



# **Universidad del Azuay**

**Facultad de Ciencias de la Administración**

**Escuela de Ingeniería de Sistemas**

## ***REGISTRO DE FICHAS MÉDICAS EN LA WEB***

**Trabajo de graduación previo a la obtención del título de  
Ingeniero de Sistemas**

**Autores:**

**Cristian Patricio Pulla Patiño**

**Verónica Elizabeth Serrano Cabrera**

**Director: Ing. Pablo Pintado Zumba**

**Cuenca, Ecuador**

**2010**

## **Dedicatorias**

Dedico el presente trabajo de graduación a mis queridos padres y hermanas, quienes con paciencia me han dado todo su apoyo incondicional a lo largo de mi vida.

Verónica Serrano.

A Dios porque por sobre todas las cosas me dio la fe y fuerza necesaria para seguir adelante.

A mis padres parte fundamental de mi vida, por el apoyo incesante día tras día para luchar por mis sueños.

A mi ñaña, a la Kata y al Mati, por estar siempre presentes para compartir los buenos y malos momentos.

A Lore, que tanto espero este día, por ser la mujer de mi vida, por su paciencia y por enseñarme que uno no se hace viejo con el tiempo, si no cuando deja de soñar.

Cristian Pulla.

## **Agradecimientos**

Agradezco a Dios por dejarme concluir una etapa más de mi vida; también doy las gracias a todos los profesores, quienes han sabido brindar sus conocimientos y experiencias; y gracias a éstos poder llegar a ser una buena profesional.

Verónica Serrano.

A todas las personas que apoyaron y creyeron en mí.

Mi más sincero agradecimiento a mi director de tesis Ing. Pablo Pintado por su sabiduría transmitida.

A todos y cada uno de los profesores que supieron compartir con nosotros no solo el conocimiento sino además experiencias de vida.

A mis familiares y amigos que aportaron con sus sugerencias y apoyo incondicional

A todas las personas que dedicaron su tiempo a realizar perfectas fuentes bibliográficas sin las cuales difícilmente se hubiese logrado realizar el trabajo.

Cristian Pulla.

## Índice de Contenidos

Dedicatoria .....	ii
Agradecimientos .....	iii
Índice de Contenidos.....	iv
Índice de Ilustraciones.....	vi
Índice de Anexos.....	viii
Resumen.....	ix
Abstract .....	x
Introducción .....	1
CAPÍTULO 1 .....	2
SITUACION ACTUAL DE LAS FICHAS CLINICAS.....	2
1.1 Introducción.....	2
1.2 Ficha clínica.....	3
1.2.1 Definición: .....	3
1.2.2 Partes de una ficha clínica .....	3
1.2.3 Descripción.....	3
1.2.4 Análisis general del manejo de las fichas clínicas.....	5
1.2.5 Análisis del software.....	8
1.3 Conclusión.....	18
CAPÍTULO 2 .....	19
FORMULACION Y PLANEACION DEL SISTEMA .....	19
2.1 Introducción.....	19
2.2 Identificación y análisis de requisitos.....	20
2.2.1 Requisitos de contenido .....	20
2.2.2 Requisitos funcionales .....	22
2.2.3 Requisitos no funcionales .....	23
2.2.4 Categorías de usuario.....	23
2.3 Ciclo de vida del proyecto.....	26
2.3.1 El proceso de programación XP .....	26
2.4 Definición de tareas.....	29

2.4.1 Definición de tiempos y recursos para cada tarea.....	30
2.5 Conclusión.....	32
CAPÍTULO 3 .....	33
ANÁLISIS DEL SISTEMA .....	33
3.1 Introducción.....	33
3.2 Requisitos para el análisis .....	34
3.3 Modelo de análisis.....	36
3.4 Modelo de contenido .....	36
3.5 Modelo de interacción .....	39
3.6 Modelo funcional.....	62
3.7 Modelo de configuración.....	63
3.8 Modelo de la Base de Datos .....	64
3.9 Conclusión.....	65
CAPÍTULO 4 .....	66
GESTION DE RIESGOS .....	66
4.1 Introducción.....	66
4.2 Identificación de riesgos.....	67
4.2.1 Del proyecto.....	67
4.2.2 Técnicos .....	70
4.3 Análisis de riesgos.....	72
4.4 Priorización de riesgos .....	74
4.5 Control de riesgos.....	76
4.5.1 Del proyecto.....	76
4.5.2 Técnicos .....	83
4.6 Conclusión.....	90
CAPÍTULO 5 .....	91
DISEÑO DEL SISTEMA.....	91
5.1 Introducción.....	91
5.2 Diseño de la interfaz.....	92
5.3 Diseño estético.....	99
5.4 Diseño de contenido .....	103

5.5 Diseño arquitectónico.....	106
5.6 Arquitectura utilizada para la aplicación.....	109
5.6.1 El modelo MVC.....	110
5.7 Diseño de navegación.....	112
5.8 Conclusión.....	118
CAPÍTULO 6.....	119
CODIFICACION Y PRUEBAS DE LA APLICACION.....	119
6.1 Introducción.....	119
6.2 Herramientas disponibles .....	120
6.2.1 Adobe Flex 3.....	121
6.2.2 ActionScript.....	127
6.2.3 XML (lenguaje de marcas extensible).....	130
6.2.4 PHP (Versión 5.2.6).....	132
6.2.5 MySQL (Structured Query Language) .....	134
6.3 Pruebas en aplicaciones web .....	135
6.3.1 Pruebas de interfaz de usuario .....	135
6.3.2 Pruebas de compatibilidad.....	136
6.3.3 Pruebas de funcionalidad.....	138
6.4 Conclusión.....	140
Conclusiones .....	141
Glosario de Términos.....	142
Referencias.....	146
Anexos.....	148

### **Índice de Ilustraciones**

Fig. 1.1 Pantalla principal MedFile.....	9
Fig. 1.2 Datos demográficos MedFile.....	10
Fig. 1.3 Ficha clínica MedFile .....	11
Fig. 1.4 Visualización de imágenes.....	12
Fig. 1.5 Pantalla principal HiDoctor .....	14

Fig. 1.6 Vista avanzada HiDoctor .....	15
Fig. 1.7 Características adicionales HiDoctor .....	16
Fig. 2.1 Actores del Sistema .....	24
Fig. 2.2 Paradigma XP .....	27
Fig. 2.3 Especificación de Tareas, Tiempo y Recursos .....	30
Fig. 2.4 Diagrama de Gantt .....	31
Fig. 3.1 Caso de Uso Medico Registrado.....	35
Fig. 3.2 Casos de Uso Paciente Registrado.....	36
Fig. 3.3 Modelo de Contenido .....	38
Fig. 3.4 Diagrama de Casos de Uso .....	40
Fig. 3.5 Diagrama de Secuencias para el Registro de Médicos .....	52
Fig. 3.6 Diagrama de Secuencias para el Registro de Pacientes.....	53
Fig. 3.7 Diagrama de Secuencias para la Administración de la Cuenta Medico .....	54
Fig. 3.8 Diagrama de Secuencias para la Administración de la Cuenta Paciente.....	54
Fig. 3.9 Diagrama de Secuencias para la Asignación de Privilegios .....	55
Fig. 3.10 Diagrama de Secuencias para la solicitud de Citas.....	56
Fig. 3.11 Diagrama de Secuencias para la Búsqueda de Pacientes.....	57
Fig. 3.12 Diagrama de Secuencias para Añadir Pacientes .....	58
Fig. 3.13 Diagrama de Estados para la Administración del Perfil Medico.....	59
Fig. 3.14 Diagrama de Estados para la Administración del Perfil Paciente .....	60
Fig. 3.15 Prototipo de Interfaz de Usuario .....	61
Fig. 3.16 Diagrama de Actividades para la Función Buscar.....	62
Fig. 3.17 Diagrama de Despliegue .....	63
Fig. 5.1 Bosquejo de la Interfaz .....	94
Fig. 5.2 Pantalla de Inicio de Sesión .....	96
Fig. 5.3 Interfaz para Médicos .....	97
Fig. 5.4 Interfaz para Pacientes .....	98
Fig. 5.5 Pantalla final inicio de sesión .....	99
Fig. 5.6 Pantalla final registro .....	100
Fig. 5.7 Pantalla final perfil medico.....	101
Fig. 5.8 Pantalla final perfil paciente .....	102
Fig. 5.9 Diseño de contenido formulario de registro .....	103
Fig. 5.10 Diseño de contenido publicidad.....	104
Fig. 5.11 Diseño de contenido administración de citas.....	105
Fig. 5.12 Estructura Jerárquica.....	106
Fig. 5.13 Estructura jerárquica cuenta medico.....	108
Fig. 5.14 Estructura jerárquica cuenta paciente .....	109
Fig. 5.15 Arquitectura multicapa .....	110
Fig. 5.16 Modelo MVC.....	111
Fig. 5.17 Arquitectura web RIA browser.....	112
Fig. 5.18 Semántica navegación publicidad.....	113

Fig. 5.19 Semántica de navegación formulario de registro.....	114
Fig. 5.20 Semántica de navegación ficha clínica .....	115
Fig. 5.21 Semántica de navegación citas .....	116
Fig. 6.1 Adobe Flex Builder 3.....	121
Fig. 6.2 Pantalla nuevo proyecto.....	122
Fig. 6.3 Pantalla configuración proyecto .....	122
Fig. 6.4 Marco de trabajo .....	123
Fig. 6.5 Tab de navegación .....	124
Fig. 6.6 Tab de componentes .....	124
Fig. 6.7 Inserción de objetos .....	125
Fig. 6.8 Código fuente AS.....	125
Fig. 6.9 Ejecución del ejemplo.....	126
Fig. 6.10 Creación de objetos para pantalla entidades .....	126
Fig. 6.11 Pantalla entidades .....	127
Fig. 6.12 Funciones AS para inicio de sesión .....	129
Fig. 6.13 Archivo es.xml.....	131
Fig. 6.14 Archivo en.xml .....	131
Fig. 6.15 Archivo PHP para registro de usuarios.....	133
Fig. 6.16 Esquema de la tabla usuario.....	134
Fig. 6.17 Verificación de mensajes y campos obligatorios.....	136
Fig. 6.18 Aplicación ejecutada en Opera .....	137
Fig. 6.19 Aplicación ejecutada en Chrome .....	137
Fig. 6.20 Comprobación de la función que actualiza antecedentes .....	138
Fig. 6.21 Comprobación de la función que reserva citas .....	139

## Índice de Anexos

Anexo 1 Ficha Clínica .....	148
Anexo 2 Encuesta a médicos .....	150
Anexo 3 Encuesta a pacientes.....	150
Anexo 4 Manual MedFile.....	150
Anexo 5 Modelo Entidad Relación.....	150
Anexo 6 Modelo Relacional .....	150

## **Resumen**

El presente trabajo de graduación se orienta en el estudio, análisis y desarrollo de una aplicación web que se encargara del mantenimiento conjunto de la información médica de una persona a lo largo de su vida, lo cual servirá de beneficio para los usuarios pacientes como para los médicos tratantes, permitiéndoles acceder a la información actualizada que estará disponible en cualquier momento, con el fin de prevenir la posterior perdida u olvido de la misma y evitar así que la historia clínica de un paciente se encuentre dispersa en los registros personales de diferentes médicos o entidades de salud a los cuales el paciente acude.

## **Abstract**

This graduation work focuses on the study, analysis, and development of a web application which will take care of the complete maintenance of a person's medical history throughout his/her life. This will benefit both patients and treating physicians, since it will let them access to updated information which is also permanently available. The purpose is to prevent the loss of information as well as to prevent a patient's medical history from being scattered in the personal files of different physicians or in any health center a patient may go to.

## **Introducción**

Hoy en día el apogeo de las comunicaciones a través de la Internet ha presentado un gran crecimiento; la evolución de la web a una web 2.0 mucho más orientada a redes de colaboración ha permitido que dicha tecnología sea cada día más servicial e indispensable.

Este siglo nos ha traído grandes adelantos tecnológicos como cloud computing o nube de información, que está haciendo que todos los sistemas o aplicaciones migren hacia la web, para que puedan ser accedidas desde cualquier lugar y en cualquier momento.

Como es común en las personas, todos hemos sufrido alguna enfermedad, pero a medida que transcurre el tiempo fácilmente olvidamos lo que padecemos, no recordamos con exactitud cuál fue la prescripción médica, alergias, médicos tratantes, etc.

Consideramos innovador la posibilidad de desarrollar un historial clínico de los pacientes mediante Internet, con la facilidad de que estos y a la vez los médicos tratantes puedan consultar en cualquier momento y en cualquier lugar sus antecedentes clínicos, con lo cual se pretende establecer una universalidad de las fichas clínicas y romper el método tradicional que se lleva hoy en día.

Este proyecto nos brindará la posibilidad de almacenar todos los datos referentes al historial médico de cada persona en un medio como la Internet, mediante la cual dicha información estará al alcance del propietario y sus médicos. De este modo en posteriores consultas podremos recordar y ayudar a los doctores a informarles sobre nuestro antecedente clínico. Y lo más importante es que toda la información se encontrará almacenada en una sola ficha y no en varias fichas como lo es actualmente.

El sistema se desarrollara bajo una plataforma web de acceso gratuito en la cual se procederá a crear cuentas por cada paciente; estas podrán ser creadas por los médicos al momento de la consulta, o por los pacientes en cualquier instancia.

Cada ficha médica contara con los datos personales del paciente, diagnostico, cirugías realizadas, medicamentos prescritos, exámenes de laboratorio, imagenología, enfermedades adquiridas, médicos tratantes, clínicas o lugares en donde se trato o está tratando, todos estos registrando fechas y responsables.

## **CAPÍTULO 1**

### **SITUACION ACTUAL DE LAS FICHAS CLINICAS**

#### **1.1 Introducción**

En el siguiente capítulo trataremos sobre la situación actual de las fichas medicas, sus problemas al momento de su administración, además conoceremos las diversas formas sobre cómo se maneja dichas fichas o reportes médicos, trataremos además sobre conceptos generales de temas como confidencialidad de la información, la seguridad al momento de su manejo, como también la privacidad de los datos que expone cada médico al momento del tratamiento.

Dicha información será obtenida mediante una investigación previa, encuestas que se realizaran a médicos como a pacientes; y mediante el análisis de software existentes en el mercado.

Presentaremos las debilidades que expone hoy en día la administración de las fichas médicas por parte de los médicos como las clínicas, dando a conocer los resultados de las encuestas en las cuales se refleja la problemática actual

## **1.2 Ficha clínica**

### **1.2.1 Definición:**

Una ficha clínica es un documento claro, reservado y confidencial que almacena la información médica de un paciente a lo largo de su vida, constando en ella datos como la anamnesis, examen físico y exámenes complementarios.

Debido a que la información básica que todo médico requiere de un paciente, ha hecho que la ficha clínica sea un documento con formato único en el mundo, y los datos adicionales que requiera un médico dependerán de su especialidad.

### **1.2.2 Partes de una ficha clínica**

De este modo una ficha clínica general se divide en las siguientes partes.

1. Anamnesis
  - Datos de filiación
  - Motivo de consulta
  - Enfermedad actual
  - Revisión de sistemas
  - Historia pasada
  - Historia familiar
  - Historia social
2. Examen físico
  - Examen general
  - Examen regional
  - Examen especial
3. Exámenes complementarios

### **1.2.3 Descripción**

#### **Anamnesis.**

**Datos de Filiación:** Son datos de identificación del paciente, además ayuda a los médicos a recoger información significativa que le ayudará en el diagnóstico.

**Motivo de Consulta:** En esta parte el paciente da a conocer al médico y explica las razones que tiene para acudir a la consulta.

**Enfermedad Actual:** Es una de las partes más importantes de la historia clínica, debido a que en ella se registran las molestias del paciente desde que inicio la enfermedad hasta el mismo instante de la consulta, esto lo tiene que hacer el médico describiendo con criterios técnicos.

**Revisión de Sistemas:** Ayuda a reconocer en el paciente alguna patología adicional que no se detectó en el Motivo de la Consulta ni en la Enfermedad Actual, esto en ocasiones es fundamental ya que ayuda a completar datos de la situación actual del paciente.

**Historia Pasada:** Llamado también antecedentes personales, es donde se registra toda la vida del paciente, tanto sus enfermedades como sus condiciones de vida normal, y comprende desde el nacimiento hasta la fecha actual. Esta puede ser patológica y no patológica. En la **Historia Patológica** se extraen todas las enfermedades del paciente y en la **Historia no Patológica** lo relacionado con los hábitos del mismo que pueden o no ser causa de la enfermedad.

