, ya que en base a estas desarrollará todo su trabajo. A continuación se muestran y analizan las API del estándar J2EE, y por último se mostrarán las ventajas de utilizar estas.

3.2.5.1 Enterprise JavaBeans Technology

Los EJB son bloques funcionales ya creados, los cuales pueden ser usados en forma individual o con otros EJB. Los EJB tienen campos y métodos que implementan distintos módulos Los EJB a menudo son utilizados para interactuar con bases de datos, ya que implementan en forma automática operaciones de acceso rápidas a estas.

3.2.5.2 Java Servlet Technology

Esta API es usada para que el programador defina clases servlet específicas para la Web. Estas clases son usadas para extender la capacidad de los servidores que tienen aplicaciones que está siendo accedidas vía Web.

Las clases servlet pueden responder cualquier tipo de requerimientos, no solo Web, pero los mayores usos de ellos se dan en este campo.

3.2.5.3 Java Server Pages Technology

Esta API permite que se pongan trozos del código de un servlet directamente en un documento de texto. Con esto se crean páginas Web JSP, las cuales son un documento de texto que contiene dos tipos de datos; estáticos (código HTML, WML o XML) y elementos JSP, los cuales determinan cómo la página construirá el contenido dinámico

3.2.5.4 Java Transaction API

Esta API, llamada en forma resumida JTA, provee una interfaz estándar para transacciones demarcadas. J2EE proporciona por defecto un auto commit para manejar transacciones confiables. Un auto commit establece que si una aplicación está viendo datos de una base de datos, éstos serán actualizados en forma automática, si es que en la base de datos, estos son modificados por otra aplicación o por la misma aplicación.

Sin embargo, si la aplicación desarrollada lleva a cabo dos operaciones separadas de acceso a la base de datos, que dependen una de la otra, se deberá utilizar esta API para demarcar toda la transacción, incluyendo ambas operaciones, sus comienzos, vueltas atrás, y commit.

3.2.5.5 JDBC API

Esta API permite invocar comandos SQL a través de métodos de programación de Java. Se puede utilizar esta API en un beans de empresa cuando se sobrescribe el manejador del contenedor de persistencia o se crea un beans de sesión tenga acceso a la base de datos. Con el manejador del contenedor de persistencia por defecto, las operaciones de acceso a la base de datos son manejadas por el contenedor, y el beans de empresa no contendrá código JDBC ni comandos SQL.

También se puede también utilizar ésta API desde un servlet o de una página JSP para tener acceso a la base de datos directamente sin pasar a través del bean de de empresa.

El JDBC API tiene dos partes: una interfaz de nivel de aplicación usada por los componentes de la aplicación para acceder a la base de datos, y un servicio provisto por la interfaz para unir un driver JDBC a la plataforma J2EE.

3.2.5.6 Java Authentication and Authorization Service

Esta API (JAAS), proporciona una forma para que aplicaciones J2EE autentifiquen y autoricen a un usuario específico o a un grupo de usuarios. JAAS es una versión de del lenguaje de programación Java del estándar Pluggable Authentication Module (PAM)

framework, el cual extiende la arquitectura de seguridad de la plataforma Java2, en cuanto a apoyar el manejo de la autorización de los usuarios.

3.2.5.7 Ventajas al utilizar las APIs

Los J2EE APIs, permiten sistemas y aplicaciones integradas, debido a múltiples características, las cuales se resumen en la siguiente lista:

- Modelo unificado de aplicaciones a través de jerarquías con los beans de empresa.
- Mecanismo simplificado de solicitud y respuesta con las páginas JSP y los servlets.
- Modelo confiable de la seguridad con JAAS.
- Fácil conectividad a base de datos con JBC API
- Integración de aplicaciones de empresa con beans de manejo de mensajes y JMS, JTA, y JNDI.

3.2.6 Oracle

Oracle es un sistema de gestión de base de datos objeto-relacional (o ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando:

- Soporte de transacciones
- Estabilidad,
- Escalabilidad
- Soporte multiplataforma.

Su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySql o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

3.2.6.1 Reseña histórica de Oracle

Oracle surge en 1977 bajo el nombre de SDL (Software Development Laboratories), luego en 1979 SDL cambia su nombre por Relational Software, Inc. (RSI). La fundación de Software Development Laboratories (SDL) fue motivada principalmente a partir de un estudio sobre los SGBD (Sistemas Gestores de Base de Datos) de George Koch. Computer World definió este estudio como uno de los más completos jamás escritos sobre bases de datos. Este artículo incluía una comparativa de productos que erigía a Relational Software como el más completo desde el punto de vista técnico. Esto se debía a que usaba la filosofía de las bases de datos relacionales, algo que por aquella época era todavía desconocido.

En la actualidad, Oracle (Nasdaq: ORCL) todavía encabeza la lista. La tecnología Oracle se encuentra prácticamente en todas las industrias alrededor del mundo y en las oficinas de 98 de las 100 empresas Fortune 100. Oracle es la primera compañía de software que desarrolla e implementa software para empresas 100 por ciento activado por Internet a través de toda su línea de productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones. Oracle es el proveedor mundial líder de software para administración de información, y la segunda empresa de software.

3.2.6.2 Características de Oracle

- Portabilidad y escalabilidad. El código de una plataforma es fácilmente transportado a otra, gracias al pre compilador SQL de Oracle. En UNIX, Oracle tiene versiones para más de 30 diferentes tipos de UNIX.
- Permite implantar procesos distribuidos de manera robusta, gracias a su protocolo SQL*Net, disponible para NetBIOS, IPX/SPX, Named Pipes, TCP/IP, entre otros, ocupando estos drivers una cantidad moderada de RAM, lo que permite diseñar soluciones flexibles.
- Oracle es conocido como un gran consumidor de recursos. Cada usuario que hace acceso al servidor requiere por lo menos 250 KB de RAM en el servidor, ya que inicia un proceso completamente nuevo. Si a esto le agregamos la propia carga del sistema operativo y del software de Oracle, obtenemos que, en PCs, un servidor de 16 MB puede soportar, de manera eficiente, solamente de 16 a 20 usuarios sin comenzar a degradar el desempeño como resultado de la poca RAM disponible.
- Oracle ofrece una extensión de SQL llamada PL/SQL (Procedural Lenguaje/SQL); este lenguaje fue desarrollado como respuesta al lenguaje Transact-SQL de Sybase, incluyendo, al igual que aquel, comandos para implantar estructuras de programación tales como ciclos y decisiones, así como para soportar arreglos, variables, funciones matemáticas y manejo de caracteres para crear aplicaciones más complejas.

3.2.6.3 Ventajas

- Oracle es el DBMS que cuenta con el segundo número más grande de productos front-end de terceros (después de SQL Server de Microsoft y Sybase).
- El ser considerado como líder del mercado de DBMSs es debido, principalmente, a su escalabilidad, ya que esto permite tomarlo con manejador de bases de datos para toda una organización, sin importar la variedad de hardware que tenga. Esto, además, reduce la necesidad y costo, de capacitar a los programadores, desarrolladores, y personal de soporte en diferentes DBMSs. Adicionalmente, los usuarios de Oracle pueden encontrar en el mercado de trabajo un gran número de personal ya capacitado.
- Los manuales de Oracle tienen una calidad muy buena; además de cubrir información abundante, son claros y concisos.

3.2.7 HTTPS

El protocolo seguro de transferencia de hipertexto es una combinación del protocolo http y del protocolo de redes seguras. Ambos trabajan en la capa más alta de TCP/IP, la capa de

aplicación, el protocolo de seguridad opera en una subcapa más baja, encriptando el mensaje http antes de transmitirlo, también se encarga de desencriptar cuando el mensaje llega.

El sistema HTTPS utiliza un cifrado basado en las Secure Socket Layers (SSL) para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo http.

Las conexiones que usan HTTPS se utilizan comúnmente en el manejo de transacciones bancarias, pagos web y sistemas que manejan información importante.

3.2.7.1 Funcionamiento de HTTPS

Para crear un servidor web que acepte conexiones HTTPS, el administrador del sitio debe crear un certificado de clave pública para el servidor. Este certificado puede ser creado para servidores basados en Unix, con herramientas como OpenSSL o SuSE. Este certificado debe estar firmado por una autoridad certificadora. Esta autoridad certifica que el certificado es de quien dice que es.

El sistema puede ser usado para autenticar clientes y limitar el acceso al servidor a usuarios autorizados. Para esto, el administrador del sitio crea un certificado por cada usuario, un certificado que es cargado en su navegador. Normalmente, este contiene el nombre y el correo electrónico del usuario autorizado y es automáticamente verificado por el servidor en cada re-conexión del usuario.

3.2.7.2 Desventajas

El nivel de protección depende de la correcta implementación por el navegador, del software del servidor y de los algoritmos actuales de criptografía aceptados.

3.2.8 XML

XML es un metalenguaje muy sencillo en su diseño que permite estructurar y realizar el intercambio de datos entre distintas aplicaciones. Se convirtió en un estándar, ya que es extensible y puede ser utilizado por cualquier aplicación independientemente de la plataforma.

XML es una versión reducida y adaptada de SGML cuya función es la de definir la estructura de lenguajes específicos, es decir XML no es un lenguaje como tal, XML ayuda a definir lenguajes para distintas aplicaciones, ejemplos de estos son XHTML y SVG. XML es muy sencillo y simple lo que le da fortaleza es la forma en que se complementa con otras tecnologías lo que la hace más grande y con más posibilidades.

3.2.8.1 Estructura de un documento XML

Un documento XML está formado por el prólogo y por el cuerpo del documento.

3.2.8.1.1 Cabecera

Los documentos XML pueden empezar con unas líneas que describen la versión XML, el tipo de documento y otras cosas. En general una cabecera está compuesta por: Una declaración XML, una declaración de tipo de documento, uno o más comentarios e instrucciones de procesamiento.

3.2.8.1.2 Cuerpo

El cuerpo no es opcional en un documento XML, el cuerpo debe contener un y solo un elemento raíz:

- Elementos: Los elementos XML pueden tener contenido (más elementos, caracteres o ambos), o bien ser elementos vacíos.
- Atributos: Los elementos pueden tener atributos, que son una manera de incorporar características o propiedades a los elementos de un documento. Deben ir entre comillas.
- Entidades predefinidas: Entidades para representar caracteres especiales para que, de esta forma, no sean interpretados como marcado en el procesador XML.
- Secciones CDATA: Es una construcción en XML para especificar datos utilizando cualquier carácter sin que se interprete como marcado XML. Solo se utiliza en los atributos.
- Comentarios: Comentarios a modo informativo para el programador que han de ser ignorados por el procesador.

3.2.8.1.3 Ventajas

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles.
- Es gratuito: al elegir XML como base para algún proyecto se tiene disposición una gran y creciente comunidad de herramientas e ingenieros experimentados en la tecnología.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones.

3.2.8.1.4 Desventajas

- Al ser un lenguaje de etiquetado XML y de texto, los documentos XML son mucho mayores en tamaño que un archivo binario común, aunque esto no represente un problema para los grandes niveles de almacenamiento que se manejan hoy en día por

los computadores y servidores, si es un inconveniente para los dispositivos móviles y la transmisión de los mismos por redes celulares.