**Historia Familiar y Social:** Recoge información de la historia patológica de la familia y sobre el ambiente socio económico en que se sitúa el paciente.

#### **Examen físico.**

**Examen General:** Aquí se percibe el estado general del enfermo, se examina pulso, tensión arterial, peso, talla, entre otros que son las partes fundamentales de un examen completo.

**Examen Regional:** Se examinan todas las regiones del cuerpo.

**Examen Especial:** Examen de todos los sistemas, en donde se presta más atención al posiblemente afectado.

### **Exámenes complementarios.**

Estos son muy variados e importantes en su debido momento; y entre estos están los exámenes de laboratorio, radiografías, electrocardiogramas, ecografías, biopsias, etc.

Anexo 1 Ficha Clínica

### **1.2.4 Análisis general del manejo de las fichas clínicas.**

Mediante encuestas realizadas a médicos y pacientes pudimos obtener y recopilar la siguiente información.

#### **Resultados de la encuesta a los médicos.**

Anexos Digitales

Anexo 2 Encuesta a **médicos**

Una de las problemáticas que se encuentra hoy en día en el registro de las fichas, es que cada médico cuenta con la información de su paciente y en caso de que dicho paciente tuviese una consulta con un médico diferente, éste le volverá a pedir la información general que todo médico tratante deberá poseer de su paciente, lo cual hace que la información sea redundante.

En la actualidad, en muy pocos lugares el manejo de las fichas es aún rudimentario, debido a que la información es registrada en hojas y éstas son almacenadas en cuartos grandes en donde existen ficheros para el almacenamiento de las carpetas de cada uno de los pacientes, esto ocasiona a que se tengan espacios muy grandes para su almacenamiento y que en algún momento se pueda traspapelar la información, haciendo que los documentos se pierdan sin poder recuperarlos ya que no existe respaldo alguno, siendo así responsables de su pérdida los médicos que manejan las fichas.

Por otra parte, en muchos lugares ya se está utilizando software para el manejo y registro de las fichas, siendo éstos implementados de acuerdo a la necesidad de cada médico en el caso de particulares; y en el caso de instituciones de salud poco de este software es

antiguo y muy poco amigable para el usuario. Este mecanismo de registro de fichas es un adelanto tecnológico que han adquirido los médicos, pero uno de los inconvenientes que esto presenta es que en la mayoría de los casos no se lleva un respaldo de dicha información en caso de daño de equipos, perdiendo por completo todos los registros de sus pacientes.

Con respecto a los exámenes complementarios que son importantes de un paciente, la gran mayoría registra solo los exámenes importantes tanto manualmente como en el software utilizado; en el caso de uso de software, muy pocos registran esa información mediante imágenes y video, pues lo habitual es registrar textualmente los resultados obtenidos.

Otro inconveniente que presenta la administración de fichas mediante software, es que la información almacenada no puede ser vista ni accedida por el médico en otro lugar que no sea su consultorio, pues la información permanece almacenada solo en su equipo, salvo el caso de que el médico cuente con un equipo portátil que le permita llevar dicha información a cualquier lugar.

Uno de los mayores problemas hoy en día y que se encuentra en debate, es sobre a quién pertenece la ficha clínica, si bien la información contenida dentro de esta pertenece al paciente y es confidencial, muchos consideran que ésta debe ser propiedad de los médicos o de las instituciones de la salud, de esta manera los médicos están en la obligación de entregar la información que se solicite, pero solo proporcionan los datos que el paciente pueda interpretar, haciendo que éste tenga acceso a cierta parte de la información. En el caso de que otro médico o institución requiera de la información del paciente, éste debe sacar autorización y los médicos le entregarán una epicrisis de su historial clínico.

En cuanto se refiere a la administración de citas, estas son llevadas en algunos casos por las secretarías, quienes reciben las llamadas y administran los horarios; o simplemente entregan los tickets de turnos de acuerdo al orden en cómo llegan los pacientes, caso contrario el mismo médico es quien administra sus citas mediante teléfono celular o convencional, siendo este el único canal de comunicación que existe entre médicos y pacientes.

También cabe recalcar la forma en cómo los médicos dan a conocer sus servicios, la mayoría de ellos se manejan por recomendaciones que hacen los pacientes y muy pocos publicitan sus servicios mediante radio e internet.

### **Resultados de la encuesta a los pacientes.**

#### Anexo 3 Encuesta a pacientes

Según datos obtenidos mediante una encuesta se llego a determinar las siguientes estadísticas.

De las personas encuestadas se detecto que todas han visitado en alguna ocasión a un médico, por alguna enfermedad para el posterior tratamiento, siendo la frecuencia de visitas a los médicos en caso de enfermedad de una a tres veces por año.

Detectamos además que los pacientes entregan una y otra vez sus datos personales a los médicos los cuales les van a tratar, haciendo de esta tarea repetitiva en las mismas preguntas, e inclusive en preguntas que difícilmente recuerdan la respuesta.

Todas y cada una de las personas, incluyéndonos nosotros difícilmente recordamos nuestro antecedente medico, debido a que olvidamos nuestras enfermedades con el transcurso del tiempo, aun mas los medicamentos que nos fueron diagnosticados para tal tratamiento, haciendo imposible recordar nuestro historial médico completo, por la falta de almacenamiento y organización de la información.

Los médicos en caso de extraviar la ficha médica vuelven a preguntar al paciente sus datos personales además de los medicamentos prescritos, el diagnostico, etc. Información que es de vital importancia para el seguimiento del paciente, que cuando no existen pueden provocar el fracaso del tratamiento por error en las dosis de medicamentos o controles posteriores. Siendo de gran importancia que esta información este respaldada y en conjunto para que el tratamiento sea prospero y correcto.

En caso de que un paciente solicite la información de su tratamiento a un medico este solo le facilita una epicrisis del mismo, lo cual es un resumen del tratamiento en términos que el paciente pueda interpretar, volviendo a este trámite poco amigable debido a las autorizaciones que se tiene que solicitar.

De los datos obtenidos detectamos que los pacientes acuden a varias clínicas para sus tratamientos o consultas, siendo una sola en caso de pertenecer a algún tipo de seguro médico. De tal modo que sus datos han sido entregados en varias clínicas o entidades encargadas de la salud de las personas.

Al momento de solicitar una consulta con algún médico los pacientes se comunican con la secretaria del médico o acuden directamente al consultorio del mismo, poniéndose en lista de espera hasta ser atendidos. Escogen a dicho médico por medio de recomendaciones, el prestigio que el médico se ha ganado a lo largo del tiempo y por la publicidad que el médico tenga en medios locales.

Al preguntar a los pacientes si les gustaría que su ficha médica se única y que ellos sean los propietarios indicaron que si, puesto que con dicha información al alcance de sus propias manos la pudieran mantener organizadamente.

### **1.2.5 Análisis del software**

Mediante una investigación previa hemos considerado dos programas disponibles en internet para analizarlos y encontrar sus fortalezas y debilidades. Este software conseguido se presenta en versiones de prueba, pero concediendo todas las características de una versión pagada del mismo, las limitaciones que presentan son para el manejo de varios médicos o cuentas de usuarios, las cuales para las pruebas no fueron tomadas en cuenta.

Software puesto a consideración.

#### **MedFile 5.04**

MedFile 5.x es un software de fácil uso, efectivo y con un precio accesible, diseñado para satisfacer las necesidades de archivo de Historias (Expedientes) Clínicas y manejo de Turnos (Citas) de un Consultorio o Institución Médica en el que se desempeñen uno o varios profesionales (hasta 200 en la versión multiusuario).

Es un software desarrollado por la empresa Varsur Corporation, disponible en inglés y español, que permite a un médico el mantenimiento de las fichas médicas de sus pacientes por medio de un software el cual le ayudara con dichas tareas.

Dispone de características como el almacenamiento de imágenes de exámenes adicionales realizados, permite llevar un registro del avance del paciente a lo largo de las visitas al especialista.

Dentro de la información de cada paciente tenemos un espacio para registrar eventos como el motivo de la consulta, antecedentes personales del paciente, antecedentes familiares, examen físico, estudios adicionales y prescripciones médicas.

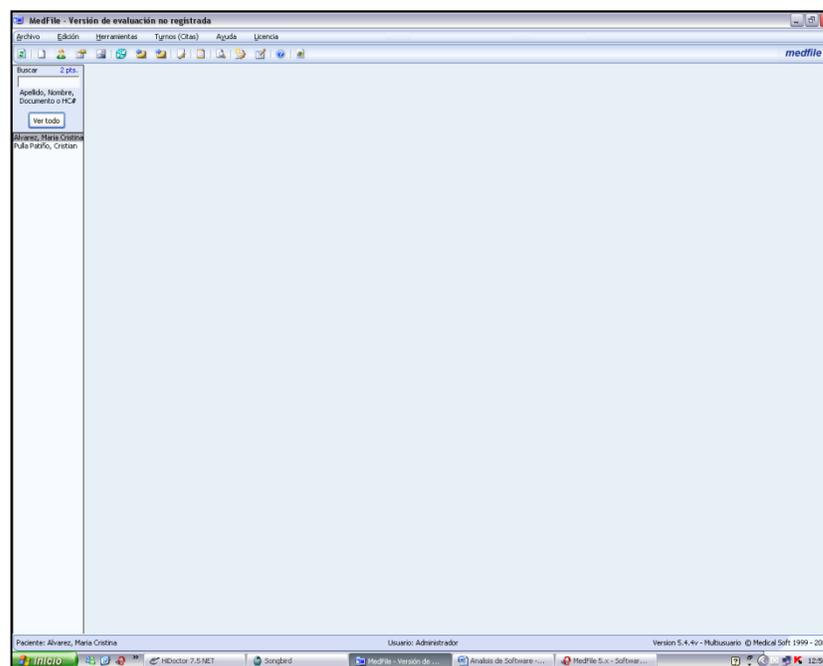
MedFile es un sistema de fácil uso, adaptable a una necesidad básica de un medico, limitando el uso para especialistas que necesiten de campos adicionales para sus registros.

Sus costos oscilan entre 130 y 270 dólares haciendo de MedFile un software relativamente económico y accesible para un medico que necesite disponer de un software que le permita registrar sus fichas medicas electrónicas.

#### **Análisis de medFile 5.04**

Consideramos adecuado el análisis de este software para conocer el estado actual de los sistemas disponibles en el mercado para el registro y manejo de fichas medicas.

**Fig. 1.1 Pantalla principal MedFile**



Pantalla principal del sistema, en donde podemos visualizar en la parte superior la barra de menú, seguida de la barra de herramientas, en la parte izquierda de la pantalla disponemos de un buscador, el cual puede ser ejecutado por varios campos como: Apellido, Nombre, CI, etc. En la parte inferior de la pantalla se presenta la barra de estado, la cual nos brinda información acerca del paciente que estamos consultando, el usuario del sistema que ha iniciado sesión.

El sistema divide a las tareas principales en tres, siendo la primera de estas la del registro de los datos personales de cada paciente

**Fig. 1.2 Datos demográficos MedFile**

The screenshot shows the 'Datos Demográficos' form in the MedFile application. The form is organized into several sections:

- Personal Information:**
  - Apellidos: Alvarez
  - Nombre(s): Maria Cristina
  - H. Clínica: H. Clínica
  - Documento: 9811667
  - Fecha Nac: 03/08/1966
  - Edad actual: 42 años
  - Sexo: F
  - Est. Civil: CAS
  - Ocupación: Ama de casa
  - Nacionalidad: Ecuatoriana
- Address and Contact:**
  - Cobertura: Onepil
  - Nº Afiliado: 47815879-A
  - Domicilio: 7235 SW 48th St.
  - Localidad: Miami
  - Provincia/Estado: Florida
  - Código Postal: 33155
  - Teléfono: 305-666-5599
  - Fax: 305-665-9987
  - Tel. móvil/celular: 305-665-1212
  - Email: mc.alvarez@jllhansen.com
- Medical Information:**
  - Médico emite: Dr. Martinez
  - Médico tratante: Dr. Arriaga G.
- Diagnósticos:**
  - CE-9
  - CE-10
  - Pegar
  - CE-ACM: 862.02 DISTICLOSIS DEL INTESTINO DELGADO CON MENORRAGIA

At the bottom of the form, there are buttons for 'Imprimir Foto', 'Imprimir', 'Guardar', and 'Cancelar'. The status bar at the bottom of the window shows 'Paciente: Alvarez, Maria Cristina', 'Usuario: Administrador', and 'Version: 5.4.4 - Multiusuario - © Medical Soft: 1999 - 2009'.

En esta pantalla podemos observar los datos personales del paciente, o en el caso de registrar uno nuevo se deberán llenar los campos.

Los datos que este software solicita son los siguientes:

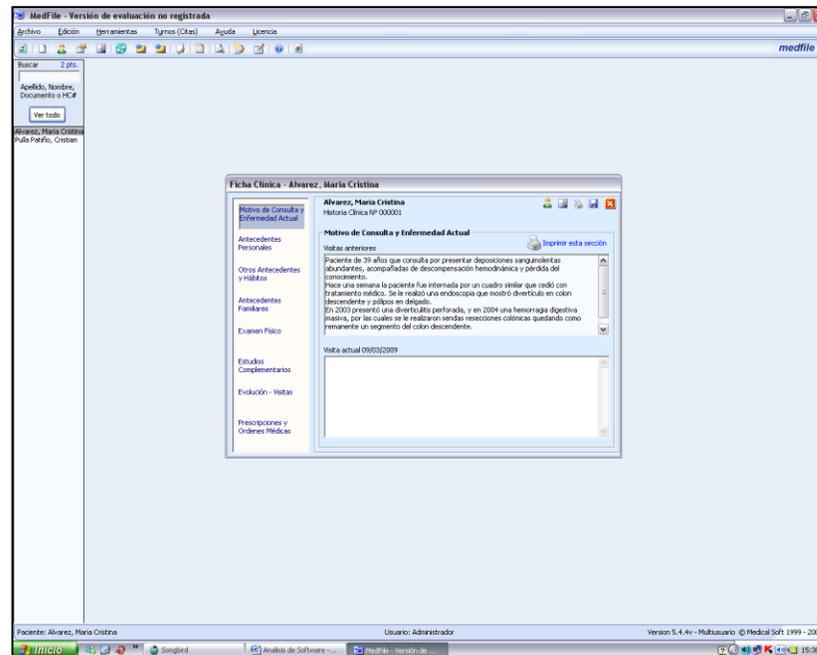
- Nombres
- Apellidos
- Domicilio

- Teléfonos de contacto
- Fotografía
- Un pequeño resumen sobre el estado del paciente.

Algunos de estos datos son registrados en su gran mayoría por los médicos de otras especialidades, dato que obtuvimos a través de la encuesta realizada a los especialistas.

La opción principal que nos presenta este software es el registro de la ficha médica en sí, la cual se encuentra agrupada según varias categorías, detalladas a continuación, con ayuda de un grafico tomado del sistema.

**Fig. 1.3 Ficha clínica MedFile**



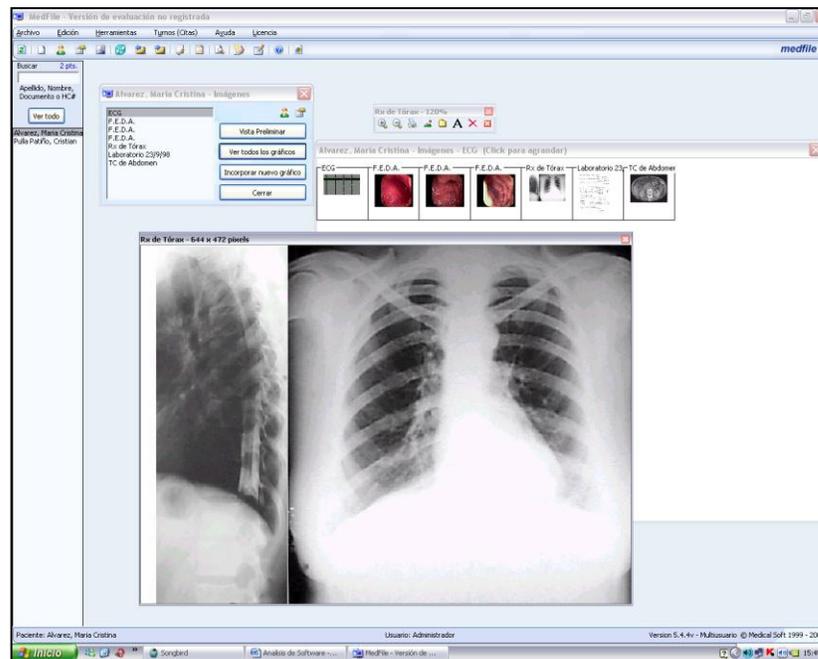
- Motivo de la consulta
- Antecedentes personales
- Otros antecedentes y hábitos
- Antecedentes familiares
- Examen físico
- Estudios complementarios

- Evolución
- Prescripciones

Todas estas opciones son presentadas en dos cuadros de texto plano para que el médico ingrese la información correspondiente a cada tema

La tercera división que nos presenta este software es el registro de imágenes, las cuales pueden ser de exámenes adicionales ordenados por el profesional de la salud.