3.2.8.1.5 Ejemplo de XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<libro>
  <título></título>
  <capitulo>
    <título></título>
    <seccion>
      <título></título>
    </seccion>
  </capitulo>
</libro>
```

Este XML describe un objeto “Libro”, que contiene un título y un capítulo, el capítulo a su vez está compuesto por otro título y una sección, y en esta última también existe otro título. Para archivos pequeños como este es fácil para una persona comprender el contenido del XML, y también es fácil ver como se podrían añadir más atributos a este libro, como la cantidad de páginas por capítulo y el nombre del autor en el título.

3.2.9 Subversion-SVN

Es un sistema de control de versiones de código abierto, maneja archivos y directorios a través del tiempo, usa un árbol de archivos en un repositorio central. El repositorio es como un servidor de archivos normal, excepto porque almacena todos los cambios hechos a sus archivos y directorios. Esto permite recuperar versiones antiguas de los datos, o examinar el historial de cambios de los mismos.

SVN puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos equipos.

3.2.9.1 Versionado de directorios

Subversion implementa un sistema de archivos versionado “virtual” que sigue los cambios sobre árboles de directorios completos a través del tiempo. Ambos, archivos y directorios, se encuentran bajo el control de versiones.

3.2.9.2 Verdadero historial de versiones

En subversión se puede crear, eliminar, copiar y renombrar archivos y directorios, y cada archivo nuevo añadido comienza con un historial nuevo, limpio y propio.

3.2.9.3 Envíos atómicos

Un conjunto cualquiera de modificaciones o bien entra por completo al repositorio, o bien no lo hace en absoluto. Esto permite a los desarrolladores construir y enviar los cambios como fragmentos lógicos e impide que ocurran problemas cuando sólo una parte de los cambios enviados lo hace con éxito.

3.2.9.4 Versionado de metadatos

Cada archivo y directorio tiene un conjunto de propiedades (claves y sus valores) asociado a él. Se puede crear y almacenar cualquier par arbitrario de clave/valor que se desee. Las propiedades son versionadas a través del tiempo, al igual que el contenido de los archivos.

3.2.9.5 Manipulación consistente de datos

Subversion expresa las diferencias del archivo usando un algoritmo de diferenciación binario, que funciona idénticamente con ficheros de texto (legibles para humanos) y ficheros binarios (ilegibles para humanos). Ambos tipos de ficheros son almacenados igualmente comprimidos en el repositorio, y las diferencias son transmitidas en ambas direcciones a través de la red.

3.2.9.6 Herramientas

Existen varias herramientas para usar Subversion, tanto programas individuales como interfaces que lo integran en IDEs.

- TortoiseSVN. Provee integración con el explorador de Windows. Esta fue la herramienta de Subversion escogida para el proyecto de telemedina, ya que su intuitivo y fácil uso fue un factor decidor al momento de escoger este tipo de herramientas.
- Subclipse. "Plugin" que integra Subversion al entorno Eclipse.
- Subversive. "Plugin" alternativo para Eclipse.
- Cervisia, RapidSVN son programas para interacción para linux
- ViewVC. Interfaz web.
- Para Mac, pueden emplearse SvnX, RapidSVN y Zigversion
- KDESvn. Provee integración con el escritorio KDE, muy parecido en apariencia/funcionamiento/características a TortoiseSVN
- Easyeclipse, es un paquete basado en eclipse es una plataforma de desarrollo, con algunos plugins de código abierto.
- sventon Interfaz web
- Versions Interfaz de escritorio para Mac OS X

3.2.10 Google code

Google Code es un sitio de Google para desarrolladores interesado en desarrollo open source. El sitio contiene código open source y una lista de servicios que soportan APIs públicos.

Es el espacio donde Google comparte con todos los usuarios parte del código de programación que se utiliza dentro de la compañía. Este código se ofrece con licencia libre, para que cualquier desarrollador pueda utilizarlo en sus proyectos, o incluso modificarlo.

No se trata del código que clasifica las páginas web del potente buscador, sino de parte de lo utilizado para crear las numerosas herramientas que ofrece Google [11].

El uso de Google Codes en el proyecto de Telemedicina, se debe a que el repositorio central se ubicó en este sitio. La combinación entre Google Codes y la herramienta de Subversion Tortoise SVN, posibilitó crear las distintas versiones de las aplicaciones y manejar la escalabilidad de estas.

3.3 Metodologías de desarrollo

Escoger la metodología de desarrollo fue uno de los temas que más ocupó tiempo de investigación en la etapa preliminar del trabajo, puesto que de ella dependía la forma de trabajo durante toda la labor de ingeniería.

Se quería usar una metodología ágil por que se esperaba reducir los tiempos para lograr los objetivos propuestos ya que la labor de investigación de dominio podría retrasar el proyecto, se evaluaron metodologías ágiles como XP (eXtremme Programming), AUP (Agile Unified Process), SCRUM y FDD (Feature Driven Development).

Los modelos escogidos para el desarrollo del proyecto de telemedicina fueron el Modelo Evolutivo y el Modelo en Cascada:

3.3.1 Modelo Evolutivo

Para la primera etapa del proyecto de Telemedicina, se seleccionó y luego se implementó la metodología de desarrollo del Modelo Evolutivo, ya que en este modelo la especificación y el desarrollo están entrelazados.

El objetivo de utilizar este modelo, fue crear un desarrollo exploratorio, que permitiera trabajar y evolucionar hacia un sistema final. Para poder implementar este modelo, fue requisito tener bien claro los requerimientos del sistema.

Las partes difíciles de especificar por adelantado (la interfaz de usuario, por ejemplo) se desarrollaron utilizando un enfoque evolutivo

El Modelo Evolutivo permitió generar prototipos desechables, ya que en la primera etapa del proyecto, no se tenía gran claridad acerca de los requerimientos del sistema, sobre todo en lo que concierne a la parte médica. La primera etapa del proyecto, comenzó con requerimientos pobremente conocidos.

3.3.1.1 Construcción de Prototipos

La construcción de Prototipos mediante el Modelo Evolutivo, se desarrolló con una implementación inicial, que fue expuesta a los comentarios del profesor guía y redefiniéndola a través de las diferentes versiones del proyecto de Telemedicina.

Las actividades de especificación, desarrollo y validación se llevaron a cabo concurrentemente, y tuvieron una realimentación rápida a lo largo del proceso, para lo anterior, se manejó principalmente, planillas en excel con los requerimientos propuestos por el alumno y las pertinentes observaciones planteadas como respuesta por parte del profesor.

El flujo anteriormente descrito, llevo a un primer prototipo de sistema que se desarrolló rápidamente, a partir de las especificaciones abstractas, y se refino en las posteriores versiones del proyecto, con la ayuda del profesor.

Un enfoque evolutivo para el desarrollo de software es muy efectivo, ya que cumple con las necesidades inmediatas de los clientes, además, las especificaciones se pueden desarrollar de forma creciente.

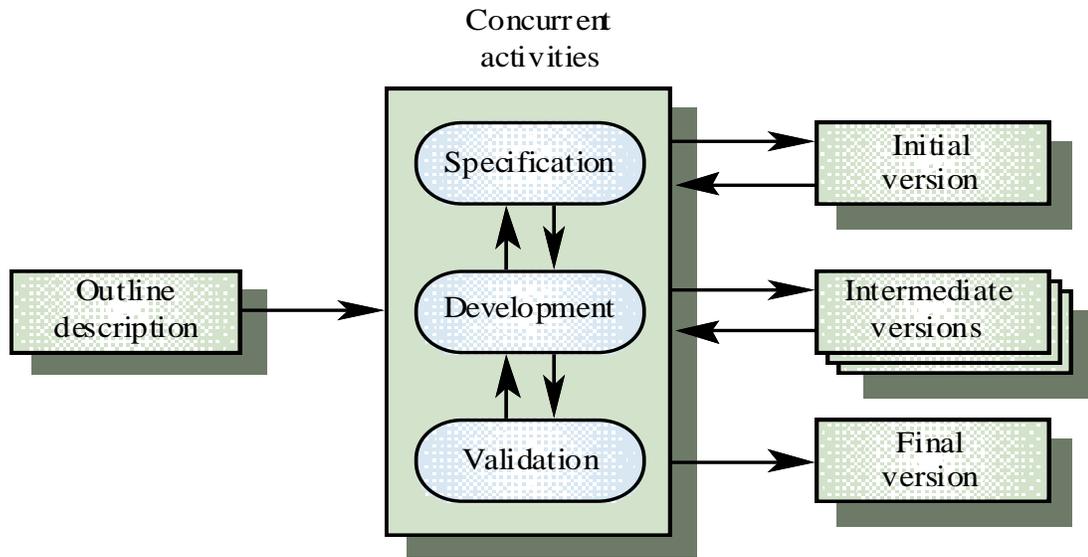


Ilustración 10 Modelo Evolutivo.

3.5.1.1 Problemas del Modelo Evolutivo

Desde una perspectiva de ingeniería y administración, el desarrollo evolutivo supone los siguientes problemas:

1. El proceso no es visible

Si los sistemas se desarrollan rápidamente, es muy costoso producir documentos que reflejan cada versión del sistema.

2. Frecuentemente los sistemas tienen una estructura deficiente

Los cambios continuos tienden a corromper la estructura del software.

3.3.2 Ventajas del Modelo Evolutivo

- Permite ir comprendiendo de mejor maneja los requerimientos a medida que se avanza en el proyecto
- El cliente puede visualizar, en etapas tempranas, las características principales que tendrá el software.

3.3.3 Modelo en Cascada

En el Modelo en cascada, denominado también como ciclo de vida del software, las fases de especificación y desarrollo son separadas y distintas [5].

El modelo se denomina “cascada” debido a la cascada de una fase a otra.

Se utilizó el Modelo en cascada, una vez que se tuvo los requerimientos claros y bien definidos, gracias al Feedback que se estableció con el profesor guía. En dicha etapa, existió una probabilidad muy baja de cambio de requerimientos en el tiempo, lo que propicio el uso de este modelo en el proyecto de telemedicina.