**Fig. 1.4 Visualización de imágenes**



Como se puede observar la imagen registrada en la paciente, de la cual estamos consultando su ficha médica, corresponde a un examen de rayos X, imagen que está disponible para que el médico lo analice y tome medidas al respecto.

Estas son las tres divisiones principales que presenta el software, adicionalmente presenta varias opciones como la personalización de prescripciones y reservas de citas, opciones que salen del contexto de lo que es una ficha médica por lo cual solo consideramos nombrarlas.

### **Debilidades o problemas**

- El sistema no realiza respaldos automáticos de la información de la base de datos, y en el caso de que este respaldo sea realizado manualmente, este se almacena en el mismo computador, de modo que en caso de falla del computador se pierde dicha información, siendo esta de extrema importancia tanto para el médico como para el paciente.
- El sistema dispone de una versión para varios computadores conectados en red, siendo necesario un conocimiento adicional para poder implementar de este modo la aplicación.
- En caso de que un médico trabaje en más de un consultorio la base de datos es independiente en cada lugar de trabajo, necesitando de una licencia para cada consultorio.
- La aplicación no se contacta de ningún modo con el paciente con el fin de que se auto informe al paciente de citas previstas.
- La información que el médico ingresa no es claramente organizada, debido a que solo se dispone de un cuadro de texto, dificultando la posterior navegación dentro de dicha información.

Durante este análisis hemos encontrado varias ventajas que pueden ser útiles para esta tesis, como también detectamos varias debilidades que podemos aprovecharlas para una mejor implementación de este tipo de sistemas.

Anexo 4 Manual MedFile

### **HiDoctor 7.5**

Otro sistema que consideramos de mucha ayuda analizarlo es uno distribuido por la empresa Centralx, disponible en inglés y portugués, que de igual manera ayuda con el registro de fichas médicas electrónicas. Este sistema dispone de características especiales tales como un mejor manejo de la información almacenada para cada paciente, dispone de un módulo que facilita la facturación y el manejo de citas para el médico.

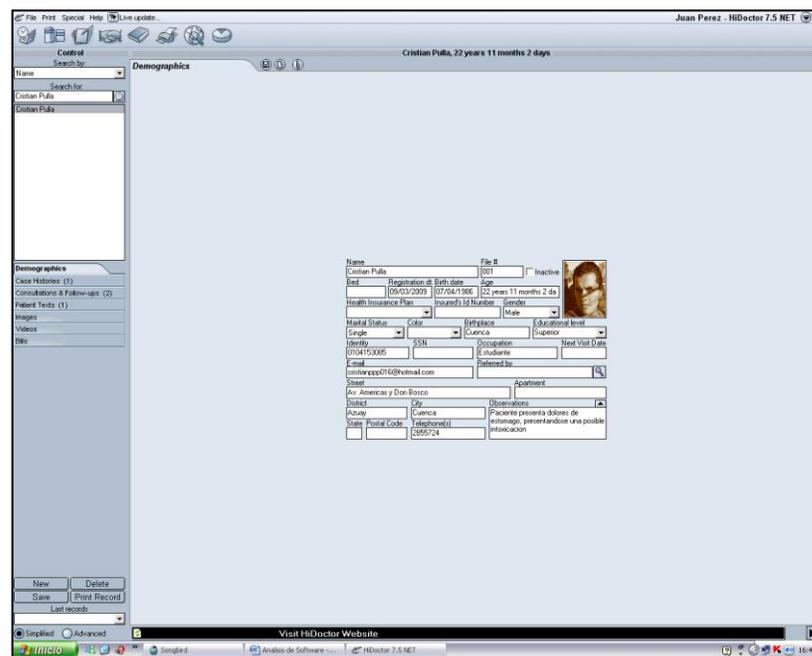
Adicionalmente a estas características, presenta una interfaz de aceptable uso, dicho de este modo puesto que su uso al principio presenta varias dificultades.

Este sistema a diferencia del anteriormente analizado divide a todos los datos del paciente en siete partes, las cuales analizaremos en el transcurso de este análisis.

### Análisis de HiDoctor 7.5

Consideramos adecuado el análisis de esta aplicación para basarnos no solo en un sistema que realiza tareas similares a la propuesta a desarrollar, es decir tener varias fuentes de información y análisis de los sistemas actuales.

**Fig. 1.5 Pantalla principal HiDoctor**



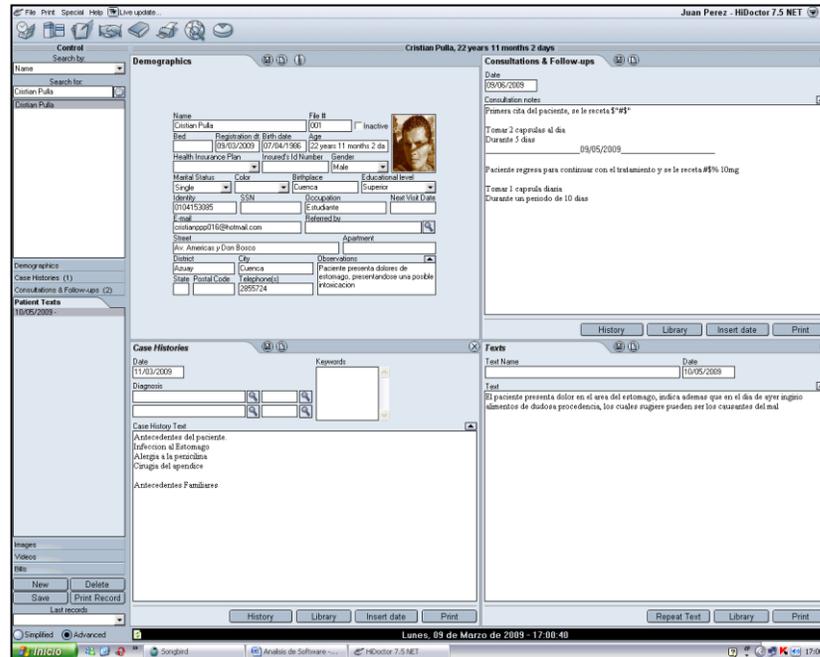
Pantalla principal del sistema, en donde se nos presenta los datos personales del paciente seleccionado, o los casilleros en blanco en caso del registro de un paciente nuevo. La información personal almacenada consta de datos como:

- Nombres
- Apellidos
- Fecha de nacimiento
- Domicilio
- Teléfonos de contacto, correo electrónico
- Fotografía

- Información en caso de recomendación
- Un pequeño resumen sobre el estado del paciente.

Una vez que estos datos están almacenados en la base de datos, se dispone de una visualización avanzada de los datos del paciente, de modo que en la pantalla se nos presentan tres ventanas adicionales para la información.

**Fig. 1.6 Vista avanzada HiDoctor**



Esta ventana dividida en cuatro partes nos presenta en la esquina superior izquierda la información personal del paciente, en la parte de abajo tenemos un cuadro para registrar la información histórica del paciente, en cuanto a enfermedades, antecedentes familiares o demás.

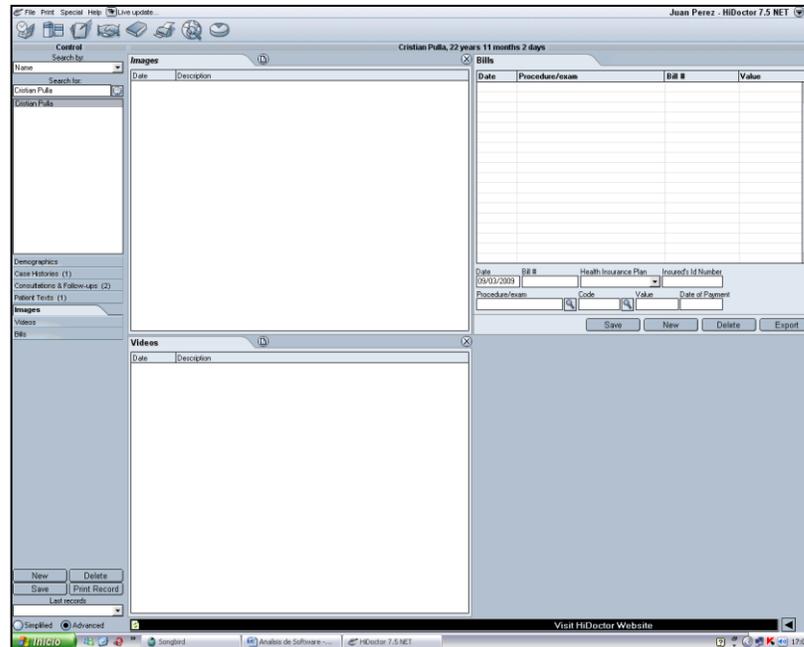
En la parte superior derecha está la información que el médico deberá ir registrando según el paciente acuda a las citas.

El cuarto recuadro de la parte derecha inferior nos permite registrar datos que el paciente brinde al médico tales como, causas del síntoma, situaciones previas o indicios que ayuden al médico a diagnosticar la enfermedad.

Todas estas tres ventanas adicionales a los datos personales se presentan en modo de recuadros de texto, los cuales se pueden ir almacenando según las fechas en las cuales hayan sido redactados.

Adicional a estas ventanas disponemos de las características adicionales nombradas anteriormente

**Fig. 1.7 Características adicionales HiDoctor**



En estas tres restantes ventanas se puede organizar imágenes, videos y adicionalmente un manejo de facturación para la contabilidad del médico.

La aplicación dispone de un mejor manejo de los archivos de cada paciente, organizándolos en la parte izquierda por categorías, las cuales las describimos anteriormente.

Estas siete divisiones son las que el sistema permite al médico administrarlas, siendo de igual manera limitadas en el caso de algún tipo de medicina especial en el cual se requieran de campos adicionales acorde con cada profesional.

### **Debilidades o problemas**

- Dispone de una interfaz poco intuitiva la cual dificulta el manejo del software

- Al igual que el otro sistema analizado no realiza respaldos automáticos de la base de datos, y en caso de hacerlos se almacenan en la misma máquina de trabajo.
- En caso de que un medico trabaje en más de un consultorio la base de datos es independiente en cada lugar de trabajo.
- La aplicación no se contacta de ningún modo con el paciente con el fin de que se auto informe al paciente de citas previstas.
- La información ingresada en cada recuadro se la realiza en modo de texto plano.

Como resultado de este análisis, podemos destacar que ambos sistemas son eficientes para el manejo de fichas médicas, pero también presentan debilidades que las identificamos con el fin de aprovecharlas y mejorarlas, así también en el caso de las fortalezas poder guiarnos en ellas y adaptarlas a nuestro sistema.

### **1.3 Conclusión**

En el presente capítulo pudimos obtener información de vital importancia para conocer y aprender sobre las fichas clínicas, la información básica que esta contiene, el motivo del porque es utilizado dicho documento. También pudimos conocer las diferentes maneras en las que se las administra en la actualidad, siendo el manejo de estas aun ineficiente; detectamos las fortalezas y debilidades que los sistemas tradicionales presentan y los requerimientos que los profesionales de la salud exigen.

## **CAPÍTULO 2**

### **FORMULACION Y PLANEACION DEL SISTEMA**

#### **2.1 Introducción**

En el siguiente capítulo se identificarán los requisitos funcionales y de contenido con los que contará el sistema, detectaremos a los actores principales involucrados en el uso del mismo, las actividades que cada uno de ellos va a realizar con el fin de obtener una visión generalizada de lo que el sistema realizará, además se dará a conocer el ciclo de vida a utilizarse y se definirá las tareas y el tiempo de cada una de ellas con lo cual se designará sus recursos y los tiempos aproximados que tomara su desarrollo.

## **2.2 Identificación y análisis de requisitos**

### **2.2.1 Requisitos de contenido**

#### **Formulario de acceso:**

El usuario ya registrado (medico o paciente) ingresara datos que le permitan ingresar al sistema, entre estos serán el nombre de usuario o e-mail y contraseña.

El sistema validara si los datos de acceso pertenecen a un paciente o un medico, con el fin de exhibir la información correspondiente a cada categoría de usuario.

#### **Acceso al formulario de registro**

La pantalla principal de ingreso al sistema también dispondrá de vínculos que permitan a los nuevos usuarios (médicos, pacientes) registrarse en el sistema.

#### **Cabecera de página**

Dispondrá de un enlace, el cual permitirá al usuario escoger el idioma con el cual desea navegar en la aplicación. Además, se mostrará un enlace para Cerrar la Sesión en el caso de que se encuentre en el perfil de usuario.

#### **Pie de página**

Contara con enlaces informativos acerca de la empresa como: políticas de privacidad, derechos de autor, contacto, soporte, etc.

#### **Información del servicio**

Se mostrara información básica sobre las principales características que el sistema ofrece, destacando nuevas versiones o actividades especiales.

## **Formulario de registro**

Los nuevos usuarios deberán ingresar datos personales que permitan crear la cuenta, se podrá indicar la categoría de usuario (medico, paciente).

Se indicara la contraseña, la cual será verificada según su dificultad.

Se mostrara el contrato de términos de uso del sistema, los cuales los usuarios deberán aceptar para poder ser miembro del mismo.

## **Búsquedas**

El sistema dispondrá de una búsqueda, la cual estará disponible según el usuario.

La búsqueda que podrá realizar un medico podrá ser de: pacientes, medicinas, entidades de salud, etc.

La búsqueda que estará disponible para un paciente será relacionada con médicos, servicios clínicos, farmacias, medicinas, etc. Pudiendo reservar una cita en el caso de que se consulte un médico especialista.

Las búsquedas se las realizaran mediante criterios avanzados que permitan al usuario filtrar la información.

## **Administración de citas**

La administración de citas será de exclusivo uso del médico registrado.

El médico podrá manejar sus lugares y horarios de atención, el estado de la consulta y duración de la misma, notificación de citas posteriores, etc.

## **Manejo de la ficha clínica**

El manejo de la ficha médica estará a cargo del paciente propietario o del médico tratante autorizado por el paciente.

Por motivos de privacidad de la información el paciente deberá administrar permisos a los médicos que él desea accedan a su información, pero en el caso de no asignar estos, el médico accederá simplemente a la información de filiación.

La información que se registre no podrá ser modificada ni eliminada, tan solo se la podrá ocultar por parte del paciente propietario de la cuenta.

Esta ficha clínica constara con todos los datos registrados desde la fecha de creación hasta la actual, además ésta incluirá información acerca de todos los tratamientos, diagnósticos, exámenes adicionales, recetas, etc. que el médico haya proporcionado al mismo durante un tratamiento.

El paciente, también podrá registrar información correspondiente a enfermedades comunes que padezca en cierta fecha, como por ejemplo gripe, tos, etc., y tratamientos que tenga, cuyo medico tratante no disponga del sistema para escribir dicha información.

La información que sea ingresada a la cuenta del paciente, registrara información adicional como la fecha, además del responsable de la información.

### **Manejo de la cuenta del medico**

El médico podrá administrar su cuenta, en la cual podrá visualizar a todos sus pacientes.

El médico además podrá manipular los datos personales de su cuenta, la información correspondiente a sus títulos profesionales, sobre sus subespecialidades y la información de contacto.

También el médico podrá ingresar sus propios registros concernientes a medicinas, enfermedades y exámenes en el caso de que no exista en la base de datos y necesite de alguno de éstos para tratar a un paciente.