Gracias a la implementación de este modelo, la documentación generada, correspondió a cada fase del proyecto que fue terminada.

3.3.3.1 Etapas del Modelo en Cascada

Las etapas del Modelo en Cascada se definen como sigue:

1. Análisis y definición de requerimientos:

Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios, en detalle, sirviendo como una especificación del sistema

2. Diseño de sistemas y de software:

El diseño de sistemas separa los requerimientos de hardware o de software y establece una arquitectura completa del sistema

3. Implementación y prueba de unidades:

Se obtiene un conjunto o unidades de programas
Cada una de las unidades debe cumplir su especificación

4. Integración y prueba del sistema:

Las unidades individuales se integran y prueban como un sistema completo, que debe cumplir los requerimientos del software

5. Operación y mantenimiento:

El sistema se instala y pone en uso; se corrigen los errores no descubiertos en las etapas anteriores, se mejora la implementación de las unidades, los servicios del sistema, si se establecen nuevos requerimientos

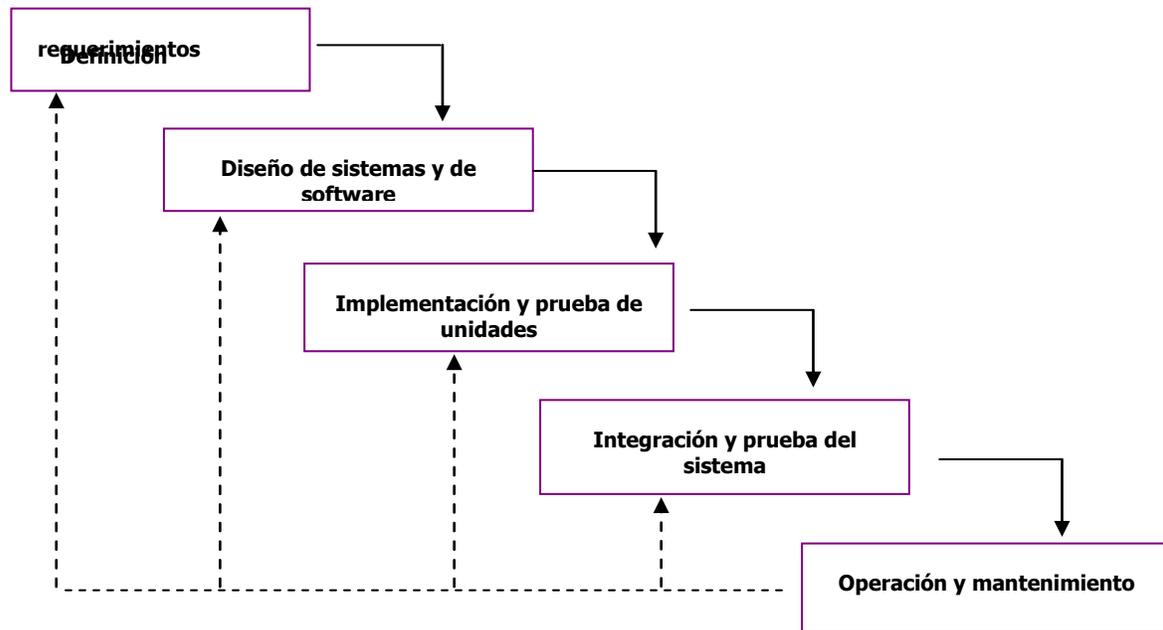


Ilustración 11 Modelo en Cascada

3.5.1.2 Problemas del Modelo en Cascada

1. Cada fase debe ser terminada antes de seguir con la siguiente

- Las etapas interaccionan e intercambian informaciones
- El proceso no es un modelo lineal simple, sino que implica una serie de iteraciones de las actividades de desarrollo

2. Las iteraciones son costosas

- Implican rehacer el trabajo y volver a aprobar documentación.
- Luego de algunas iteraciones, se tiende a romper con la regla de que cada fase debe ser terminada para seguir con la siguiente.

3. Puede aparecer la necesidad de repetir

Puede aparecer la necesidad de repetir algunas (o todas las) etapas previas del proceso, debido a los errores y omisiones que se descubren, o a la necesidad de nuevas funcionalidades

3.3.4 Metodología para BD

El diseño de Base de datos es el proceso de producir un modelo de datos detallado para una base de datos. Este modelo lógico de base de datos contiene todas las opciones de diseño físicas y lógicas, y los parámetros físicos requeridos para generar un diseño en un lenguaje de

definición de datos, el cual puede luego ser usado para crear una base de datos. Un modelo de datos con atributos completos contiene atributos detallados para cada entidad.

3.3.5 Análisis

Propósito: Analizar documentos y tareas; determinar los requerimientos del sistema.

Entrada: Descripción de documentos y tareas, Escenarios, Estadísticas de uso, Planes para el futuro del sistema, Leyes relevantes, restricciones y políticas.

Salida: Diagrama de flujo de información, moldeamiento de documentos de entrada y salida, documentos internos de entrada y salida, tareas y límites del sistema

Técnicas:

- Entrevistas con las personas involucradas
- Análisis de los documentos, escenarios y tareas
- Revisión de planes a corto y largo plazo, manuales, archivos y formularios
- Abstracción
- Trabajo de afuera hacia adentro

Herramientas: Diagramas de flujo de información

3.3.6 Especificación

Propósito: Crear una especificación detallada de los documentos internos y tareas del diagrama de flujo de información.

Entrada: Diagrama de flujo de información, estadísticas de uso, y otra información adquirida durante el análisis.

Salida: Diagrama ER, Representación de los datos (diccionario), restricciones, Descomposición de tareas, Formularios y Estadísticas

Técnicas:

- Modelamiento de datos
- Descomposición de tareas de arriba hacia abajo hasta que la especificación es lo suficientemente detallada para permitirle a un programador implementarla
- Descomposición de tareas que puede resultar en unas tareas reemplazando las originales o en subtareas controladas por la tarea original

Herramientas: Modelo ER, Formularios de Tareas

3.5.1.3 Diseño

Propósito:

- Crear diseños detallados del esquema normalizado de la base relacional
- Crear diseños detallados de las tareas usando código abstracto con SQL embebido
- Identificar la necesidad de Vistas

Entrada: Formularios de Tareas, Diagramas ER, Diseño de Entidades

Salida: Esquema relación con llaves primarias y foráneas, definición de restricciones en SQL, código abstracto con SQL, definición de vista

Técnicas: Normalización de base de datos, Código abstracto.

Herramientas:

- Mapeo: Modelo ER, Modelo Relacional
- Desencadenadores Gráficos
- Código Abstracto
- SQL
- Vistas

3.3.7 Implementación

Propósito:

- Crear un esquema conceptual
- Crear un esquema interno
- Implementar el código abstracto

Entrada: Esquema relacional con llaves primarias y foráneas, Representación de datos, restricciones en SQL, código abstracto con SQL, Descomposición de Tareas, Definición de vistas

Salida: Esquema conceptual, esquema interno, código "real" con SQL embebido

Herramientas:

- SQL
- Lenguaje de programación
- Sistema DBMS relacional
- Precompilador
- Compilador de lenguaje de programación

4 Capítulo Cuarto: Desarrollo del proyecto

4.1 Alcances

A partir de un levantamiento de requerimientos se generó: un modelo general de prototipo de sistema de información, un modelo de datos para la administración y persistencia de la información, y una interfaz entendible para los usuarios finales. El desarrollo de este proyecto pretendió brindar un apoyo al servicio médico en lo concerniente a la atención domiciliaria y la asistencia médica remota, sirviéndoles de apoyo y soporte para la consulta y modificación de la información de un paciente, sus datos básicos, su historia, sus antecedentes; además de servirles de plataforma para remitir casos que están fuera de su alcance a médicos especialistas, ordenar formulas médicas, dar incapacidades, y solicitar exámenes paraclínicos.

El personal médico podrá realizar consultas de casos complicados con otros colegas. También podrán realizar análisis médicos y dar sus comentarios a los casos publicados por otros médicos.

Además, se podrá obtener información de un paciente en tiempo real al momento de atender una emergencia, y hacer un diagnóstico previo del estado en el que se encuentra el paciente mientras es trasladado, de forma que al arribar al centro de salud, el personal que atendiera la emergencia, se encuentre preparado adecuadamente para recibirlo.

El proyecto de telemedicina, pretende servir como modelo de referencia para la creación de nuevos desarrollos orientados al área de la medicina remota, brindando a la comunidad una base documental acerca del manejo de la información médica para la construcción de proyectos de software. Esta base documental tendrá información acerca de sistemas médicos, manejo de información del paciente (básica, historia, antecedentes), estandarización internacional para la intercomunicación de sistemas médicos, entre otros.

El proyecto consistió en la construcción de un prototipo de sistema de información médica que abarco el manejo de:

- Interconsulta entre médicos, permitiendo que se puedan compartir casos para recibir apoyo de una comunidad completa de médicos.
- Acceso a historias clínicas, antecedentes, y procedimientos médicos para escritura y/o lectura.
- Manejo de visitas médicas, horarios y disponibilidad de los médicos además de permitirles manejar sus agendas creando únicamente las horas donde estos podrán atender y el lugar donde se efectuara si es una institución o la dirección donde hará la visita.
- Solicitud de servicios tales como exámenes médicos, remisión a especialistas, petición de medicamentos disponibles y el registro de dicha información en la historia clínica del paciente.
- Atención domiciliaria médico general dándole soporte a la forma tradicional de llevar historias clínicas, respetando la información básica que debe contener toda historia clínica.

El desarrollo web se implementó mediante JEE que permitió una integración fácil a base de datos y la creación de formularios web dinámicos mediante Java Script. Todo el prototipo a nivel de servidor ira montado sobre la base de datos ORACLE. A nivel de programación móvil se utilizó Android versión 2.3 que es la herramienta más popular y con mayor documentación en el mercado para el área de dispositivos móviles

También se utilizó conexión mediante HTTP, HTTPS, y SOAP para el estándar de mensajes.

4.2 Limitaciones

Para el desarrollo de este proyecto, se encontraron diversas dificultades, tanto tecnológicas como académicas; entre las cuales podemos identificar la falta de información en el área de medicina sobre todo con los conceptos propios de esa área y la falta de recursos físicos:

- El proyecto se desarrolló bajo herramientas de libre distribución, y versiones académicas que facilitaron el desarrollo del sistema.
- El proyecto no dispuso de un cliente fijo y estable encargado de revisar y retroalimentar cada una de las iteraciones hechas durante las fase de desarrollo.
- No se prevén cambios en el diseño y la codificación de cada una de las iteraciones, puesto que no contamos con el cliente encargado de retroalimentar las mismas.
- El prototipo se desarrollara para un correcto funcionamiento en los dispositivos móviles con Android versión 2.3 y superior.
- El prototipo solo contempla el formulario para el llenado de la historia clínica en consulta general.
- El prototipo en sus alcances solo es capaz de generar la información necesaria para asignar un servicio a un determinado paciente, el seguimiento y el control de cada uno de los diferentes tipos de servicios no se implementara en el prototipo.

4.3 Objetivos del Sistema

Tabla 2 Objetivo del sistema 1

Obj-1	Permitir el acceso a la información del paciente en tiempo real.
Descripción	La información del paciente debe estar disponible todo el tiempo para ser consultada y cualquier modificación que se haga debe ser reflejada en el sistema de información en el menor tiempo posible.
Importancia	Vital

Tabla 3 Objetivo del sistema 2

Obj- 2	Facilitar el registro de la información médica.
Descripción	El sistema de información servirá de herramienta para llevar un registro de la información más accesible y más organizada.

Importancia	Vital
-------------	-------

Tabla 4 Objetivo del sistema 3

Obj-3	Volver más ágiles los procesos médicos.
Descripción	El sistema reducirá los tiempos de búsqueda, de consulta y de actualización de la información médica.
Importancia	Vital

Tabla 5 Objetivo del sistema 4

Obj-4	Facilitar el acceso a la información medica
Descripción	La información del sistema estará disponible en un servidor dedicado con acceso seguro a internet.
Importancia	Importante.

Tabla 6 Objetivo del sistema 4

Obj-5	Garantizar la confiabilidad de la información.
Descripción	Se usaran estándares para tratamiento de la información médica, y protocolos seguros de conectividad y accesos seguro a la información, para garantizar la integridad de la información en el sistema.
Importancia	Vital.

Tabla 7 Objetivo del sistema 6

Obj-6	Permitir el acceso a la información del paciente desde cualquier lugar.
Descripción	La información del paciente debe estar disponible desde cualquier lugar donde haya cobertura de red celular, para ser consultada y actualizada.
Importancia	Importante.

Tabla 8 Objetivo del sistema 7

Obj-7	Mejorar la atención al paciente
Descripción	El sistema permitirá brindar al paciente información de seguimiento sobre todos sus procesos médicos.
Importancia	Importante.

4.4 Reglas de negocio

Tabla 9 reglas de negocio

Identificador	Descripción
RN.1	Únicamente el administrador general podrá crear usuario del tipo administrativo
RN.2	Todos los usuarios del sistema deberán acceder al mismo a través de su número de identificación y su contraseña
RN.3	Únicamente el administrador estará en la capacidad de afiliar nuevo personal médico
RN.4	Solamente el administrador puede estar en capacidad de registrar y modificar pacientes nuevos
RN.5	Solamente el administrador podrá registrar nuevos empleados administrativos (operador de citas)
RN.6	Solamente el administrador podrá modificar a todos los usuarios
RN.7	Solamente el medico visitante podrá agregar nuevas imágenes al historial de visitas
RN.8	Los médicos pueden atender en la consulta o visitar a casa
RN.9	Únicamente los médicos podrán asignar servicios a sus pacientes
RN.10	Los servicios que se le prestan a un paciente son del tipo: Receta medica Examen Licencia medica Modificación de historial medico
RN.11	El paciente puede revisar sus servicios
RN.12	El administrador de citas es el único que puede asignar o modificar las citas medicas
RN.13	La historia clínica podrá ser accedida en su totalidad únicamente un médico que este atendiendo a ese paciente
RN.14	Un usuario no puede ser eliminado, solamente actualizado
RN.15	Una Historia clínica no puede ser modificada, solo actualizada

4.5 Requerimientos de información

Tabla 10 Información sobre el administrador general

IRQ-01	Información sobre el administrador general
Descripción	El sistema debe guardar la información necesaria con respecto a el administrador general del sistema
Datos específicos	Cédula Nombre Apellido Contraseña Teléfonos Dirección

	Entidad para la cual trabaja Rol desempeñado
--	---

Tabla 11 Información sobre el personal médico

IRQ-04	Información sobre el personal médico
Descripción	El sistema debe guardar la información necesaria con respecto a el personal médico de una entidad prestadora
Datos específicos	Cédula nombre apellido contraseña teléfonos Dirección entidad a la cual representa tipo de sangre especialidades rol desempeñado hora de inicio hora de fin

Tabla 12 Información sobre pacientes

IRQ-05	Información sobre pacientes
Descripción	El sistema debe guardar la información necesaria con respecto a los pacientes del sistema
Datos específicos	Cédula nombre apellido contraseña teléfonos Dirección tipo de sangre rol desempeñado historia clínica fecha de nacimiento tipo de identificación sexo profesión ciudad estado (activo, bloqueado, etc.) entidad administradora a la cual está afiliado alarmas

Tabla 13 Información sobre los empleados administrativos

IRQ-07	Información sobre los empleados administrativos
Descripción	El sistema debe guardar la información necesaria con respecto a empleados administrativos (administrador de servicios, administrador de citas)
Datos específicos	<p>Cédula nombre apellido contraseña teléfonos Dirección entidad a la cual representa rol desempeñado</p>

Tabla 14 Información sobre las peticiones de exámenes

IRQ-09	Información sobre las peticiones de exámenes
Descripción	El sistema debe guardar la información necesaria con respecto a las peticiones de exámenes que pueden realizar los médicos a sus pacientes
Datos específicos	<ul style="list-style-type: none"> • fecha de solicitud • tipo de examen • descripción • resultados • médico que lo creo • paciente a la cual va dirigida • comentario

Tabla 15 Información sobre las recetas médicas asignadas a los pacientes

IRQ-11	Información sobre las formulas medicas asignadas a los pacientes
Descripción	El sistema debe guardar la información necesaria con respecto a las formulaciones que recibe un paciente por parte de su médico
Datos específicos	<p>fecha de solicitud medicamentos (cantidad y dosis) médico que lo creo paciente al cual va dirigido comentario</p>

Tabla 16 Información sobre las citas médicas

IRQ-15	Información sobre las citas médicas
Descripción	El sistema debe guardar la información necesaria con a las citas médicas asignadas a los pacientes.
Datos específicos	Fecha dirección consultorio médico paciente

Tabla 17 Información sobre los antecedentes del paciente

IRQ-18	Información sobre los antecedentes del paciente
Descripción	El sistema debe guardar la información necesaria con respecto a los antecedentes de los pacientes
Datos específicos	tipo descripción fecha de registro

4.6 Restricciones de información.

Tabla 18 Unicidad de usuarios

CRQ-1	Unicidad de usuarios
Descripción	Los números identificación de usuarios (cédula) deberán ser únicos para cada usuario en conjunto con el rol que desempeñan y la entidad para la cual trabaja, es decir, no puede haber dos usuarios distintos con el mismo número y el mismo rol.

Tabla 19 Unicidad de las citas

Tabla 19 CRQ-2	Unicidad de las citas
Descripción	No pueden existir varias citas asignadas al mismo paciente en una misma fecha

4.7 Actores del sistema

Tabla 20 Administrador General

ACT-1	Administrador General
Descripción	El administrador general simula al Estado, es el encargado de administrar el sistema.

Tabla 21 Administrador de citas

ACT-2	Administrador de citas
Descripción	Las entidades administradoras de salud, organizan las citas de sus afiliados con el personal médico de una de las entidades prestadoras con quienes trabajan. Se encarga de gestionar todo el proceso de administración de citas médicas.

Tabla 22 Personal Médico

ACT-3	Personal Médico
Descripción	El personal médico hace referencia a cualquier empleado en el área de la salud de la entidad prestadora. Médicos, enfermeras, paramédicos o especialistas. Su función es la de atender las necesidades de sus pacientes mediante el registro y seguimiento de eventos en las historias clínicas de estos.