#### **2.2.2 Requisitos funcionales**

- Administrar fichas medicas

- Registro de usuarios
- Administración de citas
- Manejo de imágenes
- Búsquedas personalizadas
- Seguridad
- Administración de recordatorios

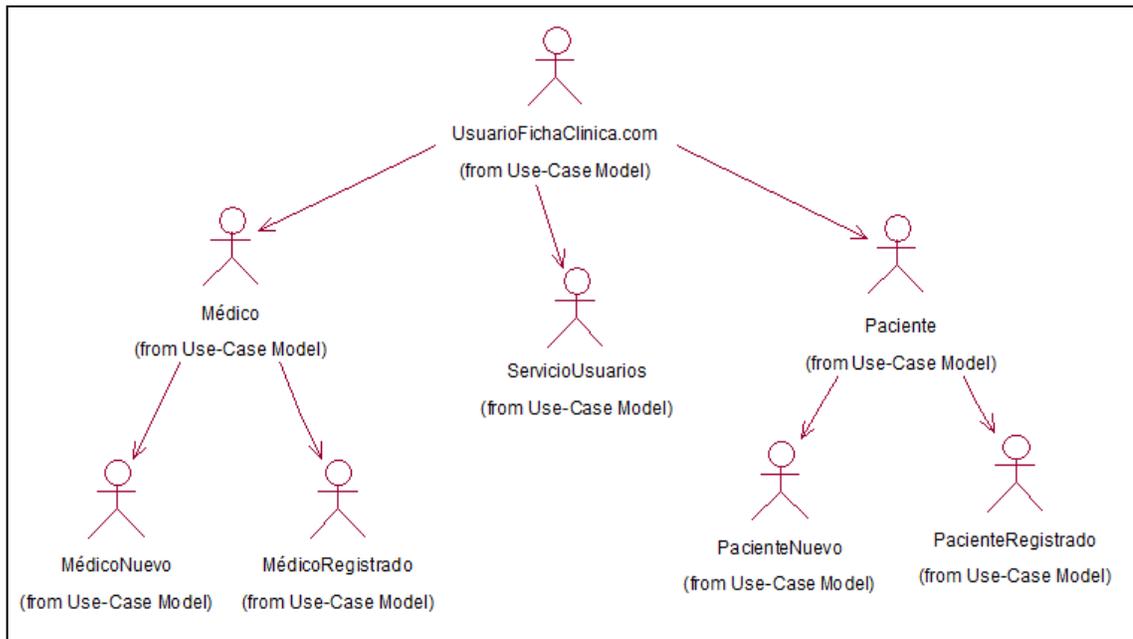
### **2.2.3 Requisitos no funcionales**

- Velocidad de conexión
- Control de Concurrencia
- Carga impredecible debido a la variación de usuarios de un día a otro.
- Desempeño de la WebApp
- Facilidad de uso
- Compatibilidad con navegadores
- Tráfico en la red

### **2.2.4 Categorías de usuario**

El siguiente diagrama muestra los actores directos que harán uso del sistema.

**Fig. 2.1 Actores del Sistema**



### **Medico nuevo**

Podrá registrarse y crear su cuenta en el sistema.

Deberá llenar el formulario de registro con preguntas que validen su identidad y aceptar los términos de uso del sistema.

El médico acorde a su especialidad podrá agregar subespecialidades en el caso de tenerlas.

El médico deberá ingresar una contraseña de acceso, la que le permitirá acceder a su cuenta personal para el manejo de la misma.

### **Medico registrado**

Administración de la cuenta personal

El médico dispondrá de acceso a los datos de sus pacientes.

Si un paciente le ha otorgado privilegios, el médico podrá escribir información acerca de tratamientos, recetas, etc.

El médico podrá subir al sistema imágenes que ayuden con el diagnóstico y tratamiento del paciente.

En el caso de que el paciente no exista, el médico deberá crear una nueva cuenta de usuario paciente.

También podrá consultar los pacientes que se encuentren en la base de datos personal.

El médico podrá administrar su agenda de citas.

Podrá visualizar publicidad de clínicas, medicinas, etc.

### **Paciente nuevo**

Podrá ingresar sus datos personales para el registro en el sistema, aceptando los términos de uso del sistema.

El paciente deberá ingresar una contraseña de acceso, la que le permitirá acceder a su cuenta personal para el manejo de la misma.

El paciente podrá o no ingresar la contraseña corta, la cual le servirá para darle permisos al médico tratante a que escriba información en su ficha. En caso de no registrar dicha contraseña, ésta será un extracto de 5 caracteres de la contraseña de acceso a su cuenta.

### **Paciente registrado**

Manipulación de su cuenta personal y cambio de contraseña en caso de requerirlo.

Su uso estará limitado con el acceso a su propia cuenta y no a la de terceros.

Asignará privilegios a médicos que él considere adecuado para la visualización de su información personal.

Con excepción de sus datos personales, el paciente no podrá modificar ni eliminar los datos escritos anteriormente por médicos o por el mismo.

En caso de que el paciente considere necesario, simplemente podrá ocultar su información, a otros médicos que no sea al que escribió.

Podrá manejar y administrar imágenes en su cuenta.

En caso de que necesite, podrá reservar una cita con especialistas registrados en el sistema.

Dispondrá de un buscador de médicos, clínicas, farmacias, etc.

Además visualizará publicidad de médicos, establecimientos de salud, farmacias, etc.

### **Servicio a usuarios**

Estarán encargados del mantenimiento de la página y del ingreso de nuevos datos a la base de datos.

## **2.3 Ciclo de vida del proyecto**

### **2.3.1 El proceso de programación XP**

Es un paradigma que se basa en la simplicidad comunicación y retroalimentación de código, además propone un desarrollo secuencial del software, el cual va progresando a través del análisis, diseño, codificación y pruebas.

**Fig. 2.2 Paradigma XP**



Esta metodología está compuesta por las siguientes actividades:

**Planificación:** se definen los requerimientos del usuario además de identificar los diferentes requisitos con los que el sistema contara, también se realiza una lista ordenada de cada una de las tareas en donde se estimaran sus tiempos y se designaran sus recursos, de igual manera se determinan los posibles riesgos a presentarse de modo que se los pueda prevenir por medio de un plan de mitigación de riesgos.

**Diseño:** se realizara un diseño sencillo del sistema con el fin de cumplir las necesidades más inmediatas del cliente, se utilizan metáforas las cuales brindan una visión evolutiva del sistema con el paso del tiempo, la misma que define su alcance y propósito.

Adicionalmente se utilizan tarjetas CRC (Clase, Responsabilidad y Colaboración) las cuales ayudan a definir tareas durante el diseño del proyecto, sus responsabilidades y colaboraciones con otras tareas.

**Codificación:** una vez completadas las etapas anteriores y haber definido con precisión las tareas a realizar, se empieza con la codificación del mismo, que consiste en programar pequeños fragmentos de código y escribir ciertas pruebas que estos fragmentos deberán pasar antes de continuar codificando las siguientes tareas.

Esto se lo hace con el fin de que al término de la codificación se obtenga un software con un porcentaje mínimo de fallas, debido a que cada parte perteneciente al mismo fue probada con anterioridad.

Las tareas realizadas y probadas se las debe integrar con las tareas ya finalizadas anteriormente, con el fin de ir acoplando las partes del sistema hasta llegar a la integración total.

**Pruebas:** todas las tareas realizadas en el sistema deben pasar determinadas pruebas individuales acerca del funcionamiento de cada una de ellas, una vez que el sistema está integrado en su totalidad se revisa una vez más esta etapa, la cual en esta fase del sistema consisten en pruebas de integración.

**Características:** una de las características principales de este modelo es que nos permite definir el objetivo del sistema, lo cual hace que se tenga presente lo que se pretende alcanzar. Además nos permitirá evitar retrasos en la entrega de los proyectos ya que se realiza una planeación adecuada mediante la elaboración de diagramas de actividades que permitirán el cumplimiento de las mismas.

### **Ventajas**

- Los programadores novatos aprenderán de los expertos al emparejarse con ellos.
- Evita la incompatibilidad del código al integrarlo en cada versión.
- Se introduce la planeación y administración.
- La fase de codificación se pospone hasta comprender bien el objetivo del proyecto y hacer una respectiva planificación.
- Se realiza primero el diseño de todo el sistema, garantizando así que los productos cumplan con todo lo que el usuario requiera.

- Se maneja una lista de todas las fases preestablecidas con el fin de que si en algún momento se detecta un problema se pueda regresar a la fase anterior, resolver el problema y continuar con el desarrollo.
- Permite la reutilización de código

### **Desventajas**

- Es difícil poder establecer detalladamente al principio todos los requisitos.
- La mejora o cambios en el estilo de programación puede resultar más complejo
- No se permite reutilizar los datos que ya se tenían
- No existen modelos establecidos para las pruebas, por lo que se generan contratiempos cuando un proyecto llega a crecer.

### **Aplicaciones**

Proyectos en los que los requisitos tienen altas probabilidades de cambiar con el tiempo.

Proyectos con alto riesgo. Proyectos con un grupo pequeño de programadores (entre 2 y 12)

#### **2.4 Definición de tareas**

Con el objeto de llevar una planificación ordenada y cronológica del proyecto, se definirán tareas, tiempos y recursos, con el fin de obtener una lista de actividades y el tiempo necesario para la elaboración y cumplimiento del mismo.

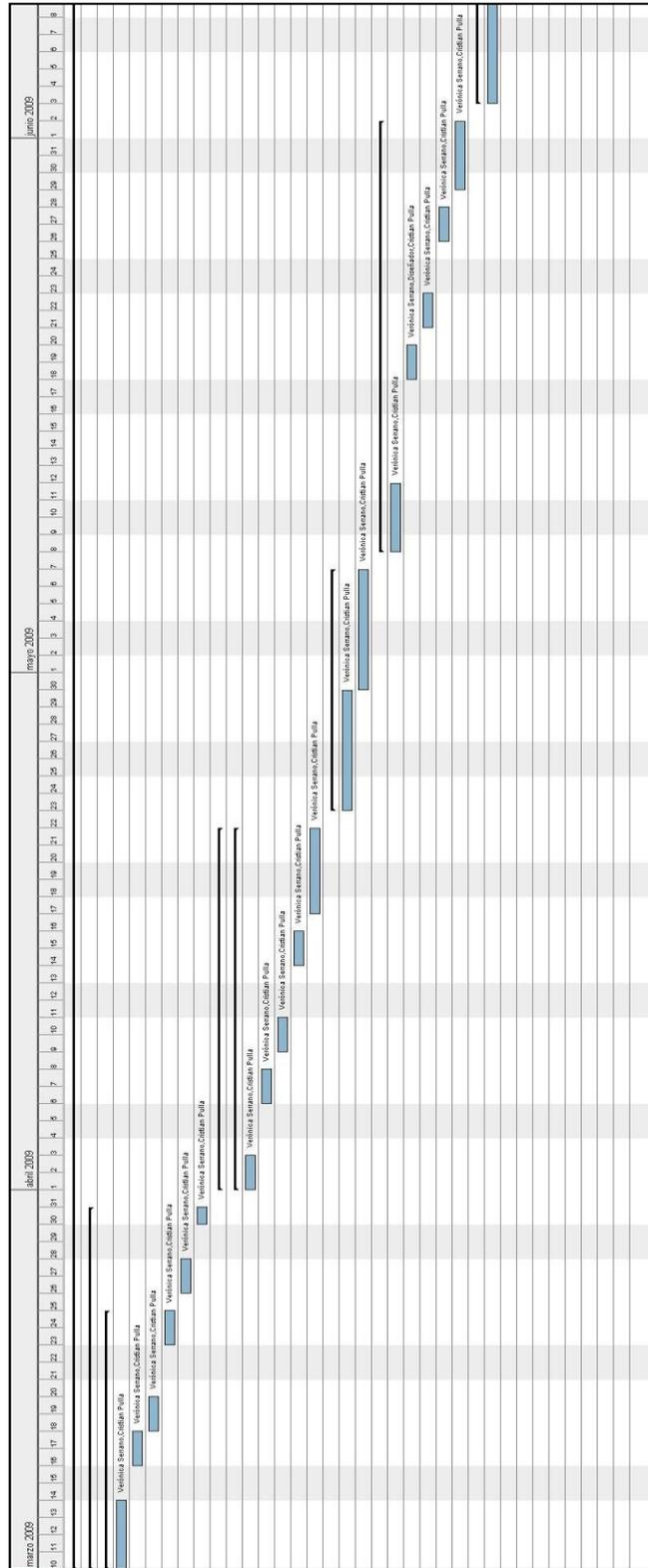
El orden de las tareas es de vital importancia, puesto que existen algunas que son dependientes directas de otras, de igual manera el tiempo calculado para cada tarea deberá basarse en experiencias anteriores; agregando un tiempo adicional por cada una, para prevenir que un retraso inesperado en una tarea afecte al tiempo total estimado para el proyecto.

## 2.4.1 Definición de tiempos y recursos para cada tarea

Fig. 2.3 Especificación de Tareas, Tiempo y Recursos

Nº Tarea	Nombre	Descripción	Fecha de inicio	Fecha de fin	Duración	Coordinador
0	WebApp		10/03/09	12/03/10	263	
1	Formulación y Planeación del Proyecto	Fase de estimación general del proyecto	10/03/09	31/03/09	15	
1.1	Análisis de Requerimientos		10/03/09	25/03/09	11	
1.1.1	Requisitos de Contenido	Identificación de las áreas imprescindibles para el funcionamiento de la aplicación	10/03/09	14/03/09	4	Verónica Serrano, Cristian Pulla
1.1.1.1	Requisitos Funcionales	Definición de los requerimientos directos del sistema	16/03/09	18/03/09	2	Verónica Serrano, Cristian Pulla
1.1.1.3	Requisitos No Funcionales	Definición de los requerimientos externos al sistema	18/03/09	20/03/09	2	Verónica Serrano, Cristian Pulla
1.1.1.4	Definición de Usuarios	Definición de los actores divididos por categorías de usuario que interactuarán con el sistema	23/03/09	25/03/09	2	Verónica Serrano, Cristian Pulla
1.2	Definición del Ciclo de Vida a Utilizar	Identificación del paradigma que se utilizará para el desarrollo del proyecto	28/03/09	28/03/09	2	Verónica Serrano, Cristian Pulla
1.3	Especificación de Tareas	Se definirá cada una de las tareas con sus tiempos y recursos	30/03/09	31/03/09	1	Verónica Serrano, Cristian Pulla
2	Análisis del Proyecto Web	Análisis teórico previo del sistema	1/04/09	22/04/09	15	
2.1	Modelo de Análisis		1/04/09	22/04/09	15	
2.1.1	Modelo de Contenido	Determinación de los contenidos que estarán disponibles en el sitio web	1/04/09	3/04/09	2	Verónica Serrano, Cristian Pulla
2.1.2	Modelo de Interacción	Identificar cómo los usuarios interactuarán con la web	6/04/09	8/04/09	2	Verónica Serrano, Cristian Pulla
2.1.3	Modelo Funcional	Descripción de los procesos con los que contará el sistema	9/04/09	11/04/09	2	Verónica Serrano, Cristian Pulla
2.1.4	Modelo de Configuración	Definición de herramientas de hardware y software que permitan una correcta navegación del usu...	14/04/09	16/04/09	2	Verónica Serrano, Cristian Pulla
2.1.5	Análisis de Navegabilidad	Establecer políticas que faciliten la navegación del usuario en el sistema	17/04/09	22/04/09	3	Verónica Serrano, Cristian Pulla
3	Gestión de Riesgos	Identificación y manejo de posibles riesgos	23/04/09	7/05/09	10	
3.1	Identificación de Riesgos	Análisis y definición de los posibles riesgos que puedan presentarse en el proyecto	23/04/09	30/04/09	5	Verónica Serrano, Cristian Pulla
3.2	Mitigación de Riesgos	Maneras de prevenir cada uno de los riesgos en caso de presentarse	30/04/09	7/05/09	5	Verónica Serrano, Cristian Pulla
4	Diseño del Proyecto Web	Construcción de la parte gráfica del sistema	8/05/09	2/06/09	17	
4.1	Diseño de la Interfaz	Determinar las características con las que contará la interfaz de la web	8/05/09	12/05/09	2	Verónica Serrano, Cristian Pulla
4.2	Diseño Estético	Diseñar estéticamente la interfaz de modo que sea atractiva para el usuario	18/05/09	20/05/09	2	Verónica Serrano, Cristian Pulla
4.3	Diseño de Contenido	Enlazar los atributos de la página con el diseño estético	21/05/09	23/05/09	2	Verónica Serrano, Cristian Pulla
4.4	Diseño Arquitectónico	Definición de la estructura de navegación a ser utilizada en la web	26/05/09	28/05/09	2	Verónica Serrano, Cristian Pulla
4.5	Diseño de Navegación	Creación de las rutas de navegación con las que contará la página	29/05/09	2/06/09	2	Verónica Serrano, Cristian Pulla
5	Codificación y Pruebas	Desarrollo del código y pruebas del sistema	3/06/09	13/03/10	405	
5.1	Herramientas disponibles	Investigación de las herramientas a ser utilizadas para el desarrollo de la aplicación	3/06/09	10/08/09	203	Verónica Serrano, Cristian Pulla
5.2	Desarrollo de la Aplicación para la Administración del sitio	Generación del código para la parte administrativa del sistema	11/08/09	21/10/09	67	Verónica Serrano, Cristian Pulla
5.3	Desarrollo de la Aplicación para los Usuarios	Conversión de todos los modelos en una aplicación de software lista para ser usada	5/10/09	13/03/10	135	Verónica Serrano, Cristian Pulla
5.4	Pruebas en Aplicaciones Web	Realización de las pruebas de la aplicación	11/08/09	13/03/10	202	Verónica Serrano, Cristian Pulla

Fig. 2.4 Diagrama de Gantt



## **2.5 Conclusión**

Durante la elaboración de este capítulo se ha identificado los diferentes tipos de requisitos con los que contará el sistema, detectando de esta manera los funcionales, no funcionales y de contenido. Además se definió las categorías de usuarios, los cuales interactuarán con el sistema. Se estableció el ciclo de vida de programación extrema que será utilizado como base para la elaboración del proyecto; esto lo vimos conveniente, debido a que en este caso los clientes son los mismos desarrolladores. Además esta metodología es muy flexible, puesto que nos permite regresar a fases anteriores del proceso en caso de ser necesario; también nos permitirá la reutilización de código y una buena planificación mediante la elaboración de cronogramas, lo cual hará que el proyecto sea más eficiente. Finalmente se definió cada una de las tareas con sus respectivos tiempos y responsables de modo que se pudo obtener un cronograma de las actividades que el proyecto abarca.

## **CAPÍTULO 3**

### **ANÁLISIS DEL SISTEMA**

#### **3.1 Introducción**

En el siguiente capítulo se empezara a reflejar los requisitos del sistema a través de diferentes modelos, los cuales nos ayudaran de una forma grafica a entender de mejor manera que es lo que se quiere desarrollar; mediante estos modelos detectaremos la forma en cómo los usuarios interactúan con el sistema, así como la funcionalidad con la que contara el mismo, la estructura interna de cada una de sus partes, etc. Una vez que se hayan analizado los diversos factores con los que el sistema deberá cumplir, se podrá describir detalladamente cada una de las funciones que los actores podrán realizar dentro del sistema de acuerdo a los casos de uso.

## **3.2 Requisitos para el análisis**

### **Formulación**

Luego de un análisis exhaustivo sobre el manejo actual de las fichas medicas, las cuales son llevadas de una manera desordenada, no pertenecen a los pacientes, y la información se encuentra distribuida en los diferentes establecimientos en donde el paciente fue atendido por los diferentes médicos a lo largo de su vida. Nace la idea de crear un sistema, el cual reúna toda la información o historial médico de un paciente a lo largo de su vida, con el fin de que esta ficha sea única, se pueda compartir cierta información con médicos que lo soliciten y con la posibilidad de consultar dicho contenido en cualquier lugar y en cualquier momento. La forma en la que esto se puede llevar a cabo es realizando la aplicación bajo un sistema web, la cual permita reunir toda la información en una base de datos general.

### **Categorías de usuarios**

Se identificaron los siguientes actores que interactuaran con el sistema:

- Medico Nuevo
- Medico Registrado
- Paciente Nuevo
- Paciente Registrado
- Servicio a Usuarios

Véase Cap2 Categorías de Usuarios

### **Recopilación de requisitos**

Mediante un análisis exhaustivo sobre el problema que trata el registro de fichas medicas usando un software, se llegaron a determinar ciertos requisitos los cuales son necesarios para la adaptación de la nueva propuesta, además estos requisitos han sido analizados basándonos en sistemas que realizan tareas similares disponibles comercialmente en el mercado.

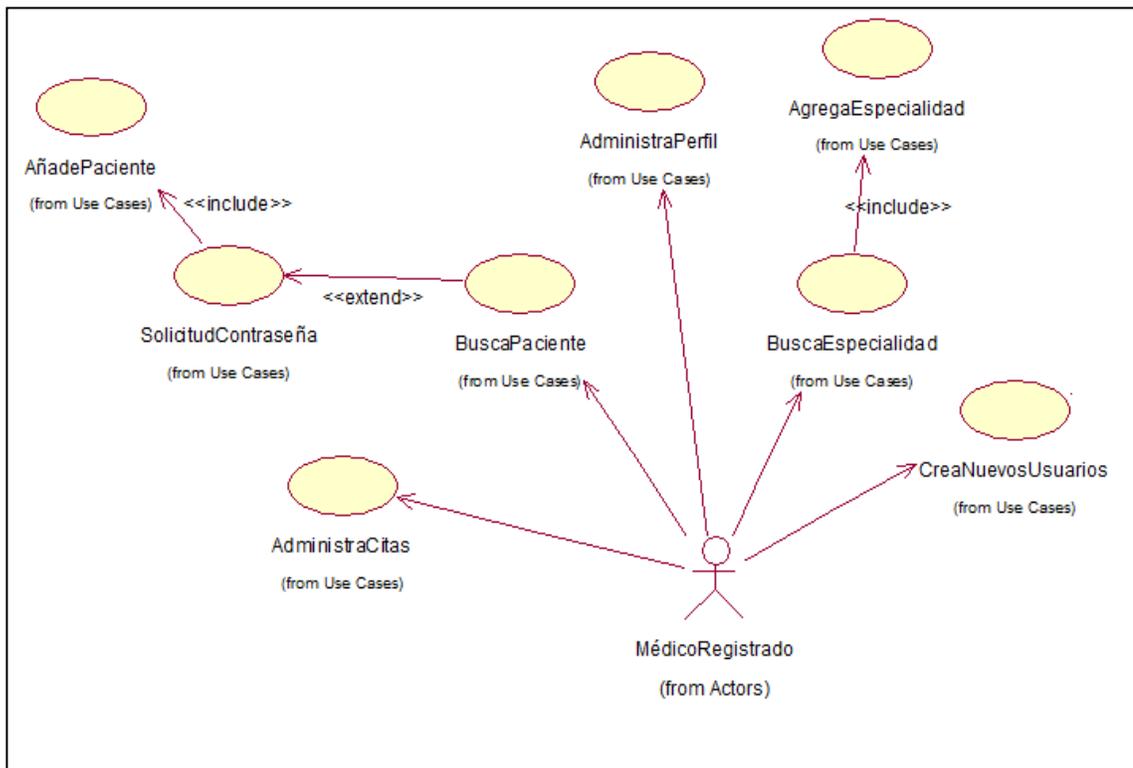
Estos requisitos fueron anteriormente identificados y analizados en el capítulo anterior.

Véase Cap2 Requisitos de Contenido

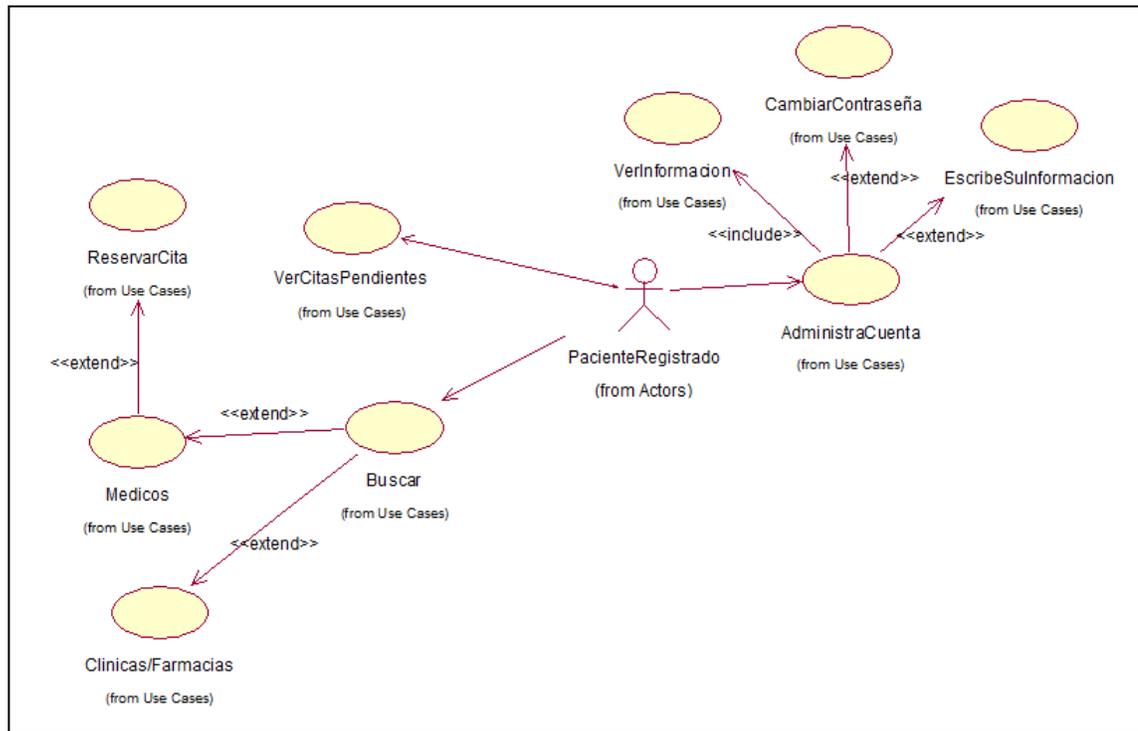
Véase Cap2 Requisitos Funcionales

### Escenarios de interacción

**Fig. 3.1 Caso de Uso Medico Registrado**



**Fig. 3.2 Casos de Uso Paciente Registrado**



### 3.3 Modelo de análisis

A continuación se construirán los diferentes modelos los cuales contendrán elementos estructurales y dinámicos con los que contara el sistema.

### 3.4 Modelo de contenido

En el siguiente modelo trataremos de explicar gráficamente el contenido de la WebApp, organizado de manera estructural acorde a los requisitos del sistema para que de esta manera se identifiquen claramente las estructuras base, como las dependientes las que conformaran el sistema.

Los objetos con los que el sistema contara pueden estar definidos individualmente, o se los puede jerarquizar de modo que se contemplen las relaciones entre los mismo según su nivel en la jerarquía.

El modelo de contenido está conformado por objetos de contenido y de clases de análisis

- **Objetos de contenido**

Son los artículos de información que serán presentados a los usuarios finales que interactúen con el sistema, los cuales se accederán por medio de referencias en la navegación por el sistema.

- **Clases de Análisis**

Son cada una de las entidades que pueden ser vistas por los usuarios.



### **3.5 Modelo de interacción**

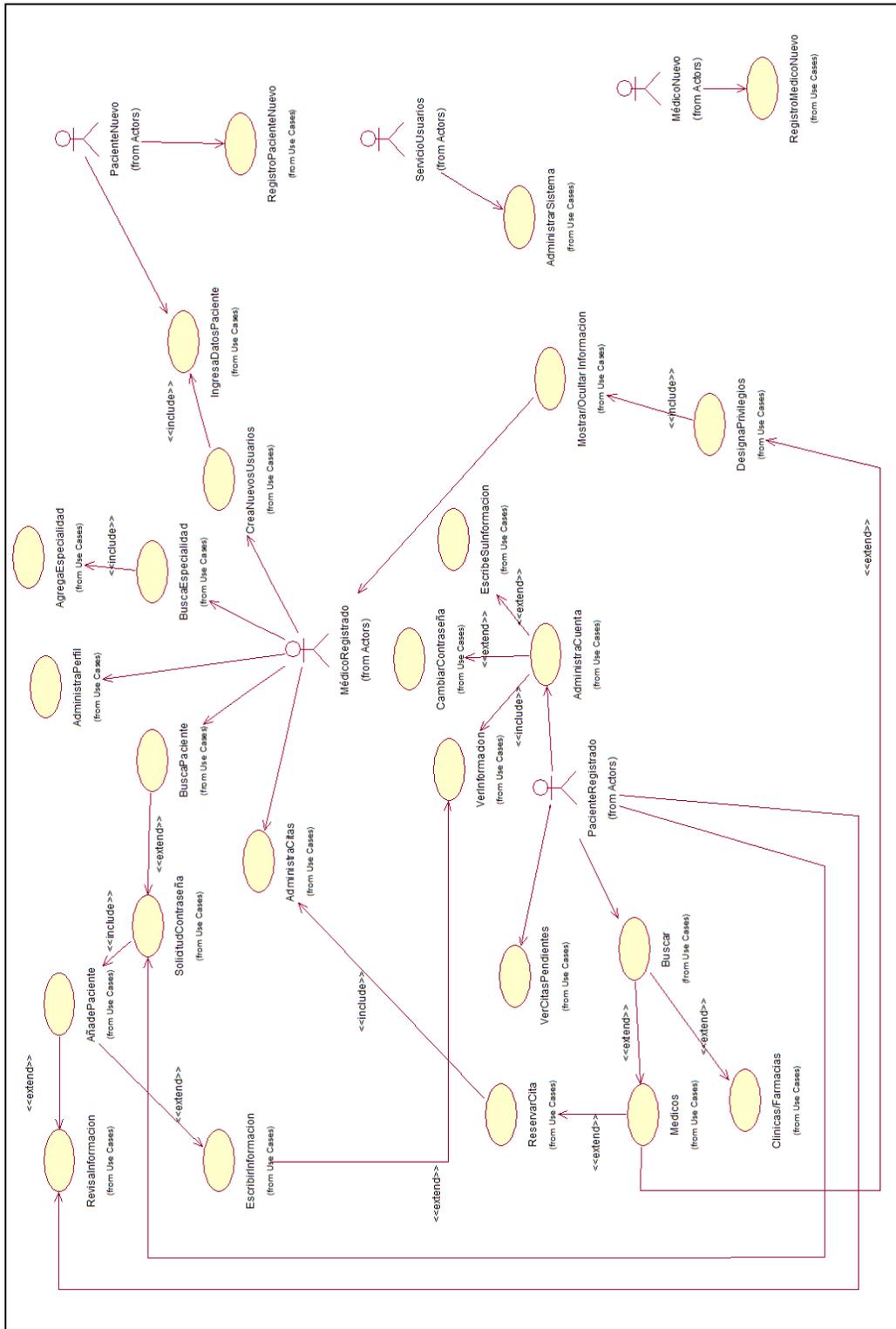
Estos modelos permiten visualizar la forma en cómo los usuarios interactúan con las diferentes partes del sistema.

Los siguientes elementos nos permitirán representar las diversas interacciones entre el usuario y el sistema.

- **Casos de Uso**

Este diagrama muestra los diferentes escenarios con los que cuenta el sistema, muestra como el sistema responderá ante los eventos generados por los actores principales del sistema.