Tabla 23 Médico General

ACT-4	Médico General
Descripción	Se encarga de evaluar, revisar y atender a los pacientes dentro del sistema de salud, en un contexto de una consulta general. Es el único actor con acceso total a la historia clínica de un paciente, además de poder anexar información en ella.

Tabla 24 Paciente

ACT-5	Paciente
Descripción	El paciente es cualquier persona natural afiliada al sistema de salud, está en capacidad de ser atendido por un personal médico en caso de recurrir a algún servicio de salud. Puede revisar sus citas pendientes, y también puede recibir notificaciones de eventos relacionados con su salud.

4.8 Requerimientos funcionales

Tabla 25 Consultar agenda médico desde el móvil

FRQ-01	Consultar agenda médico desde el móvil
Descripción	El médico podrá consultar su agenda de citas desde el dispositivo móvil.
importancia	alta

Tabla 26 Sincronizar Agenda con el servidor

FRQ- 02	Sincronizar Agenda con el servidor
Descripción	La agenda del médico en el móvil, debe estar actualizada en todo momento, se debe realizar una sincronización de la aplicación móvil con el servidor que es donde se encuentra la información actualizada de la agenda médica en particular.
importancia	alta

Tabla 27 Ver detalles de cita en la agenda Médica móvil

FRQ- 03	Ver detalles de cita en la agenda Médica móvil
Descripción	Se pueden ver los detalles de una cita: nombre del paciente, número de identificación, dirección ó consultorio, fecha y hora de la cita, numero de contacto del paciente. Desde la aplicación móvil
importancia	alta

Tabla 28 Ver detalles de cita en la agenda Médica Web

FRQ- 04	Ver detalles de cita en la agenda Médica Web
Descripción	Se pueden ver los detalles de una cita: nombre del paciente, número de identificación, dirección ó consultorio, fecha y hora de la cita, numero de contacto del paciente. Desde un navegador web, en el portal.
importancia	alta

Tabla 29 Consultar agenda desde Web

FRQ- 05	Consultar agenda desde Web
Descripción	El médico podrá consultar su agenda de citas desde el portal web.
importancia	alta

Tabla 30 Realizar consulta Médica General desde el móvil

FRQ- 07	Realizar consulta Médica General desde el móvil
Descripción	El médico podrá realizar consultas médicas generales. La aplicación mostrara al médico una interfaz donde podrá añadir los detalles de la consulta.
importancia	alta

Tabla 31 Realizar petición de exámenes paraclínicos desde el móvil

FRQ- 08	Realizar petición de exámenes paraclínicos desde el móvil.
Descripción	El médico podrá solicitar paraclínicos para un paciente desde el móvil, esta solicitud será guardada en el servidor y dará un número de orden para que el paciente pueda reclamar dicho servicio.
importancia	alta

Tabla 32 Realizar consulta de Historia clínica de un paciente desde el móvil

FRQ- 09	Realizar consulta de Historia clínica de un paciente desde el móvil.
Descripción	El personal médico puede consultar la historia clínica de un paciente desde la aplicación móvil, ingresando el documento del paciente y solo si el personal médico está autorizado para atender a ese paciente.
importancia	alta

Tabla 33 Autenticar usuarios

FRQ- 12	Autenticar usuarios
Descripción	La aplicación móvil y el portal médico solicitaran y validaran un nombre de usuario y contraseña para acceder al sistema.

importancia	alta
-------------	------

Tabla 34 Registrar personal médico en el sistema

FRQ- 14	Registrar personal médico en el sistema.
Descripción	Únicamente los funcionarios administrativos pueden registrar nuevos usuarios de personal médico en el sistema, se deben describir los datos referentes a: nombres completos, edad, cargo, número de identificación.
importancia	alta

Tabla 35 Cambiar contraseña

FRQ- 15	Cambiar contraseña
Descripción	Un usuario puede cambiar su contraseña desde el móvil y desde el portal web, la nueva contraseña no puede ser igual al número de identificación.
importancia	alta

Tabla 36 Anexar multimedia

FRQ- 16	Anexar multimedia
Descripción	El personal médico puede agregar contenido multimedia a los eventos registrados sobre un paciente y a las historias clínicas en caso de una consulta médica general. La multimedia que se puede anexar son fotos, y audio.
importancia	baja

Tabla 37 Administrar citas médicas

FRQ- 19	Administrar citas médicas.
Descripción	El administrador de citas, podrá asignar, reasignar y/o eliminar citas, verificando la disponibilidad del médico y del paciente.
importancia	alta

Tabla 38 Generar formula médica desde el móvil

FRQ- 24	Generar formula médica desde el móvil.
Descripción	El médico podrá recetar medicina y enviar una orden de medicamentos desde la aplicación móvil, esta solicitud será guardada en el servidor y dará un número de orden para que el paciente pueda reclamar la medicación correspondiente.
importancia	alta

Tabla 39 Generar orden de remisión a especialista desde el móvil

FRQ- 25	Generar orden de remisión a especialista desde el móvil.
Descripción	El médico podrá remitir al paciente a un especialista para hacer una inspección más detallada, esta remisión será guardada en el servidor y dará un número de orden para que el paciente pueda solicitar la cita con el especialista.
importancia	alta

Tabla 40 Registrar Evento médico desde el móvil

FRQ- 28	Registrar Evento médico desde el móvil
Descripción	El personal médico autorizado podrá registrar eventos médicos en las historias clínicas de los pacientes, estos eventos están compuestos por: descripción, fecha, lugar, procedimiento. Este registro debe quedar listado en el servidor.
importancia	alta

Tabla 41 Generar Log de cada suceso en el servidor

FRQ- 29	Generar Log de cada suceso en el servidor
Descripción	En el servidor se debe registrar en un log, cada evento generado por cualquier usuario, y por las comunicaciones realizadas por el servidor. Este log será accesible para el administrador general.
importancia	alta

Tabla 42 Actualizar Historia clínica del paciente desde el móvil

FRQ- 30	Actualizar Historia clínica del paciente desde el móvil
Descripción	La aplicación móvil, actualizara la información concerniente a la historia clínica del paciente en el servidor cada vez que el personal médico realice un evento o una consulta médica.
importancia	alta

4.9 Requisitos no funcionales

Tabla 43 Usar SOAP como protocolo para comunicación

NFR-2	El sistema debe usar SOAP como protocolo para comunicación
Descripción	El sistema debe hacer uso de protocolos que garanticen la portabilidad de la información, en este caso que sean compatibles con otros sistemas que usen servicios web
Importancia	Alta

Tabla 44 Usar Android

NFR-5	El sistema debe usar Android
Descripción	El sistema debe hacer uso de un framework y/o un lenguaje que esté disponible en la mayoría de dispositivos móviles, que además ofrece una gran portabilidad y facilidad en las interfaces
Importancia	Alta

Tabla 45 La información del sistema se debe mantener en un servidor

NFR-9	La información del sistema se debe mantener en un servidor
Descripción	El sistema debe conectarse a un servidor que mantiene segura y centralizada toda la información del sistema(paciente, citas, servicios, etc.), debido a lo importante y crítico de esta
Importancia	Alta

Tabla 46 La información del sistema se debe manejar con el DBMS ORACLE

NFR-10	La información del sistema se debe manejar con el DBMS ORACLE
Descripción	Toda la información del sistema debe estar registrara en una base de datos relacional, orientada a objetos, que además de garantizar la confiabilidad y la portabilidad da una gran ventaja de tiempo al momento de desarrollar.

Importancia	Alta
-------------	------

Tabla 47 La comunicación debe hacerse por medio de web services

NFR-11	La comunicación debe hacerse por medio de web services
Descripción	El sistema web y el sistema móvil deben comunicarse a través de web services lo que garantiza la portabilidad y la integrabilidad del sistema.
Importancia	Alta

Tabla 48 El sistema web se debe hacer usando J2EE

NFR-12	El sistema web se debe hacer usando J2EE
Descripción	El sistema web se hará usando un framework web reconocido por su agilidad al momento de desarrollo y la calidad de los productos que se crean además de usar un lenguaje de programación que tiene una curva de aprendizaje mucho más corta que otros en el mercado, funciona todo OS y con soporte para web services (soaplib.py)
Importancia	Alta