**Fig. 3.4 Diagrama de Casos de Uso**



- **Descripción de Casos de Uso**

<b>Caso de uso 1.</b>	<b>Administra Perfil</b>
<b>Actor:</b>	Medico Registrado
<b>Descripción:</b>	El médico podrá manipular su cuenta.
<b>Prioridad:</b>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El médico podrá visualizar todos los datos de su cuenta y modificarlos como él considere necesario.</li> <li>• El médico podrá realizar un mantenimiento de sus títulos profesionales.</li> <li>• Podrá además administrar las contraseñas y el manejo de privilegios que él ha otorgado.</li> <li>• El médico registrado podrá visualizar a sus pacientes personales.</li> </ul>	

<b>Caso de uso 2.</b>	<b>Búsqueda de pacientes</b>
<b>Actor:</b>	Medico Registrado
<b>Descripción:</b>	Sección en la cual se facilitara al medico la búsqueda de un paciente por medio de criterios.
<b>Prioridad:</b>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El médico solicitara al paciente datos personales los cuales le ayudaran en la búsqueda de la ficha del paciente.</li> <li>• El sistema buscara con prioridad en la base de datos del médico, en caso de no encontrar al paciente, se buscara en la base de datos general.</li> <li>• En caso de que el paciente se encuentre en la base de datos del médico, se mostrara la información de la ficha de dicho paciente</li> <li>• En caso de que el paciente no se encuentre en la base de datos del médico, y se encuentre en la base de datos general, el sistema solicitara una contraseña corta para la agregación de la ficha a la base de datos personal.</li> <li>• Si el paciente no disponga de una cuenta, el médico deberá registrarlo.</li> </ul>	

<b>Caso de uso 3.</b>	<b>Solicitud de Contraseña corta</b>
<b>Actor:</b>	Medico Registrado, Paciente Registrado
<b>Descripción:</b>	El médico solicitara la contraseña corta al paciente para poder registrar al paciente en su base de datos personal y poder acceder a la ficha del mismo.
<b>Prioridad:</b>	Necesario
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El médico solicitara la contraseña corta al paciente</li> <li>• El paciente entregara la contraseña corta, o de acceso general.</li> <li>• El sistema verificara dicha contraseña.</li> </ul>	

<b>Caso de uso 4.</b>	<b>Añadir Paciente</b>
<b>Actor:</b>	Medico Registrado
<b>Descripción:</b>	El sistema otorgara privilegios al médico, para que este pueda acceder a la ficha clínica del paciente.
<b>Prioridad:</b>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• Una vez que la contraseña haya sido verificada, el paciente constara en la base de datos personal del médico, teniendo este acceso a la misma.</li> <li>• En caso de que el paciente haya asignado privilegios previos al médico, este tan solo accederá a la información permitida.</li> <li>• En caso de que el paciente no haya asignado privilegios al médico, este tan solo podrá visualizar la información de filiación de dicho paciente en consulta.</li> <li>• Este paciente estará disponible en la base de datos personal para posteriores consultas.</li> </ul>	

<b>Caso de uso 5.</b>	<b>Revisa Información</b>
<b>Actor:</b>	Medico Registrado
<b>Descripción:</b>	El médico visualizara información del paciente.

<b>Prioridad:</b>	Opcional
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El médico visualizara la información que el paciente le haya otorgado por medio de privilegios.</li> <li>• En caso de que el paciente no haya asignado privilegios, el médico podrá solamente ver la información básica o de filiación.</li> <li>• El médico podrá ver la información completa de lo que él ha escrito en anteriores consultas.</li> </ul>	

<b>Caso de uso 6.</b>	<b>Escribir Información</b>
<b>Actor:</b>	Medico Registrado
<b>Descripción:</b>	El médico digitara información concerniente a la consulta actual.
<b>Prioridad:</b>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El médico colocara información concerniente al diagnostico y tratamiento del paciente.</li> <li>• Podrá subir imágenes, video que ayuden con el tratamiento al médico.</li> <li>• Podrá registrar los resultados de laboratorio de exámenes adicionales solicitados.</li> <li>• El médico en base a su diagnostico realizara una prescripción médica.</li> <li>• Se registrara los datos del examen físico correspondientes a la fecha actual.</li> <li>• El especialista no podrá modificar ni eliminar información que haya sido ingresada, por el u otro médico, en fechas anteriores a la actual.</li> <li>• El sistema registrara automáticamente la fecha de la consulta actual.</li> <li>• El médico registrara la fecha de siguiente consulta programada, con el fin de que sea un recordatorio para el paciente cuando acceda a su cuenta.</li> </ul>	

<b>Caso de uso 7.</b>	<b>Crear Nuevas Cuentas de Usuario</b>
<b>Actor:</b>	Medico Registrado
<b>Descripción:</b>	En caso de que el paciente no disponga de una cuenta en el

	sistema, el médico usuario del sistema ayudara a crearla.
<b>Prioridad:</b>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• Si la búsqueda del paciente resulto negativa, el médico ingresara los datos del paciente al sistema.</li> <li>• El médico solicitara datos personales al paciente, dichos datos son requeridos por el sistema.</li> <li>• Una vez que el paciente sea registrado en el sistema general, pasará a formar parte de la base de datos personal del médico tratante.</li> </ul>	

<b>Caso de uso 8.</b>	<b>Ingresar Datos del Paciente</b>
<b>Actor:</b>	Medico Registrado
<b>Descripción:</b>	El médico solicitara datos al paciente para su registro.
<b>Prioridad:</b>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El médico ingresara los datos básicos de filiación.</li> <li>• El paciente podrá saber su contraseña de acceso a su perfil ya que será el mismo quien le otorgue dicho dato al médico.</li> </ul>	

<b>Caso de uso 9.</b>	<b>Administrar Citas</b>
<b>Actor:</b>	Medico Registrado
<b>Descripción:</b>	El médico dispondrá de una pantalla de administración de citas
<b>Prioridad:</b>	Opcional
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El médico podrá llevar un registro de las fichas reservadas por los pacientes.</li> <li>• Podrá visualizar las citas planeadas, reservadas, en consulta o atendidas.</li> <li>• El sistema mostrara al médico las citas que tenga en una determinada fecha.</li> </ul>	

<b>Caso de uso 10.</b>	<b>Busca Especialidad</b>
<i>Actor:</i>	Medico Registrado
<i>Descripción:</i>	El médico podrá añadir la especialidad a la que el pertenezca.
<i>Prioridad:</i>	Opcional
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>El médico podrá buscar subespecialidades de acuerdo a la especialidad a la que el pertenezca.</li> </ul>	

<b>Caso de uso 11.</b>	<b>Agrega Especialidad</b>
<i>Actor:</i>	Medico Registrado
<i>Descripción:</i>	El médico agregara subespecialidades.
<i>Prioridad:</i>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>El médico agregara más subespecialidades en caso de tenerlas, acorde a la especialidad.</li> </ul>	

<b>Caso de uso 12.</b>	<b>Administra Cuenta</b>
<i>Actor:</i>	Paciente Registrado
<i>Descripción:</i>	El paciente podrá manipular los datos concernientes a información personal, privilegios, etc.
<i>Prioridad:</i>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>El paciente como propietario de la ficha clínica podrá administrar y manipular su cuenta como él considere necesario.</li> <li>El acceso a esta parte del sistema se la realizara por parte de la contraseña de administrador de cuenta.</li> </ul>	

<b>Caso de uso 13.</b>	<b>Ver Información</b>
<i>Actor:</i>	Paciente Registrado

<b>Descripción:</b>	El paciente podrá visualizar toda la información escrita en su cuenta de usuario.
<b>Prioridad:</b>	Opcional
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• Esta información corresponderá a datos proporcionados por sus médicos tratantes y la información escrita personalmente.</li> <li>• La información visualizada podrá ser en texto, imágenes y video, según como se la haya ingresado.</li> </ul>	

<b>Caso de uso 14.</b>	<b>Escribir Información</b>
<b>Actor:</b>	Paciente Registrado
<b>Descripción:</b>	El paciente podrá ingresar datos.
<b>Prioridad:</b>	Opcional
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El paciente podrá auto-registrar información en su cuenta, esta información será registrada en el caso de que el médico no lo haga.</li> <li>• El paciente si desea podrá ingresar datos sobre su historia pasada y antecedentes personales.</li> <li>• Esta información tendrá carácter de no verificada, puesto que no es escrita por un medico autorizado.</li> <li>• El paciente no podrá eliminar ni modificar datos ingresados con anterioridad por los especialistas.</li> <li>• Los únicos datos actualizables serán relacionados con información personal del paciente (Domicilio, Telf., etc.).</li> </ul>	

<b>Caso de uso 15.</b>	<b>Cambiar Contraseña</b>
<b>Actor:</b>	Paciente Registrado
<b>Descripción:</b>	El paciente podrá actualizar su contraseña de acceso al sistema.
<b>Prioridad:</b>	Opcional

<b>REQUISITOS ASOCIADOS</b>	
	<ul style="list-style-type: none"> <li>• El paciente está en la obligación de cambiar su contraseña una vez que el médico le ha registrado en el sistema, por motivos de privacidad.</li> <li>• La contraseña que el paciente registre deberá ser mayor a ocho caracteres (Números, Letras, Caracteres Especiales)</li> <li>• El motivo de que la contraseña sea larga, ya que esta servirá para el acceso del paciente, mientras que para que el médico pueda acceder a una cuenta de un paciente este deberá solicitar una contraseña corta, que será una parte de la contraseña original.</li> <li>• El sistema indicara al usuario la dificultad de la contraseña ingresada</li> </ul>

<b>Caso de uso 16.</b>	<b>Asignar Privilegios</b>
<b>Actor:</b>	Paciente Registrado
<b>Descripción:</b>	El paciente como propietario de su cuenta podrá otorgar privilegios para el acceso a su información.
<b>Prioridad:</b>	Opcional
<b>REQUISITOS ASOCIADOS</b>	
	<ul style="list-style-type: none"> <li>• El paciente podrá permitir a ciertos médicos el acceso a su ficha clínica.</li> <li>• Podrá administrar los permisos otorgados a médicos tratantes.</li> <li>• En caso de que un nuevo medico no tenga privilegios asignados, este simplemente visualizara los datos de filiación, historia familiar, etc. del paciente.</li> </ul>

<b>Caso de uso 17.</b>	<b>Mostrar/Ocultar Información</b>
<b>Actor:</b>	Paciente Registrado
<b>Descripción:</b>	Una vez que el paciente ha otorgado el acceso a su ficha, este puede filtrar la información que el médico visualizara.
<b>Prioridad:</b>	Opcional
<b>REQUISITOS ASOCIADOS</b>	

- El paciente podrá asignar o remover permisos de visualización acerca de información que él cree conveniente mostrarla.
- Estos permisos se verán reflejados simplemente en los médicos tratantes del paciente.
- La información escrita por cierto profesional no podrá ser ocultada al mismo, teniendo este acceso a la información completa que él ha escrito.
- En caso de que el paciente no haya modificado los permisos, los médicos solamente visualizarán la información básica además de la que ellos hayan escrito en la cuenta del paciente.

<b>Caso de uso 18.</b>	<b>Ver Citas Pendientes</b>
<i>Actor:</i>	Paciente Registrado
<i>Descripción:</i>	El paciente podrá visualizar las citas pendientes.
<i>Prioridad:</i>	Opcional
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El paciente podrá revisar las citas que tenga pendientes con los diferentes médicos con quienes haya sacado alguna cita.</li> <li>• Las citas que serán visualizadas serán a partir de la fecha actual en adelante.</li> <li>• En cada cita pendiente, se mostrará el día, la hora, el lugar y el médico con quien sacó la cita</li> </ul>	

<b>Caso de uso 19.</b>	<b>Buscar</b>
<i>Actor:</i>	Paciente Registrado
<i>Descripción:</i>	El paciente mediante una búsqueda podrá consultar farmacias, clínicas, etc. que se encuentren registrados en el sistema.
<i>Prioridad:</i>	Opcional
<b>REQUISITOS ASOCIADOS</b>	

- El paciente podrá visualizar información acerca de clínicas, servicios médicos, laboratorios, etc. por medio de una búsqueda avanzada.
- La búsqueda se basara en criterios como: lugar de ubicación, horarios de atención, etc.

<b>Caso de uso 20.</b>	<b>Buscar Médicos</b>
<b>Actor:</b>	Paciente Registrado
<b>Descripción:</b>	El paciente podrá visualizar información de médicos.
<b>Prioridad:</b>	Opcional
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El paciente realizara búsquedas de acuerdo a especialistas que el requiera.</li> <li>• Estos especialistas visualizados, serán simplemente los que hayan solicitado servicios de publicidad.</li> <li>• En esta búsqueda se mostrara la ubicación de su consultorios, horarios de atención, información que facilitara al paciente la ubicación del medico</li> <li>• El motivo de esta consulta es que una vez encontrado el especialista requerido, el paciente podrá reservar una cita con el mismo.</li> </ul>	

<b>Caso de uso 21.</b>	<b>Reservar Cita</b>
<b>Actor:</b>	Paciente Registrado
<b>Descripción:</b>	El paciente podrá reservar citas con médicos especialistas registrados en el sistema.
<b>Prioridad:</b>	Opcional
<b>REQUISITOS ASOCIADOS</b>	

- El paciente una vez realizada la búsqueda podrá reservar una cita en cierto día para la consulta con el médico seleccionado.
- Al reservar la cita el paciente otorgara privilegios al médico, con el fin de que este pueda ver los datos básicos de filiación.
- En caso de que el paciente necesite asignar privilegios adicionales al médico, lo podrá hacer.

<b>Caso de uso 22.</b>	<b>Administra Sistema</b>
<b>Actor:</b>	Servicio al Usuario
<b>Descripción:</b>	Este departamento se encargara del manejo y mantenimiento del sistema.
<b>Prioridad:</b>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El servicio al usuario maneja las actualizaciones que se vayan dando al sistema.</li> <li>• Además se encargara de administrar las cuentas, y detectar un mal manejo que se pueda estar dando.</li> <li>• Este departamento será el encargado del manejo general del mantenimiento del sistema.</li> </ul>	

<b>Caso de uso 23.</b>	<b>Registro Medico Nuevo</b>
<b>Actor:</b>	Medico Nuevo
<b>Descripción:</b>	Se solicitaran los datos necesarios para que el nuevo medico se registre en el sistema.
<b>Prioridad:</b>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	

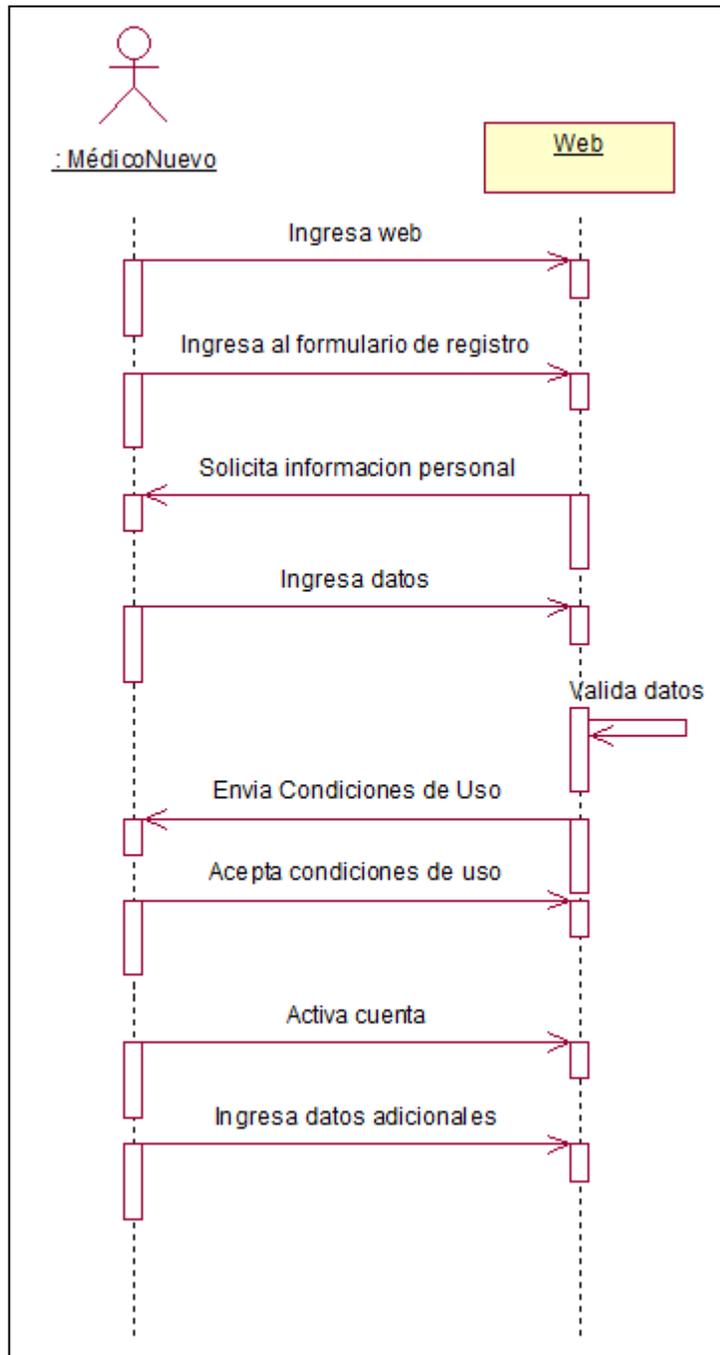
- Un formulario de datos básicos serán ingresados por parte del médico.
- Se solicitara la información correspondiente a lugar de trabajo, horarios de atención, etc.
- El médico al momento de su registro podrá asignar privilegios para que su asistente ingrese a su cuenta y administre las citas.
- El médico deberá aceptar las condiciones de uso del sistema.