Tabla 49 El sistema deberá ser probado sobre equipos reales

NFR-13	El sistema deberá ser probado sobre equipos reales
Descripción	El sistema se probará sobre equipos reales (celulares), que tengan el sistema ANDROID mencionados, y que tengan en lo posible un teclado QWERTY, además de conexión a internet, para demostrar su versatilidad.
Importancia	Alta

Tabla 50 El sistema deberá ser sometido a pruebas

NFR-14	El sistema deberá ser sometido a pruebas
Descripción	El sistema será probado mediante una metodología de pruebas que permita al desarrollador usar las pruebas para generar el código y además permita al usuario final hacer las pruebas de usabilidad metodología por definir
Importancia	Alta

4.10 Diagrama de Casos de uso

4.10.1 Diagrama Caso de Uso General

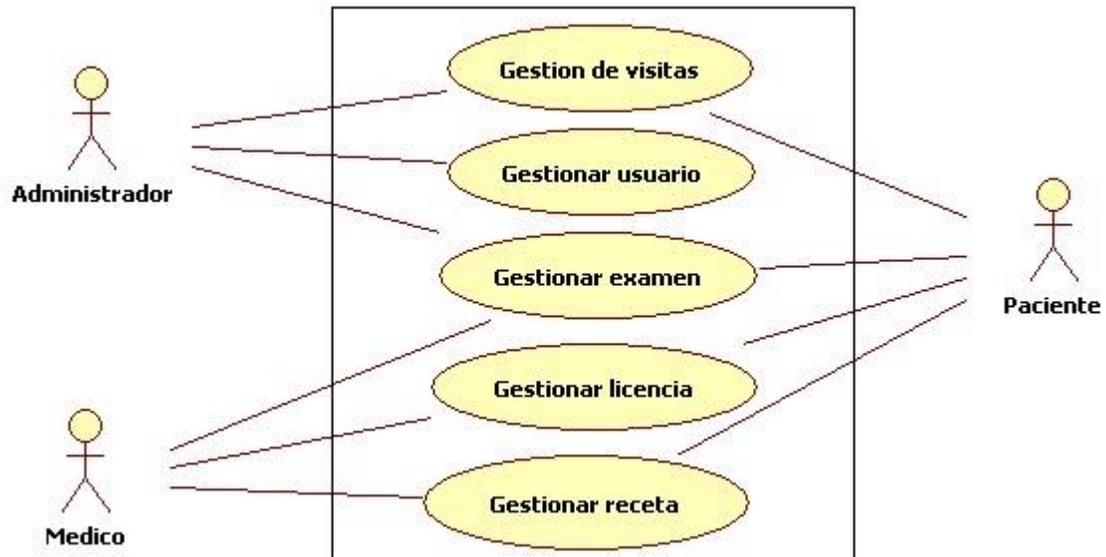


Ilustración 12 Diagrama Caso de Uso General

4.10.2 Caso de Uso Gestionar Usuario:

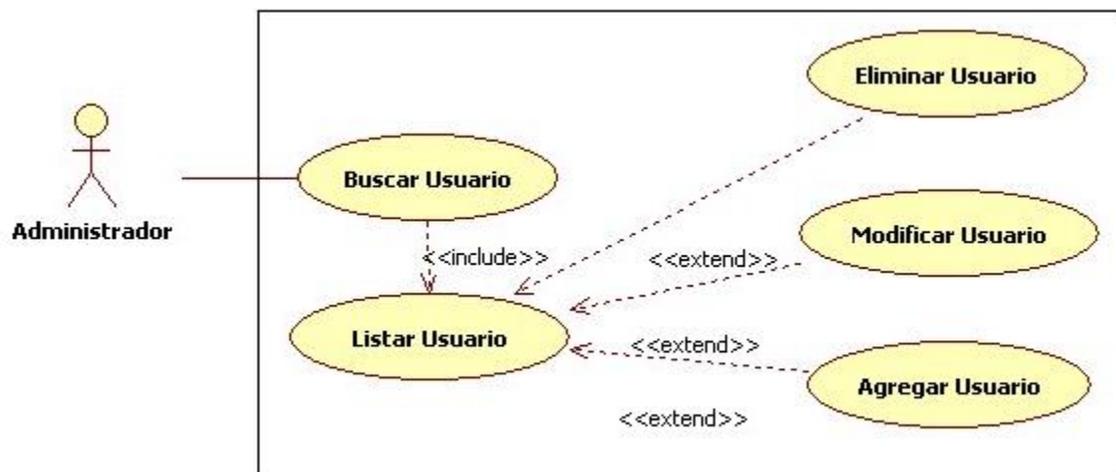


Ilustración 13 Caso de Uso Gestionar Usuario

4.10.3 Caso de Uso Gestionar Visita:

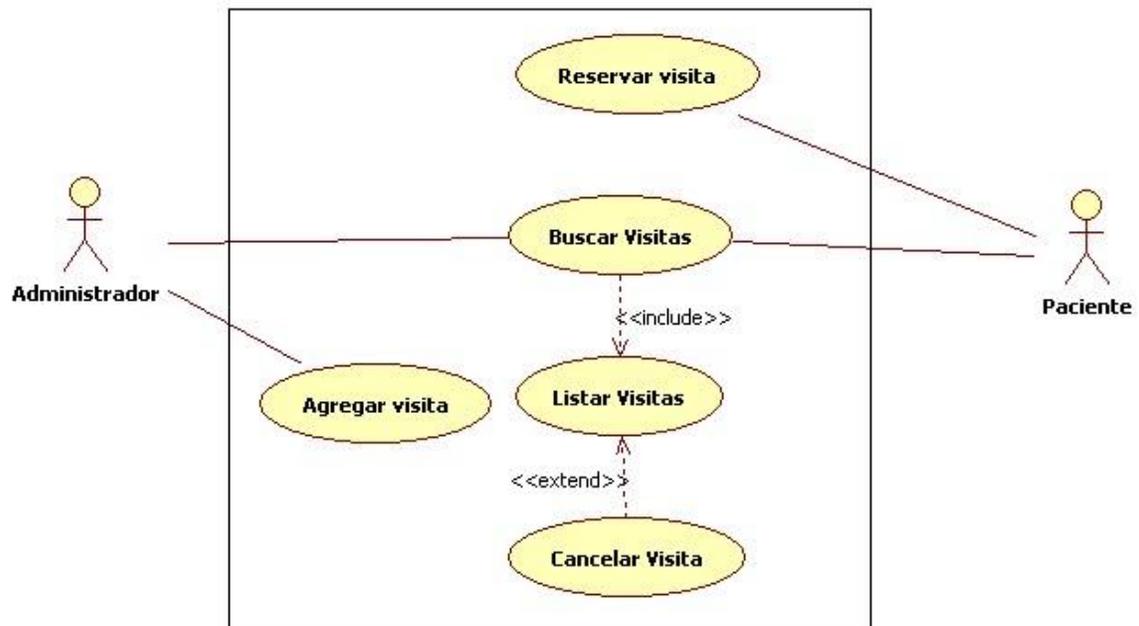


Ilustración 14 Caso de Uso Gestionar Visita

4.10.4 Caso de Uso Gestionar Receta:

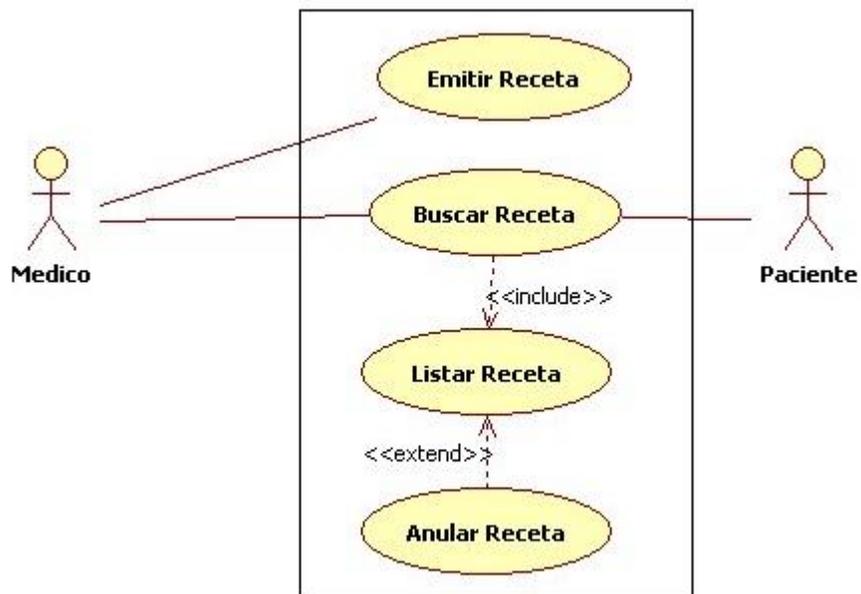


Ilustración 15 Caso de Uso Gestionar Receta

4.10.5 Caso de Uso Gestionar Examen:

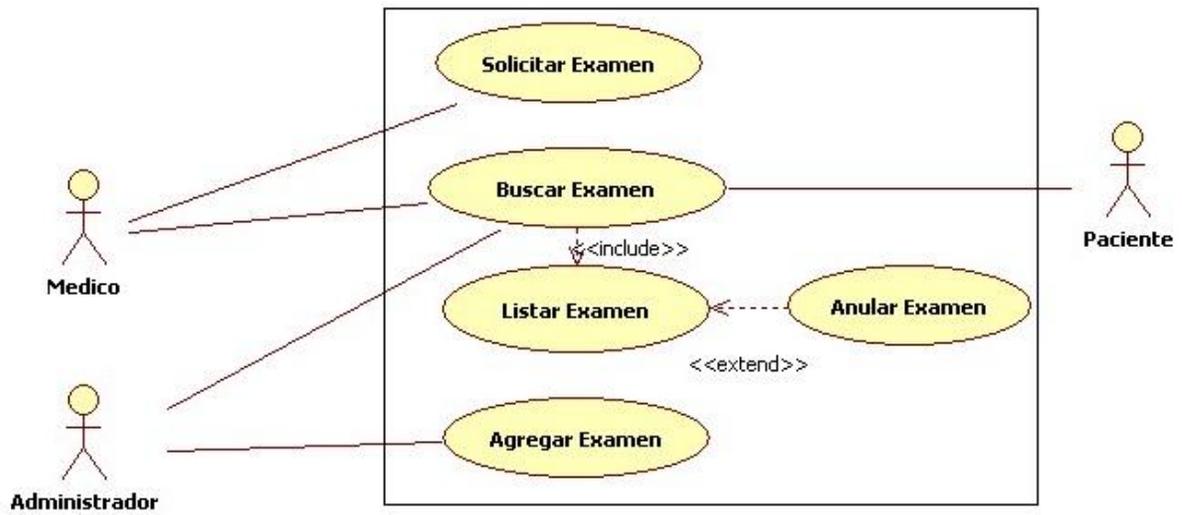


Ilustración 16 Caso de Uso Gestionar Examen

4.10.6 Caso de Uso Gestionar Licencia:

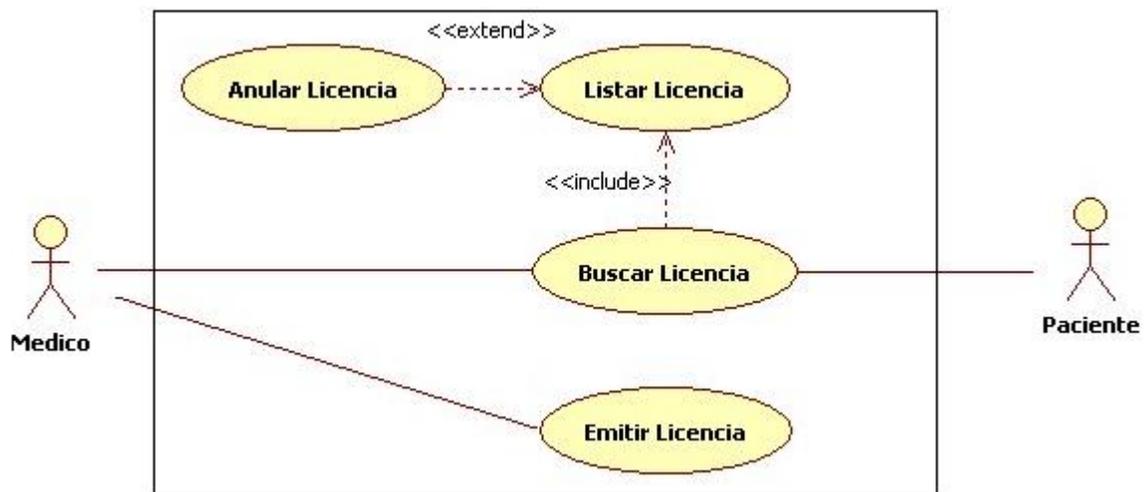


Ilustración 17 Caso de Uso Gestionar Licencia

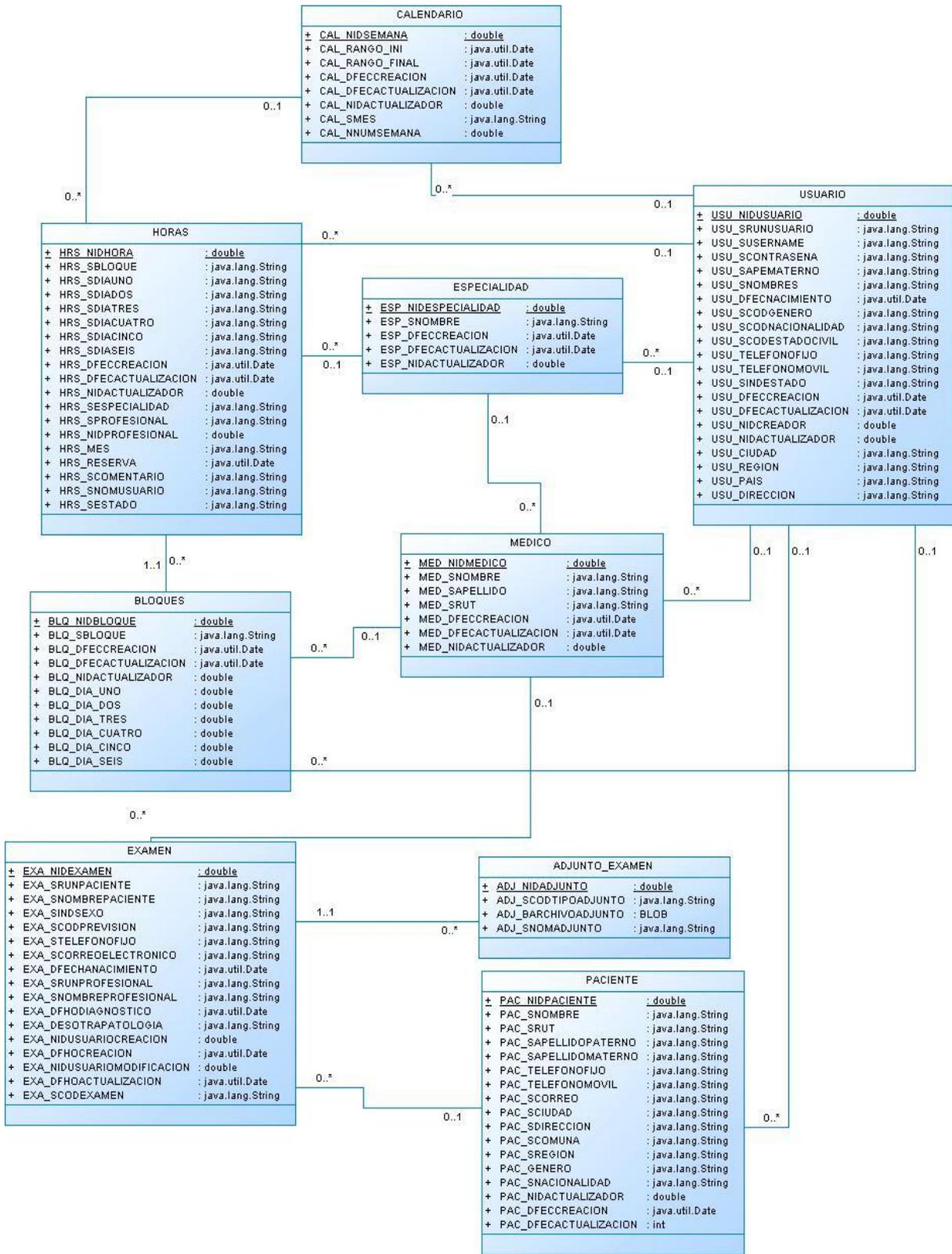
4.11 Diagrama de clases

4.11.1 Sistema web

Gracias al DBMS que usamos podemos definir modelos OO y estos se verán especificados en forma física en una base de datos relacional, por este motivo las clases descritas en el modelo E-R de la base datos son diseñadas en del sistema web y a estas clases además se le agregan los métodos necesarios para lograr las características requeridas, sin embargo existen otras clases emergentes relacionadas sobre todo con la comunicación, como por ejemplo la clase `generador_wsdl`, que se encarga al momento de ser instanciada de agregar todas las funciones al WSDL.

Gracias a la arquitectura del framework de desarrollo web no son necesarias clases para la interfaz gráfica, ya que todas las funcionalidades son implementadas a nivel del modelo y la forma de desplegarlas y controlar la información a desplegar es realizada a otro nivel; estas vistas son en primera instancia definidas por una “libreta de direcciones” (`urls.py`) que describe que función (`views.py`) se va a encargar de consultar los métodos de la lógica del negocio (`models.py`), y esta función a la vez se encarga de llamar al template que dibuja la interfaz gráfica.

Figura 1 Diagrama Inicial de clases Web



4.11.2 Sistema Móvil

Después de un análisis exhaustivo del dominio, de los componentes fundamentales dentro del sistema y de analizar su relaciones, en base al modelo global del sistema se realizó una abstracción de todas aquellas entidades que eran de importancia y se debían modelar dentro del sistema, como resultado de esto se diseñaron las clases que modelan el sistema móvil.

Realizando una separación de clases se modelaron 4 subsistemas que mostraron ser relevantes: el paquete de lógica donde se abstraen y representan la mayoría de entidades del sistema; El paquete de presentación que contiene todas aquellas clases que proveen interfaz al usuario para interactuar con el sistema; El paquete de persistencia que contiene las clases encargadas de manipular aquella información que el sistema debe almacenar; Y por último el paquete de conexión, que contiene las clases que se encargan de manejar los diferentes protocolos de conexión del sistema Móvil de Telemedicina con el modulo Web de Telemedicina.

Aunque se genera un modelo de clases para el proyecto, no significa que sea único e inalterable, es muy probable en este tipo de desarrollos que se presenten cambios al diseño que no fueron contemplados cuando se crearon.

Ilustración 18 Diagrama de clases Conexión Sistema Móvil

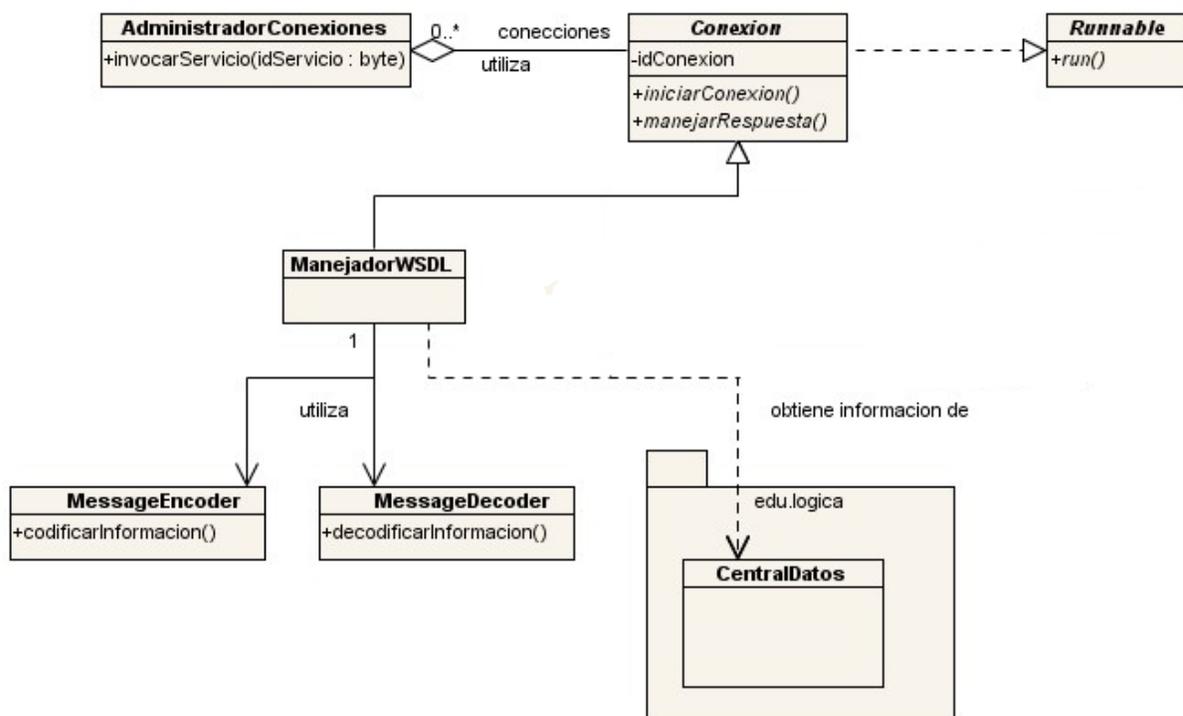
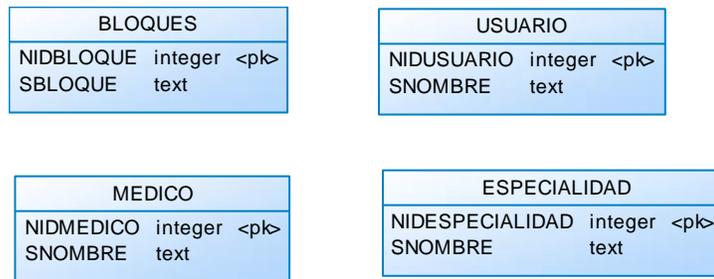


Ilustración 19 Diagrama de Clases Móvil



4.12 Diagrama de Secuencias

4.12.1 Diagrama de Secuencia Inicio de Sesión Modulo Web

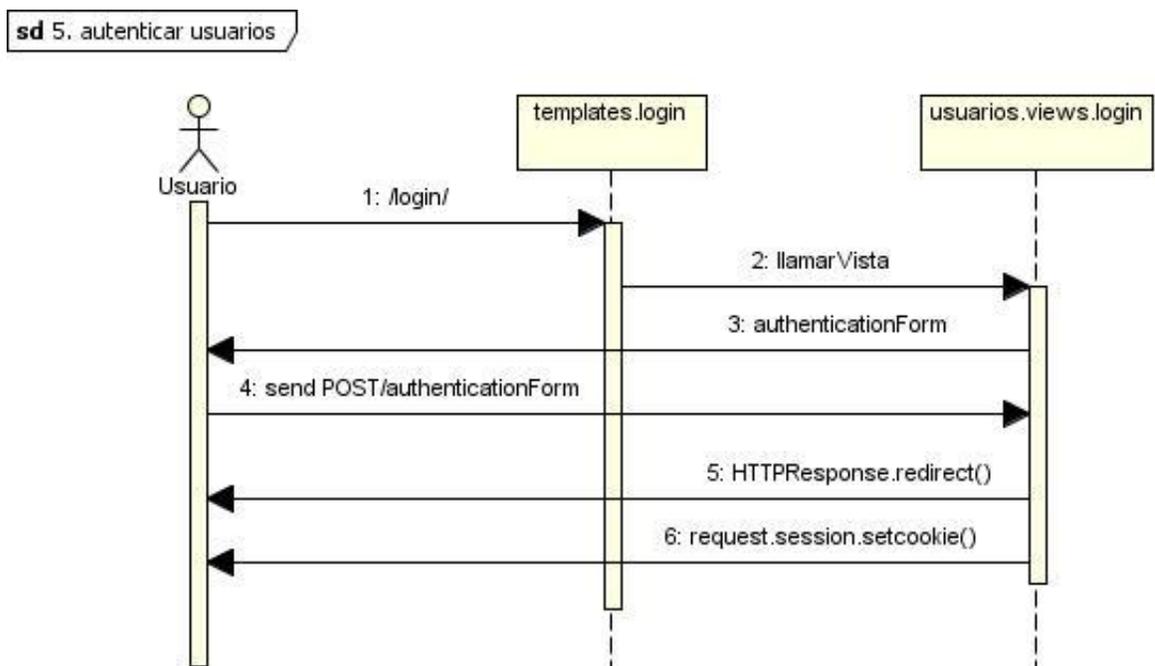


Ilustración 20 Inicio de Sesión Modulo Web

4.12.2 Diagrama de Secuencia Inicio de Sesión Modulo Móvil

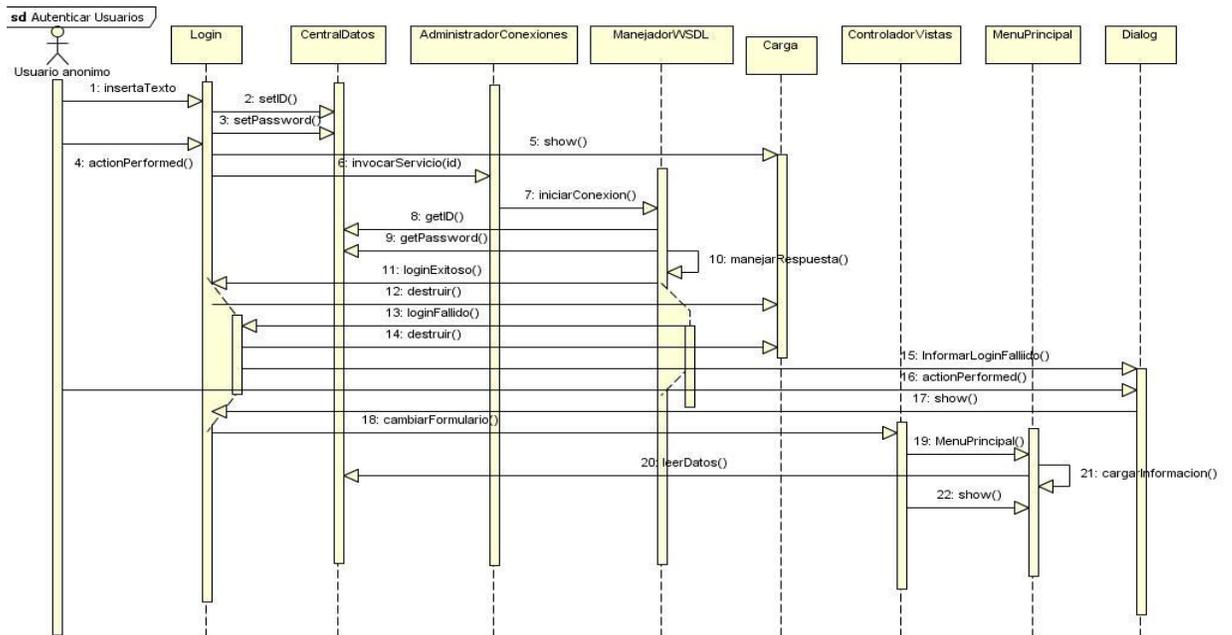


Ilustración 21 Inicio de Sesión Modulo Móvil

4.12.3 Diagrama de Secuencia Gestionar usuarios

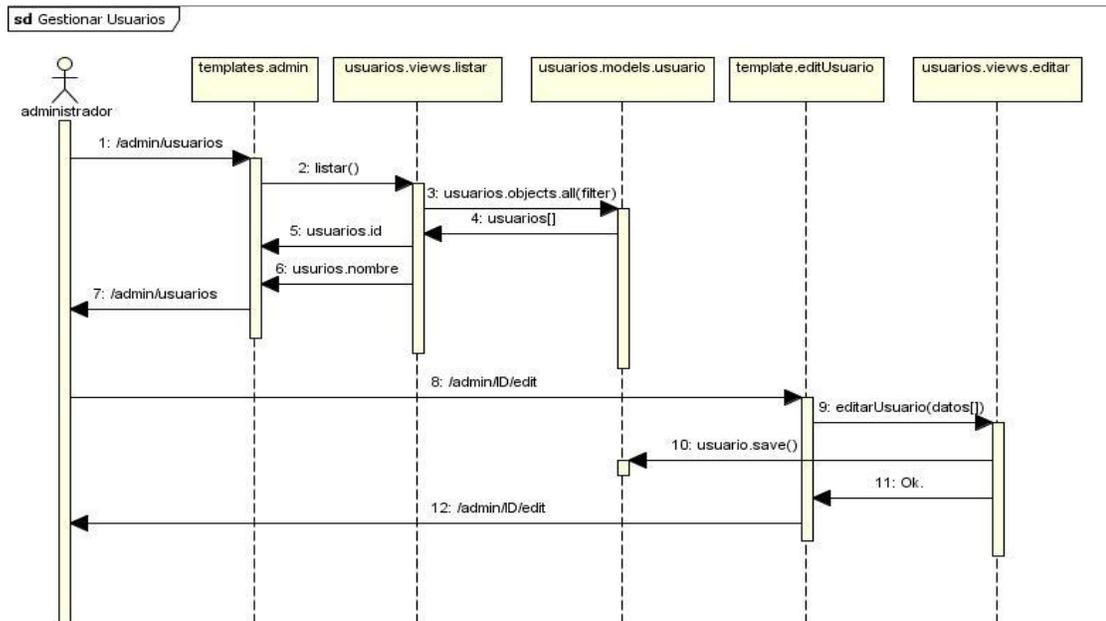


Ilustración 22 Gestionar usuario

4.12.4 Desplegar agenda web

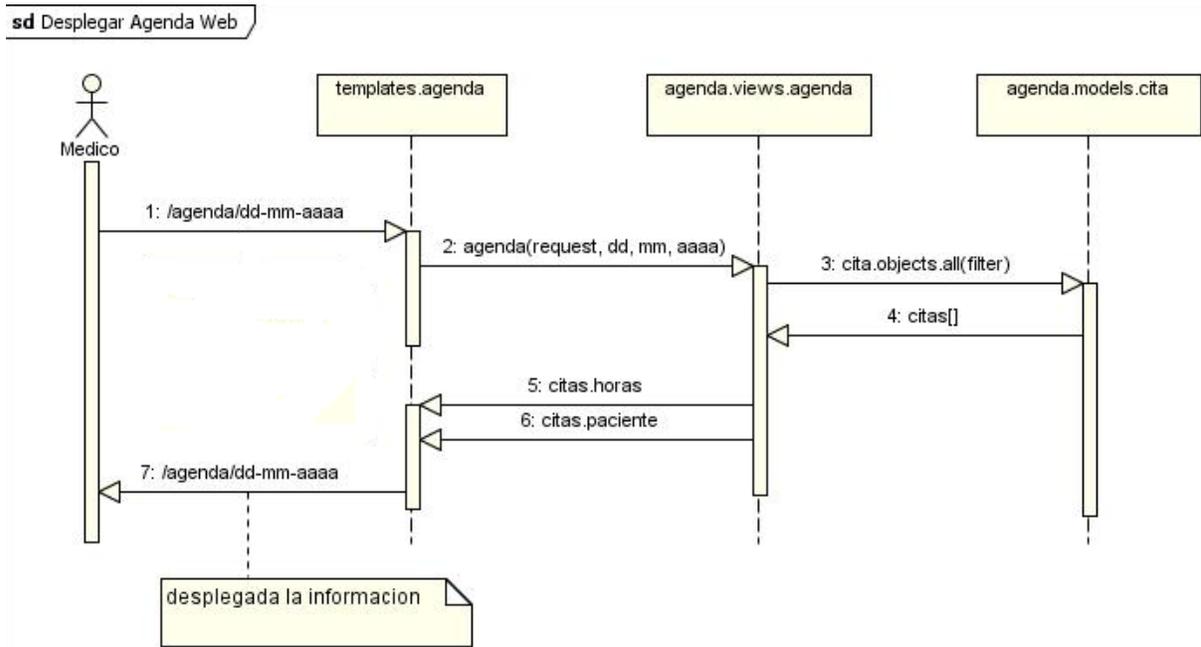


Ilustración 23 Agenda Web

4.12.5 Diagramas de Actividad

Tanto la aplicación Web como la aplicación móvil para Pacientes, permiten al usuario fijar una cita médica de acuerdo a la especialidad que este requiera. La cita médica estará condicionada a la disponibilidad de horarios del profesional de la salud requerido.

4.12.6 Diagrama de Actividad Agendar Cita Médica

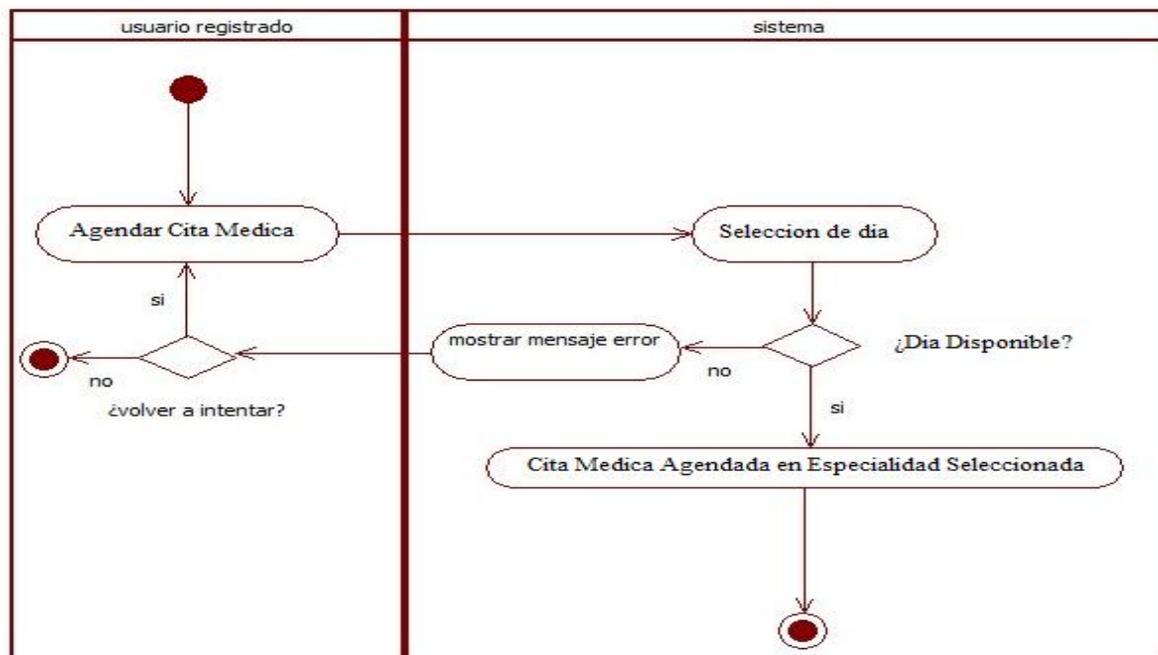


Ilustración 24 Agendar Cita Médica

4.12.7 Diagrama de Actividad Guardar Archivos Adjuntos

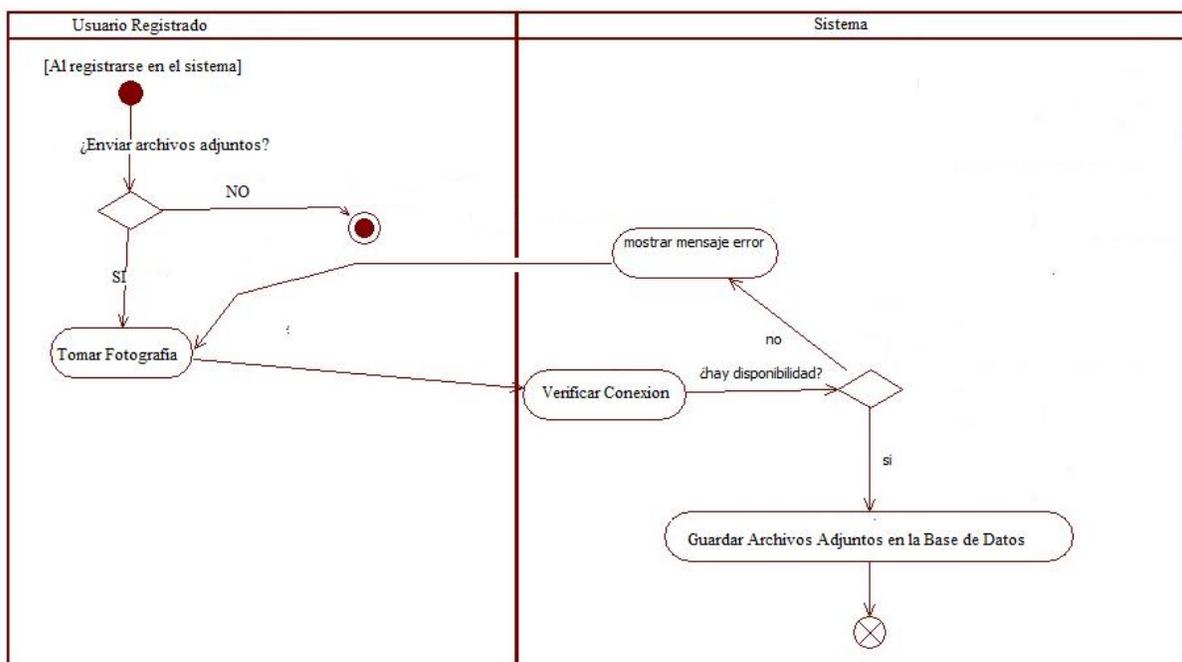


Ilustración 25 Diagrama de Actividad Guardar Archivos Adjuntos

4.13 Diseño de la base de datos

Basados en la información recogida durante la fase de análisis del dominio, y en los requerimientos de información levantados, se diseñó la primera versión del modelo Relacional y el modelo físico de la base de datos del proyecto.

4.13.1 Diseño conceptual

4.13.2 Usuarios del sistema

- **Administrador general:** Es el Súper-Usuario solo tiene acceso al módulo Web y está encargado de crear y modificar entidades prestadoras, y entidades administradoras de salud, también se encarga de la crearon o modificación de los usuario de tipo “usuario entidad administradora” y de tipo “usuario entidad prestadora”, mantener la aplicación y mirar los logs del sistema.
- **Personal médico:** estas personas pueden ser médicos, paramédicos, enfermeras, o auxiliares en general, se encargan de actualizar la información de las historias clínicas de los pacientes, los médicos pueden realizar publicaciones de casos anónimos.

- **Empleado citas:** Las entidades administradoras de salud, organizan las citas de sus afiliados con el personal médico de una de las entidades prestadoras con quienes trabajan. Se encarga de gestionar todo el proceso de administración de citas médicas.

4.13.3 Entidades

- **Adjunto:** la entidad adjunto representa el contenido multimedia agregado a las historias clínicas, a través de los eventos médicos generados sobre la misma.
- **Administrador General:** en esta entidad se modelan los datos referentes al usuario llamado administrador general.
- **Alarma:** la entidad alarma es usada para guardar la información referente a las citas médicas, y exámenes asignados a los pacientes.
- **Análisis médico:** describe los comentarios hechos por otros médicos a los casos anónimos publicados en el portal.
- **Cita:** la entidad cita representa la información acerca de las visitas que realiza un médico a un paciente, se compone de una hora, y un lugar específico.
- **Empleado Citas:** en esta entidad se modela la información referente a los datos de usuario del empleado que se encarga de asignar y modificar las visitas médicas.
- **Examen:** en la entidad examen se almacena toda la información referente a los exámenes para clínicos asignados a los pacientes, fecha, tipo de examen y resultados del mismo.
- **Fecha Alarma:** en esta entidad se guardan todas las fechas programadas para una alarma específica asignada a un paciente, la información contenida aquí se usará para poder avisar a un paciente acerca de sus visitas o los resultados de sus exámenes.
- **Log:** entidad creada con el fin de llevar un control de todas las actividades que realiza un usuario dentro del sistema, aquí se guardará información con respecto a los cambios hechos, el usuario que los hizo, y la fecha en que los hizo.
- **Motivo y diagnóstico:** la entidad motivo y diagnóstico, como su nombre lo indica, es usada para almacenar información concerniente a los motivos que tiene un paciente para ir a una cita médica, y al diagnóstico dado por el médico, luego de los exámenes de rigor.
- **Paciente:** la entidad paciente modela la información necesaria para llevar a cabo una buena gestión de los datos de las personas registradas en el sistema, y garantizar la confidencialidad y unicidad de los mismos.
- **Persona Médico:** esta entidad modela la información de usuario relacionada con todo el personal médico (médicos, paramédicos, enfermeras, auxiliares).
- **Revisión Sistemas:** en esta entidad se almacena información referente a la revisión hecha por un médico, a todos los sistemas de un paciente durante una cita médica.
- **Usuario:** esta entidad es la generalización de todos los usuarios en el sistema, contiene la información primaria para todos los usuarios (documento de identificación, password, nombre, tipo de sangre, rol, etc.).

4.13.4 Diagramas relacional

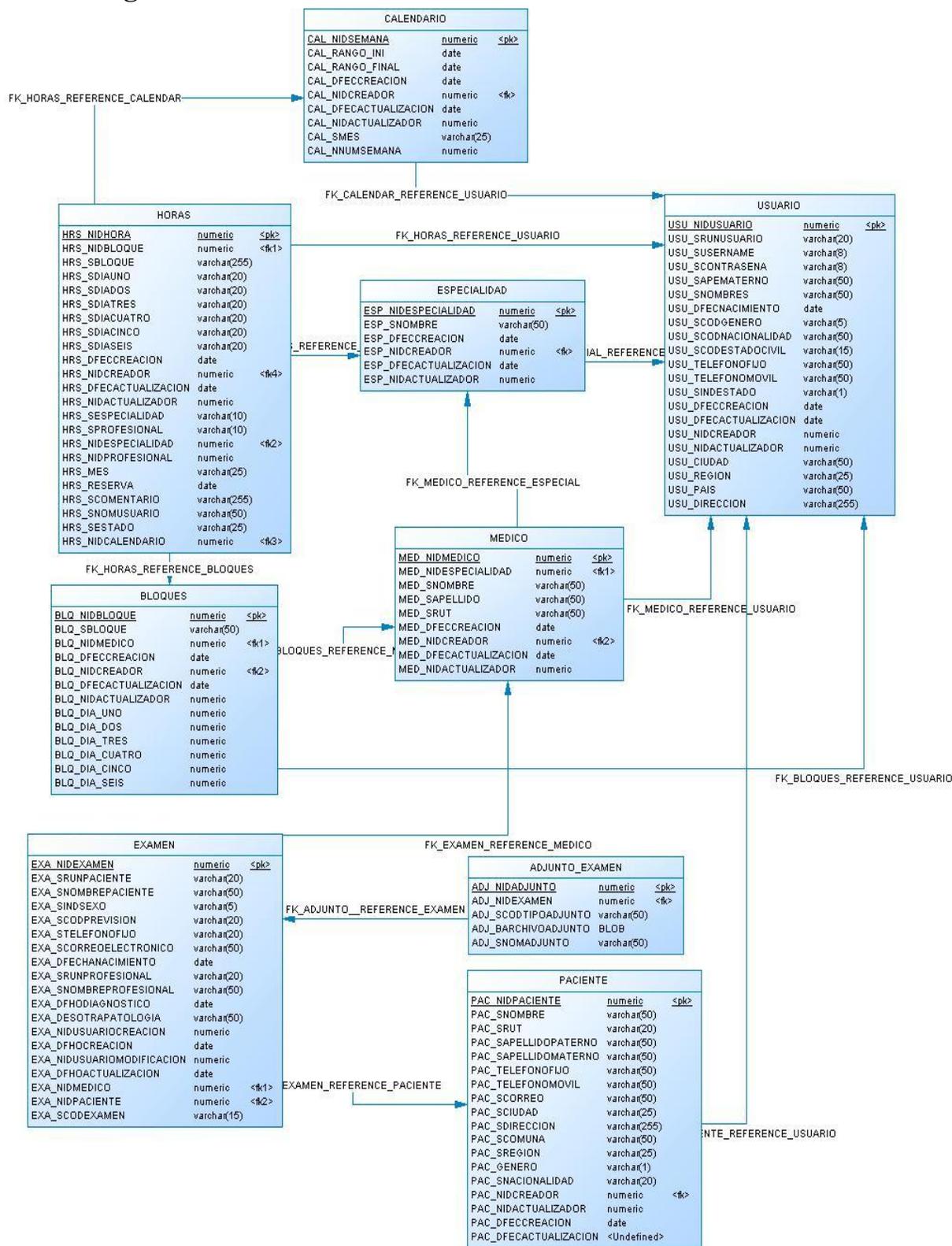


Ilustración 26 Diagrama Relacional

5 Capítulo Quinto: Prototipo modulo Paciente

5.1 Web

Dentro de la aplicación web ya está construido el módulo de pacientes completo, y es el que se detallara a continuación las interfaces, se presentaran 4 interfaces, que son ingresar al sistema, página de bienvenida, reservar hora y listar horas.

5.1.1 Ingresar

En la ilustración 27, se presenta la pantalla de bienvenida del Módulo Web al usuario invitado, que tiene la posibilidad de ingresar al sistema y autenticarse con su Nombre de usuario y contraseña.

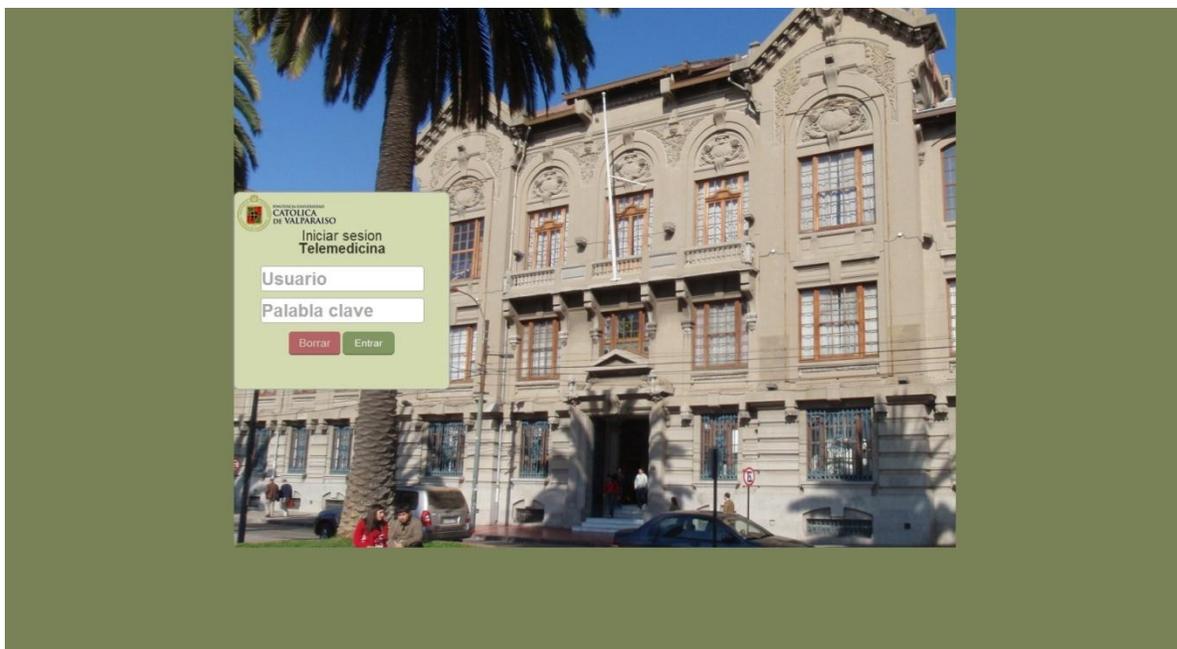


Ilustración 27 Ingresar Modulo Web

5.1.2 Bienvenida Modulo Web

Una vez que el usuario ha podido acceder con éxito al sistema Web de Telemedicina, este le ofrece una interfaz de bienvenida donde le comunica al usuario algunos datos como: Nombre de Usuario que se logeo, la fecha actual y los links a todas las opciones que maneja el modulo.



Ilustración 28 Bienvenidos

5.1.3 Reservar Hora

El modulo Web, al igual que el modulo móvil, permite agendar citas médicas al usuario que este registrado en el sistema como paciente, en esta interfaz, se deberán proporcionar datos como la especialidad a la que el paciente requiere atención médica, el nombre del médico asociado a la especialidad seleccionada y el bloque de horas en el que el profesional tiene disponibilidad para la atención.

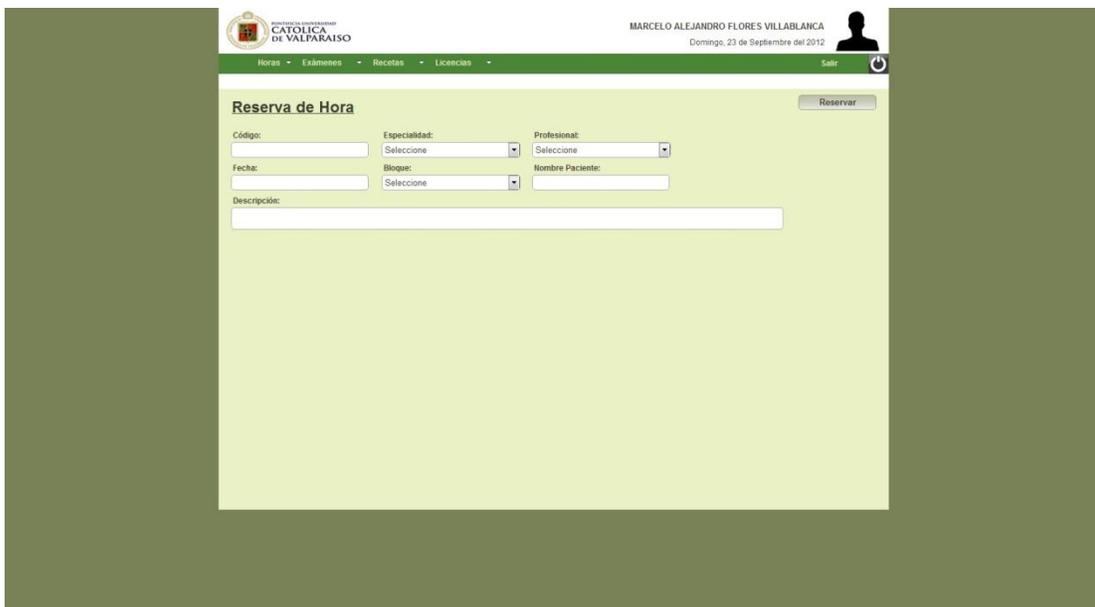


Ilustración 29 Reservar horas

5.1.4 Listar Hora

El modulo Web, permite al profesional que ha accedido al sistema, conocer todas las fechas en la que los usuarios pacientes han agendado alguna cita médica. Esta agenda despliega información de las citas médicas, agrupadas por semana, en las cuales el profesional podrá conocer también el detalle de cada una de citas al hacer doble click sobre alguna de ellas. En el detalle de las citas médicas, el profesional podrá conocer los datos personales del paciente que agendo la cita y conocer el bloque de atencion.

Bloque	Fecha	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
08:30 - 09:15	2012-09-24 00:00	Daniel Barraza					
12:15 - 13:00	2012-09-26 00:00			Marcelo Flores			

Ilustración 30 listar horas

5.2 Móvil

La parte móvil está dividida entre paciente y medico visitador, la parte del paciente permite al usuario que este registrado en el sistema, agendar alguna cita médica para la especialidad que este requiera, ofreciéndole además, los profesionales de la salud, agrupados por especialidad y los horarios disponibles de cada uno de ellos.

La aplicación móvil para médico, permite fotografiar y registrar documentos que pueden ser del tipo: licencia, examen o receta y almacenarlos en la base de datos central del sistema, con respaldo en la base de datos del dispositivo móvil, para su posterior validación legal.

5.2.1 Ingresar

La ilustración 31, muestra la pantalla de inicio de sesión del módulo móvil, tanto para su versión de paciente, como para su versión de médico. Una vez registrado, podrá acceder al detalle de las funciones proporcionadas por la aplicación.

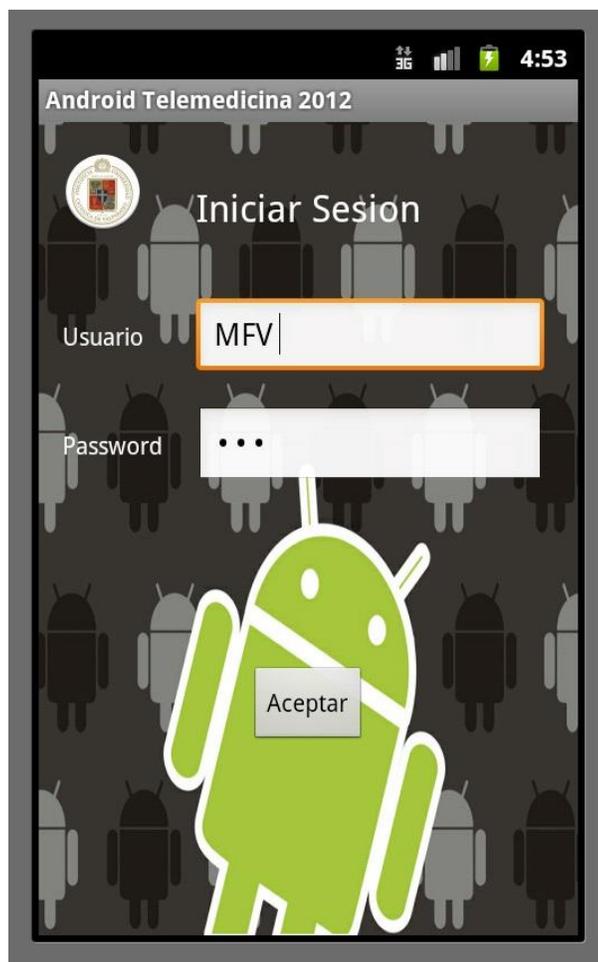


Ilustración 31 ingresar

5.2.2 Bienvenida Modulo Móvil

La ilustración 32, muestra la pantalla que despliega el modulo móvil una vez que se ha accedido correctamente al sistema. En esta pantalla se despliega un aviso sobre las funcionalidades a las que el usuario registrado podrá acceder.

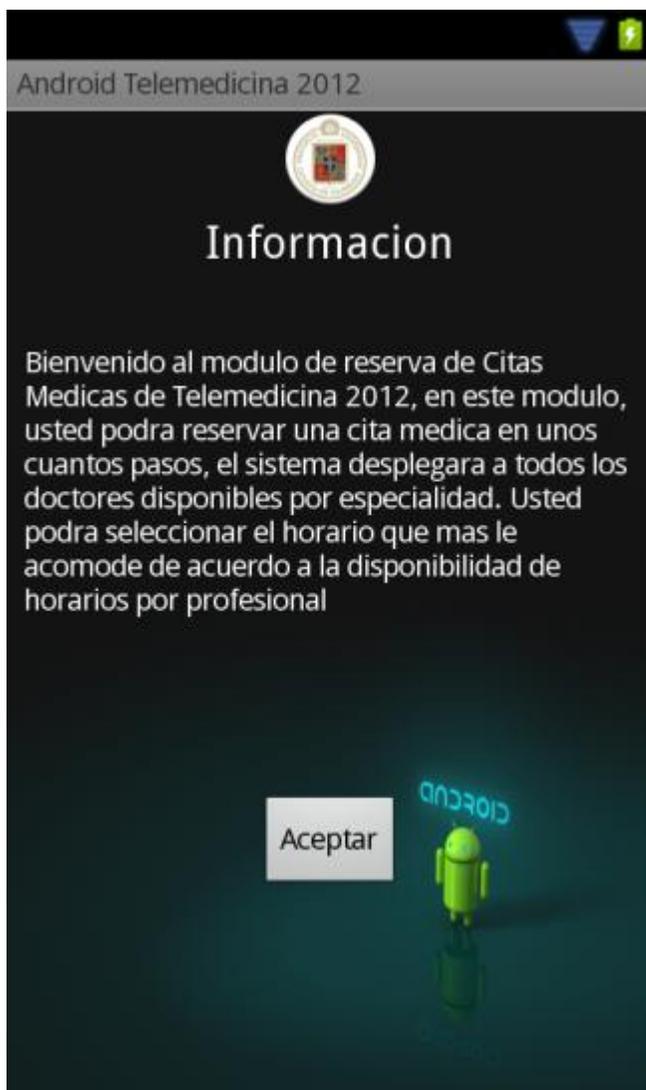


Ilustración 32 Bienvenida Móvil

5.2.3 Selección de Especialidad

La aplicación móvil para pacientes, permite al usuario fijar una cita médica de acuerdo a la especialidad que este requiera atención. La cita médica estará condicionada a la disponibilidad de horarios del profesional de la salud requerido.



Ilustración 33 Selección de Especialidad

5.2.4 Selección de Día

Una vez que el usuario paciente ha seleccionado la especialidad y el profesional de la salud requerido, el sistema le ofrece este, la selección de día en el deseo agendar la cita médica. Una vez seleccionado el día, el sistema consultara a la Base de Datos central y desplegara todos los bloques de horario disponible asociado al profesional para el día seleccionado.



Ilustración 34 Selección de día

5.2.5 Resultado Final

Luego de las selecciones anteriores, el sistema desplegara al usuario una pantalla final, donde le indicara a este que su cita médica fue agendada exitosamente



Ilustración 35 Resultado Final

6 Capítulo Sexto: Pruebas

En esta sección se hablara de algunas pruebas desarrolladas a los módulos, dentro de las pruebas, que se alcanzaron a realizar, están las de los módulos de ingreso y perfilamiento de las aplicaciones tanto Web como Móvil, las pruebas futuras abarcan todos los módulos y web servicios, también contemplan pruebas de usabilidad.

6.1.1 Pruebas unitarias

Es necesario diseñar pruebas unitarias para el subsistema web y el subsistema móvil de forma independiente.

6.1.2 Sistema web

1. Pruebas de interfaz: Se verificarán los flujos de navegación desde la ventana de login a cada tipo de usuario que puede ser validado dentro del sistema.
2. Pruebas de la estructura de datos locales: Los usuarios activos en el sistema deben corresponder a usuarios que han hecho login valido.
3. Condiciones límite: Se verificarán casos donde un mismo usuario ingrese al sistema desde distintas terminales.
4. Caminos independientes: Se validará de igual manera que en la prueba de interfaz. Adicional de las alertas necesarias cuando el usuario no se autentica de forma correcta dentro del sistema.
5. Camino de manejos de errores: Se corregirán todos los errores encontrados durante la evaluación de los casos de prueba, ya que esta es una característica crítica dentro del sistema que garantiza la confiabilidad y seguridad de la información dentro del sistema.

Tabla 51 Caso de prueba sistema web

Prueba	Éxito
No se ingresan datos en los campos y se hace login.	El usuario entra al sistema.
Se debe ingresar una identificación válida y contraseña errónea	La aplicación permite el login del usuario con la identificación descrita.
Se debe ingresar una identificación errónea y una contraseña de usuario valida.	La aplicación permite el login de usuario.
Se debe ingresar datos de identificación y contraseña erróneas	La aplicación permite el login de usuario.

6.1.3 Pruebas de integración

Este módulo se integrara por completo con el módulo anterior de creación de usuarios, de la misma manera que el anterior en forma ascendente.

Tabla 52 Caso de prueba sistema móvil 1

Prueba	Éxito
Se ingresara un usuario válido dentro del sistema desde distintas terminales web.	Se puede navegar dentro del sistema como usuario autenticado desde cada una de las terminales web.
Se hará login por cada tipo de usuario y se verificara la información personal de cada uno.	La información no corresponde al tipo y rol del usuario autenticado.
Se modificara la contraseña de un usuario, se terminara la sesión, y se conectara con el mismo usuario y la antigua contraseña en otra terminal.	El usuario entra al sistema de forma exitosa.
Se modificara la contraseña de un usuario, se terminara la sesión, y se conectara con el mismo usuario y la nueva contraseña en otra terminal.	El usuario no logra acceder al sistema.

6.1.4 Sistema móvil

1. Pruebas de interfaz: Se verificarán los flujos de navegación desde la ventana de login al menú principal respectivo de auxiliar médico y médico general.
2. Pruebas de la estructura de datos locales: Los usuarios activos en el sistema deben corresponder a usuarios que han hecho login valido, se puede hacer esta verificación dentro de la base de datos en tiempo de ejecución.
3. Condiciones límite: Realizar login desde la misma terminal móvil, varias veces, saliendo y entrando al sistema sin cerrar la aplicación para verificar memoria en el dispositivo.