<b>Caso de uso 24.</b>	<b>Registro Paciente Nuevo</b>
<b>Actor:</b>	Paciente Nuevo
<b>Descripción:</b>	Se solicitaran los datos necesarios para que el nuevo paciente se registre en el sistema.
<b>Prioridad:</b>	Obligatorio
<b>REQUISITOS ASOCIADOS</b>	
<ul style="list-style-type: none"> <li>• El sistema solicita al paciente los datos personales para su registro.</li> <li>• El paciente ingresara datos como la contraseña que le permitirá el posterior ingreso al sistema.</li> <li>• El paciente deberá aceptar las condiciones de uso del sistema.</li> <li>• Posteriormente el paciente podrá manipular su cuenta, otorgando y revocando privilegios, etc.</li> </ul>	

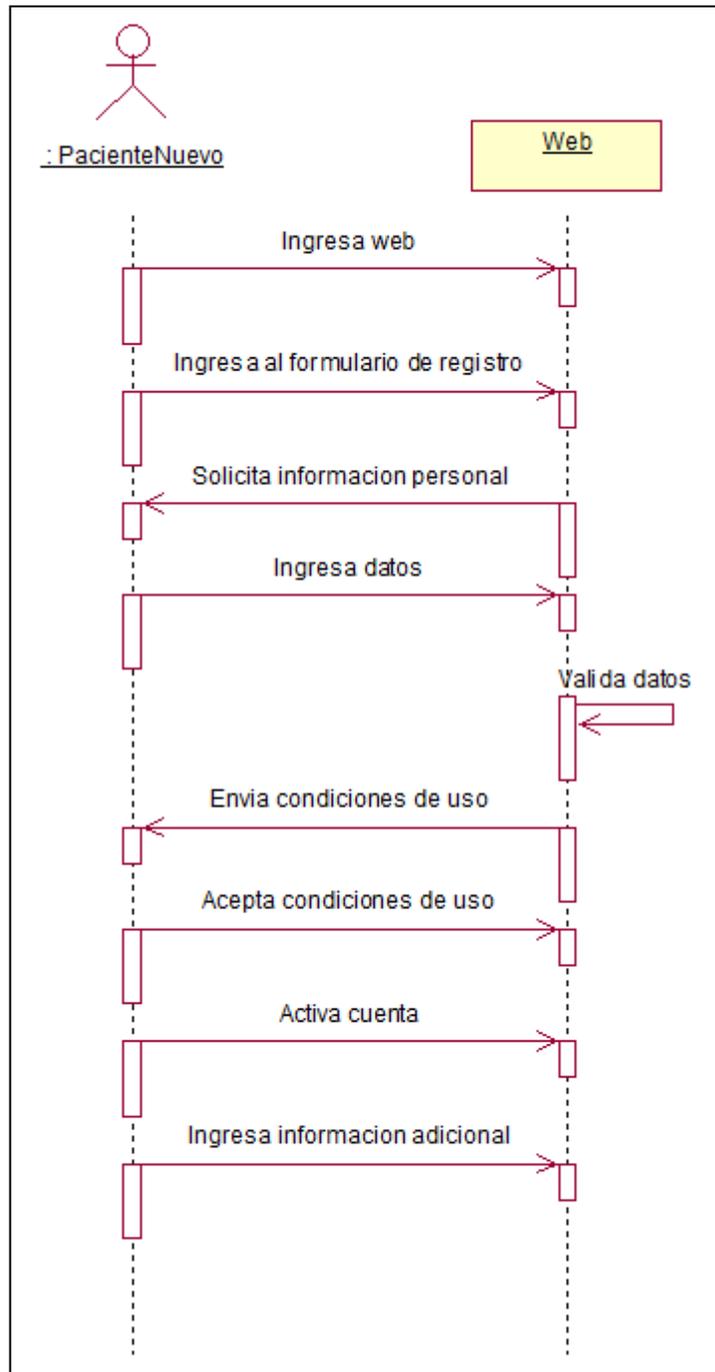
- **Diagramas de Secuencia**

Este modelo representa gráficamente y de manera cronológica el intercambio de mensajes entre un usuario y un evento para el desarrollo del mismo; además este modelo muestra el escenario óptimo de los sucesos a realizarse sin contemplar excepciones.

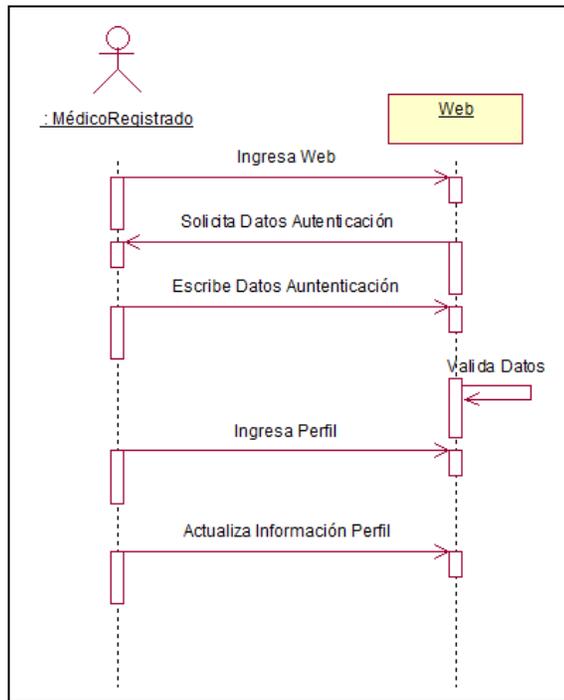
**Fig. 3.5 Diagrama de Secuencias para el Registro de Médicos**



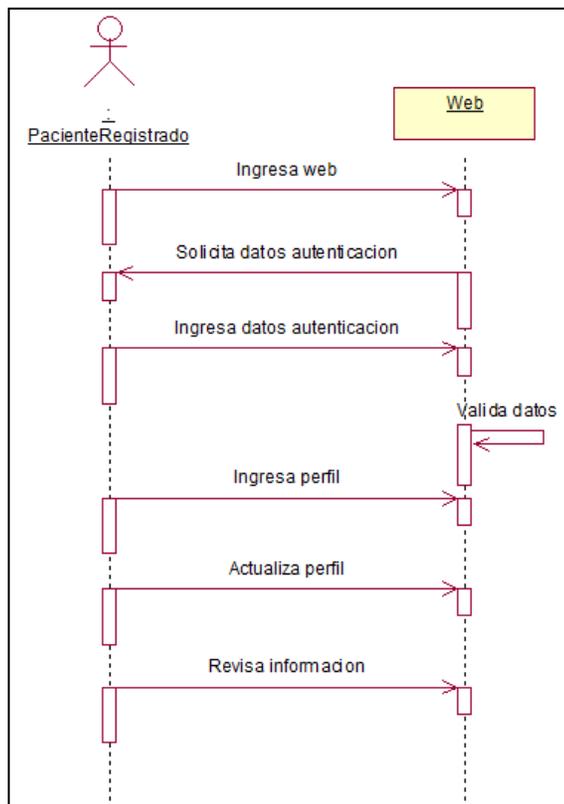
**Fig. 3.6 Diagrama de Secuencias para el Registro de Pacientes**



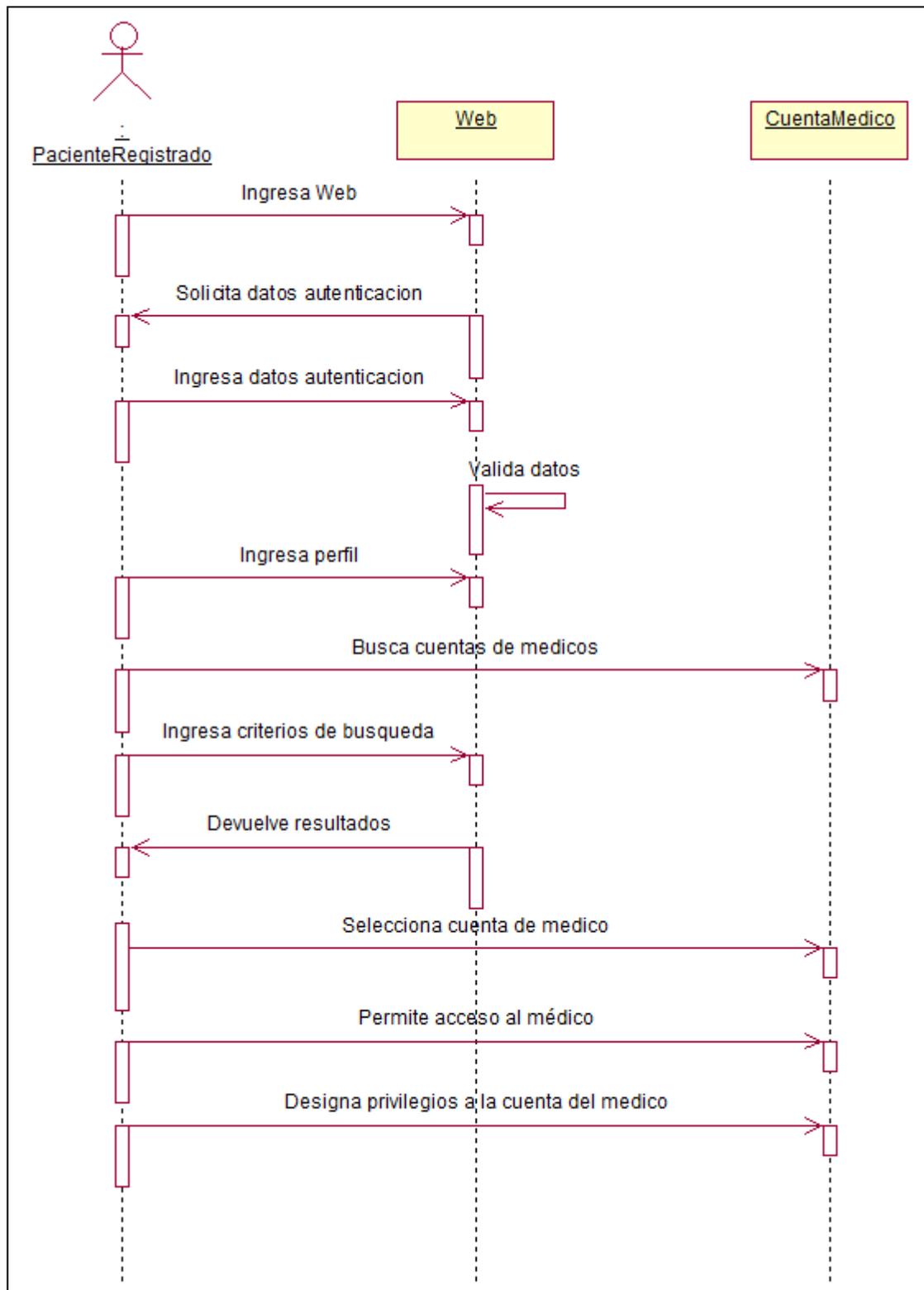
**Fig. 3.7 Diagrama de Secuencias para la Administración de la Cuenta Medico**



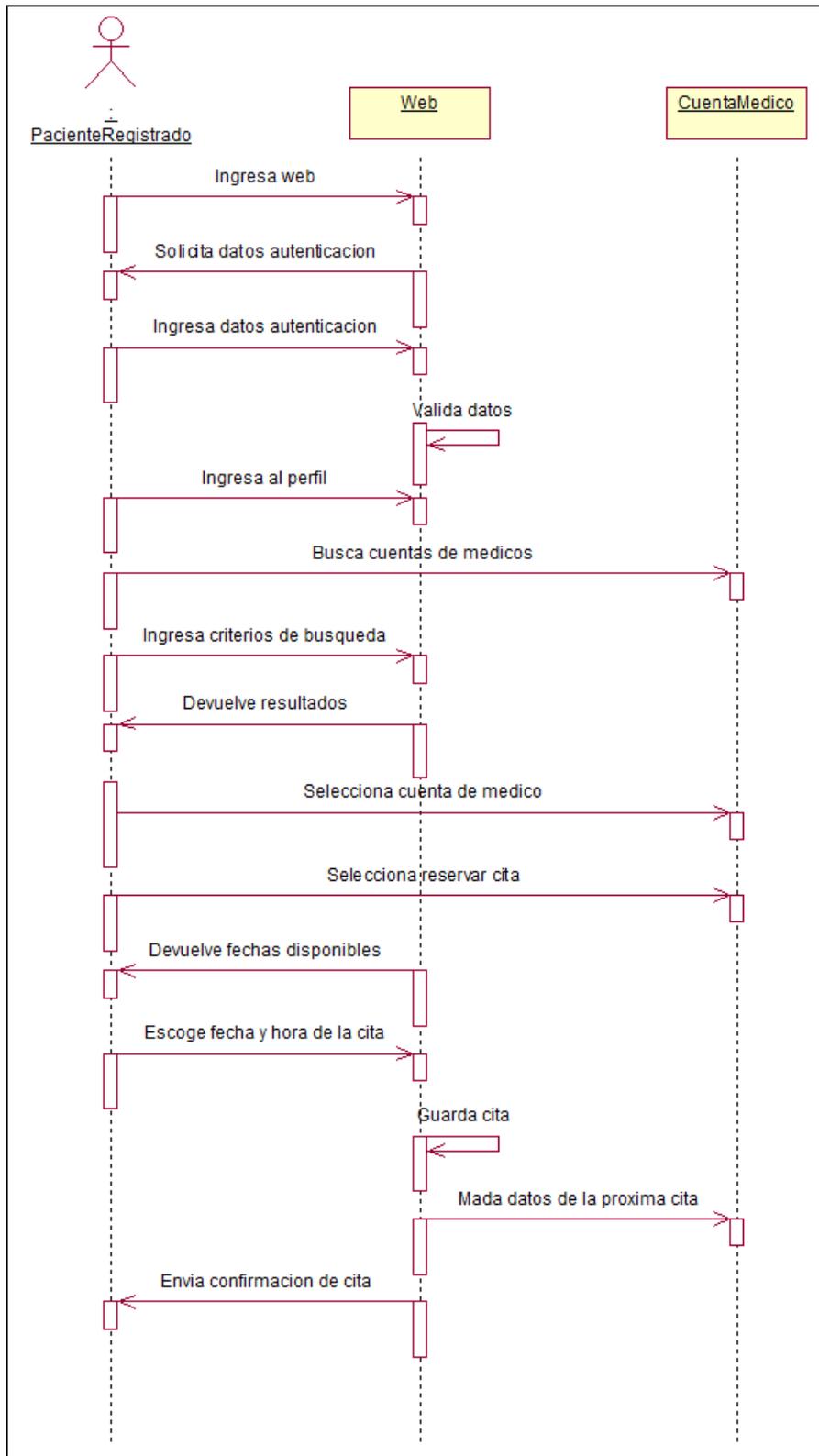
**Fig. 3.8 Diagrama de Secuencias para la Administración de la Cuenta Paciente**