4. Caminos independientes: Se validara de igual manera que en la prueba de interfaz. Adicional de las alertas necesarias cuando el usuario no se autentica de forma correcta dentro del sistema.
5. Camino de manejos de errores: Se corregirán todos los errores encontrados durante la evaluación de los casos de prueba, ya que esta es una característica crítica dentro del sistema que garantiza la confiabilidad y seguridad de la información dentro del sistema.

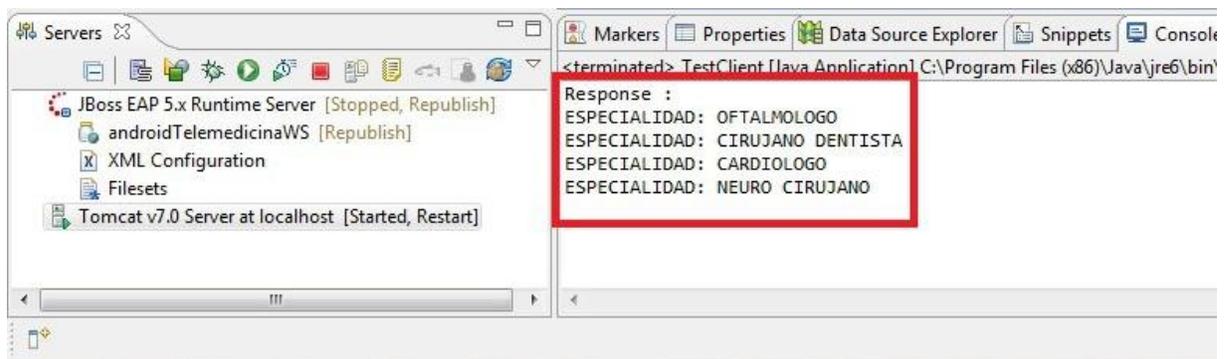


Ilustración 36 Respuesta web service

Tabla 53 Caso de prueba sistema móvil 2

Prueba	Éxito
Se debe ingresar ambos campos nulos en el formulario de login.	La aplicación entra al menú principal.
Se debe ingresar una identificación válida y contraseña errónea	La aplicación permite el login del usuario con la identificación descrita.
Se debe ingresar una identificación errónea y una contraseña de usuario valida.	La aplicación permite el login de usuario.
Se debe ingresar datos de identificación y contraseña erróneas	La aplicación permite el login de usuario.
Se iniciara sesión y se cerrara sesión varias veces en el sistema sin cerrar la aplicación.	La aplicación se torna lenta, el estado de la memoria en uso aumenta. (Para esta prueba se puede usar el memory manager del WTK).

7 Capítulo Séptimo: Conclusiones y Trabajo Futuro

El proyecto de Telemedicina planteo un prototipo funcional que busca con su implementación:

- Promover la estandarización del manejo de historias clínicas en el país para que se pueda emplear de forma efectiva los beneficios de un estándar único para el manejo de la información clínica y el intercambio de los datos médicos de los pacientes chilenos.
- La puesta en producción de un sistema de información médico que sea compatible con otros sistemas internacionales gracias al estándar de mensajes Soap, dándole entrada a compañías prestadoras y/o promotoras de salud nacional e internacional.
- Generar un módulo de internacionalización para el cliente móvil que le permita ser usado en otros países donde la implementación del estándar de intercambio de información médica por dispositivos móviles es un éxito.

A futuro se pueden generar:

- Módulos especiales y con formularios e información específica para especialistas, por ejemplo fisioterapeutas o personal médico de terapia respiratoria debido a que muchos de estos profesionales suelen hacer sus consultas en el lugar donde se encuentra el paciente.
- Para mayor funcionalidad a aquellos médicos que hacen visitas remotas o atienden pacientes en lugares alejados, sería mucho conveniente poder acceder a todo tipo de información médica a través del teléfono celular o Tablet y no limitarse solo a acceder a recetas o agendar citas médicas.
- Generar un módulo que permita la transmisión en vivo de audio y video desde el sitio desde donde se está atendiendo al paciente, incluso georeferenciando dicha transmisión.

A la velocidad a la que está avanzando la tecnología móvil la implementación de sistemas como estos se vuelve mucho más sencilla y provechosa, ya que por ejemplo se pueden implementar capturas de videos, de imágenes de altísima resolución, traspaso de imágenes diagnosticas por dispositivos bluetooth entre otros.

De acuerdo con el actual estado del arte se puede generar un módulo que permita al personal médico investigar o acceder a conocimiento en línea desde el dispositivo móvil. (Vademécum, términos médicos, estándares completos de medicina, etc)

Luego de realizar un análisis más detallado sobre el dominio del proyecto, las leyes rigentes en Chile, el uso del estándar SOAP para mensajes entre aplicaciones, el marco de una historia clínica completa y todo lo que conlleva su tratamiento en conjunto con las entrevistas desarrolladas, y los requerimientos levantados, se vio en la necesidad de replantear todos los alcances del proyecto pues la meta que se planteó en principio era demasiado ambiciosa y

difícil de lograr en los tiempos estimados para el desarrollo del prototipo. Con el nuevo planteamiento de los alcances, se tuvo una visión más clara del proyecto y mucho más consistente de acuerdo a lo que se quería lograr con respecto a la idea original del proyecto

A medida que se avanzó en el trabajo del proyecto, se fue encontrando con nuevas limitaciones que en ningún momento durante el planteamiento del proyecto se llegó a contemplar, como por ejemplo la gran cantidad de estándares y normatividades que involucra el estándar SOAP, o la cantidad de información que puede tratarse al momento de realizar un diagnóstico médico, esto generó nuevas limitaciones que se tuvo que contemplar y solucionar sobre la marcha, lo cual se consideró como inapropiado dentro del contexto de trabajo pues cada limitante nueva genera nuevas complicaciones y nuevos “costos” dentro del proyecto, costos que afortunadamente fueron tratables y solucionables fácilmente, pero que en otro contexto podrían conllevar a la cancelación del proyecto. Como conclusión a todo esto se considera que el proceso de limitación de un proyecto debe requerir un mayor esfuerzo de investigación y tiempo, así como un estudio del contexto general que abarca el proyecto, con el fin de solventar todas las limitaciones que puedan presentarse y desarrollar el proyecto de forma óptima y sin riesgos de alto nivel.

Con respecto a la metodología de desarrollo usada se puede decir que la experiencia fue en principio un tanto complicado, pues hubo que enfrentarse a contextos y definiciones completamente nuevas. En ocasiones muchos conceptos no eran bien interpretados o eran ambiguos, fue necesario plantear varias veces el proyecto y leer bastante documentación sobre la metodología para poder aclarar todos los inconvenientes que se presentaban. Dicha experiencia con el paso del tiempo fue evolucionando hasta hacerse bastante enriquecedora y fácil de manejar y comprender. De la experiencia de trabajo con la metodología se puede decir que el modelo mixto entre el modelo evolutivo y el modelo en cascada, se puede implementar mucho mejor en proyectos cortos y equipo de trabajo reducidos, pues con proyectos bastantes grandes se haría mucho más tediosa la labor de diseño, desarrollo y documentación de cada característica. También se puede hacer énfasis en que la definición de los conjuntos de características (features set) así como las características (features) que componen cada conjunto es uno de los procesos más vitales dentro de este modelo mixto, pues si no se presta la atención adecuada se corre el riesgo de definir características bastante granulares lo cual extendería enormemente la cantidad de iteraciones que deben hacerse, o en caso contrario se pueden definir características bastante amplias lo cual conllevaría a hacer que el desarrollo de cada iteración fuera un proceso bastante lento y complejo de manejar.

8 Capítulo Octavo: Bibliografía y Referencias

8.1 Bibliografía

- [1] MARTINEZ, José Alirio, Entrevista acerca de las partes de una Historia Clínica, Tecnoparque Agosto de 2008.
- [2] CALABRIA, Luis, Metodología FDD, Universidad ORT Uruguay. 2003.
- [3] CONTEL SEGURA, Joan Carles, Atención domiciliaria: Organización y práctica., Elsevier España, 2003 ISBN: 8445813617.
- [4] DURÁN, Amador, Metodología para la Elicitación de Requisitos de Sistemas Software, Universidad de Sevilla. 2002.
- [5] MORALES MACHUCA Carlos Andrés, Estado del arte: Servicios Web. Universidad Nacional de Colombia, Bogotá 2008

8.2 Referencias electrónicas

- [6] Diccionario Técnico de Medicina Estética [Web en línea] <http://www.med-estetica.com/Cientifica/Diccionario_Tecnico/index.html> [Consulta: 05-03-2008].
- [7] feature driven development [Web en línea]. <<http://www.featuredrivendevelopment.com/>>. [Consulta: 04-05-2009].
- [8] Origen y breve descripción de J2ME [Web en línea]. Disponible en:< http://leo.ugr.es/J2ME/INTRO/intro_2b.htm> Consulta [15-01-2008].
- [9] Resolución Número 4144 De 1999 [Web en línea]. <<http://www.idsn.gov.co/resoluciones/99/RES041441999.htm>>. [Consulta: 04-06-2009].
- [10] Mercado celular a madurar. [Web en línea]. <http://www.enter.com.co/enter2/ente2_cert/ARTICULO-WEB-NOTA_INTERIOR_2-35141010.html> Consulta [25-01-2008]
- [11] Google code a replacement for sourceforge.net [Artículo en línea] <<http://internetducttape.com/2006/07/28/google-code-project-hosting-a-replacement-for-sourceforge/>> Consulta [04-04-2009]
- [12] glosario de términos médicos [Web en línea] <<http://www2cero.blogspot.com/2008/01/glosario-de-trminos.html>> [Consulta: 13-12-2008].