**Fig. 3.9 Diagrama de Secuencias para la Asignación de Privilegios**



**Fig. 3.10 Diagrama de Secuencias para la solicitud de Citas**



**Fig. 3.11 Diagrama de Secuencias para la Búsqueda de Pacientes**

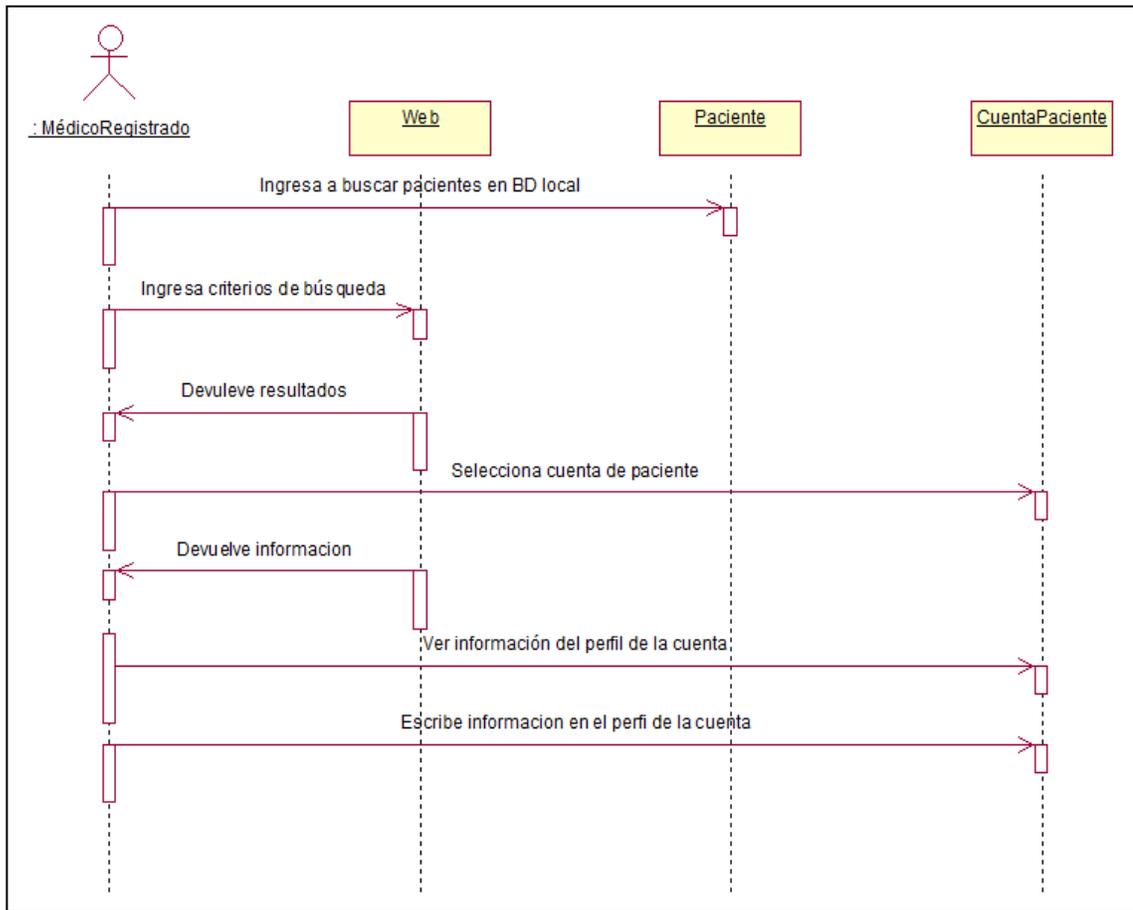
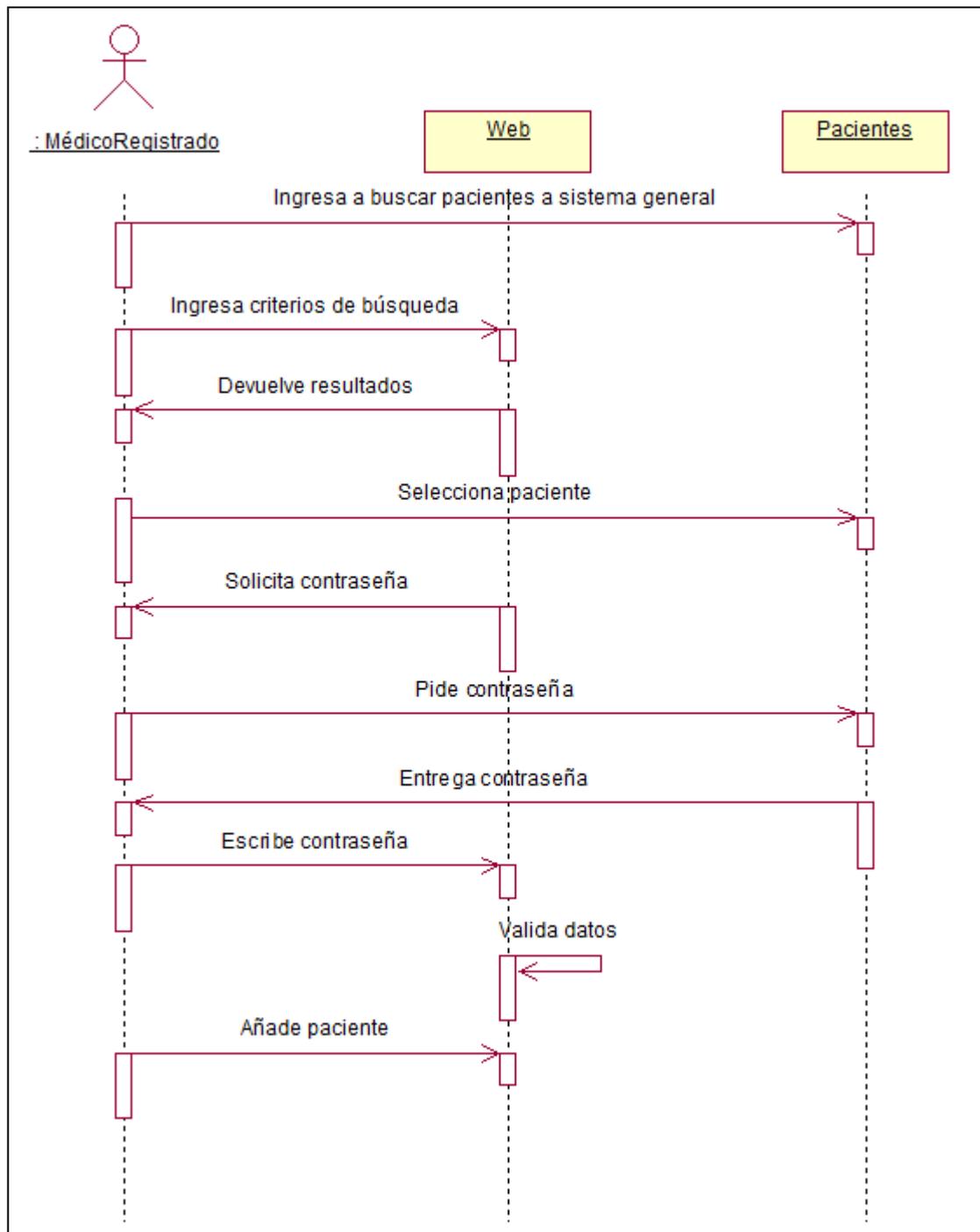


Fig. 3.12 Diagrama de Secuencias para Añadir Pacientes

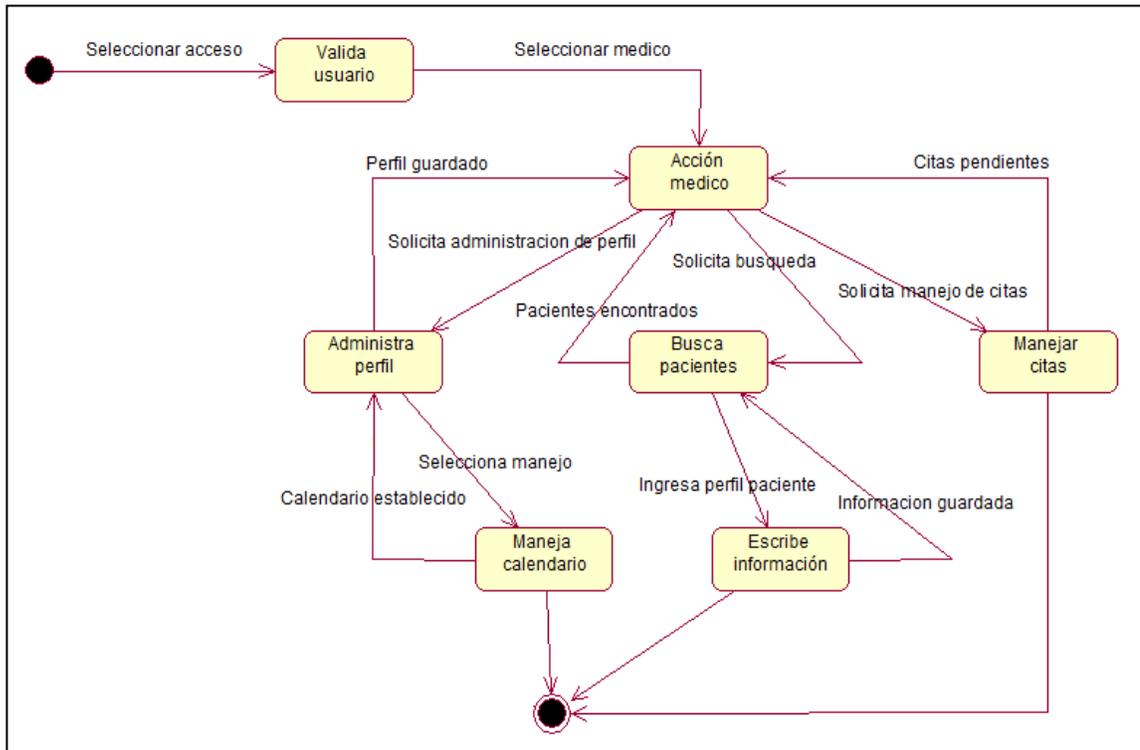


- **Diagrama de Estados**

Muestra tridimensionalmente información que es imposible representar en los casos de uso y en los diagramas de secuencia, se encarga básicamente de mostrar el inicio y el final de un proceso, indicando individualmente los enlaces de navegación entre las entidades actoras.

Cada elemento dispone de un estado de modo que cuando se lleva a cabo dicha condición se cumple alguna acción.

**Fig. 3.13 Diagrama de Estados para la Administración del Perfil Medico**



**Fig. 3.14 Diagrama de Estados para la Administración del Perfil Paciente**

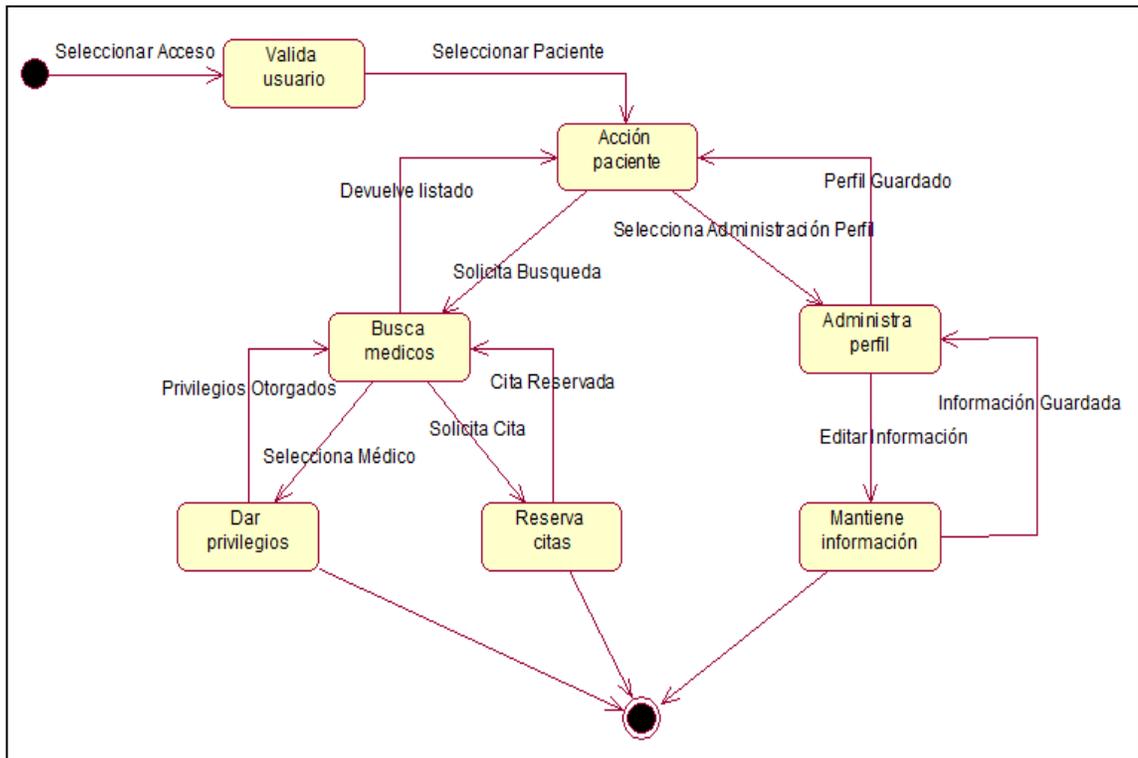
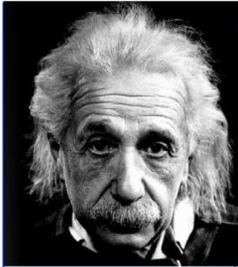


Fig. 3.15 Prototipo de Interfaz de Usuario

# FiMed

Usted Tiene 3 Mensajes Pendientes Salir

[Perfil](#) [Fotos](#) [Medicos](#) [Configuracion](#) [Buscar](#)



**Albert Eistein, 45**  
Av. Don Bosco 3-24  
Telf: 2857394  
Cel: 093847523

[Tratamientos](#)

[Anamnesis](#)

[Ex. Fisico](#)

[Complementario](#)

[Historial](#)



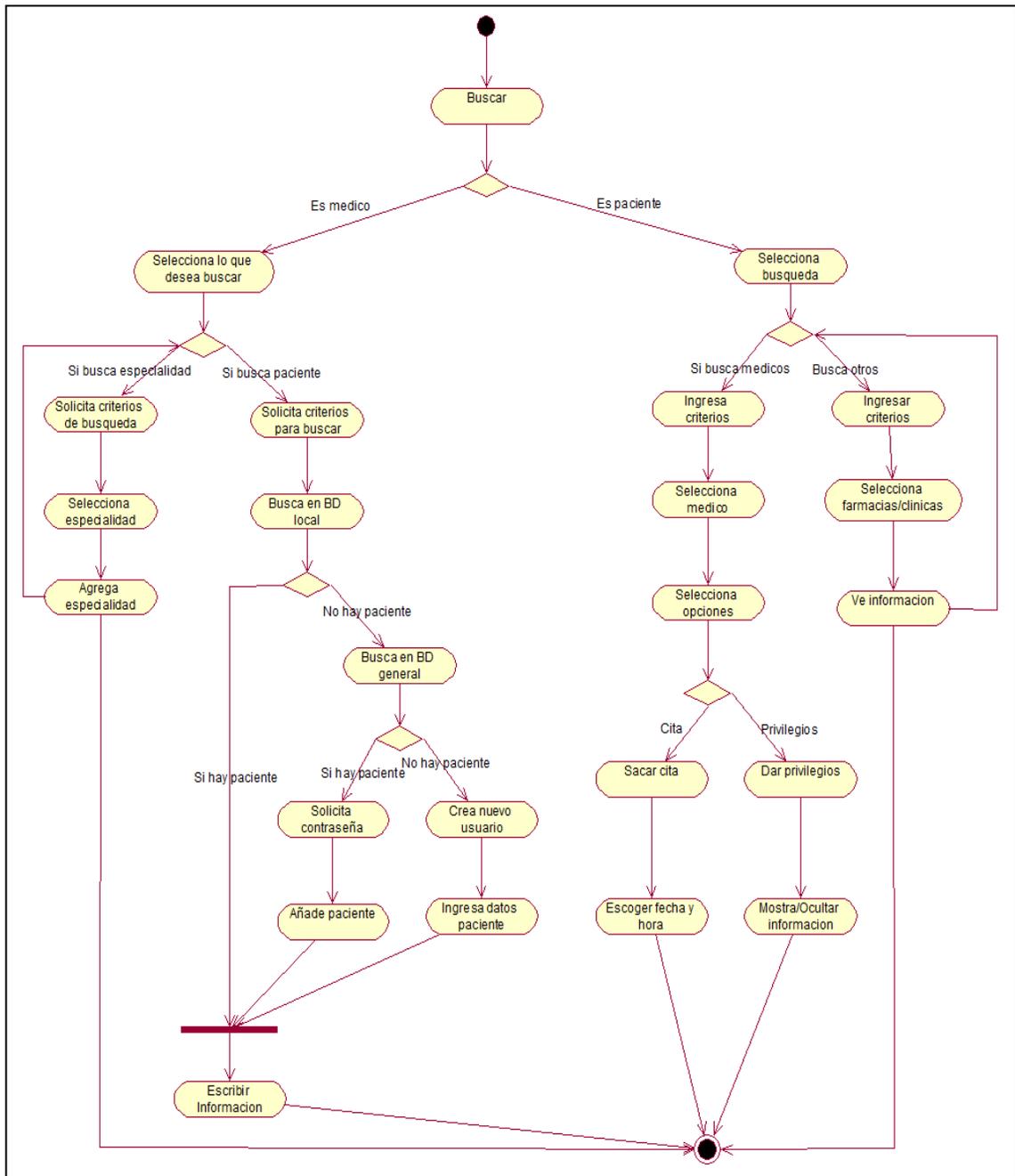
[Contactenos](#) [Acerca de](#) [Trabaje con Nosotros](#) [Privacidad](#)

FiMed 2009

### 3.6 Modelo funcional

Muestra los procesos internos que realiza una función al recibir valores de entrada los cuales necesitan ser procesados para la obtención de resultados.

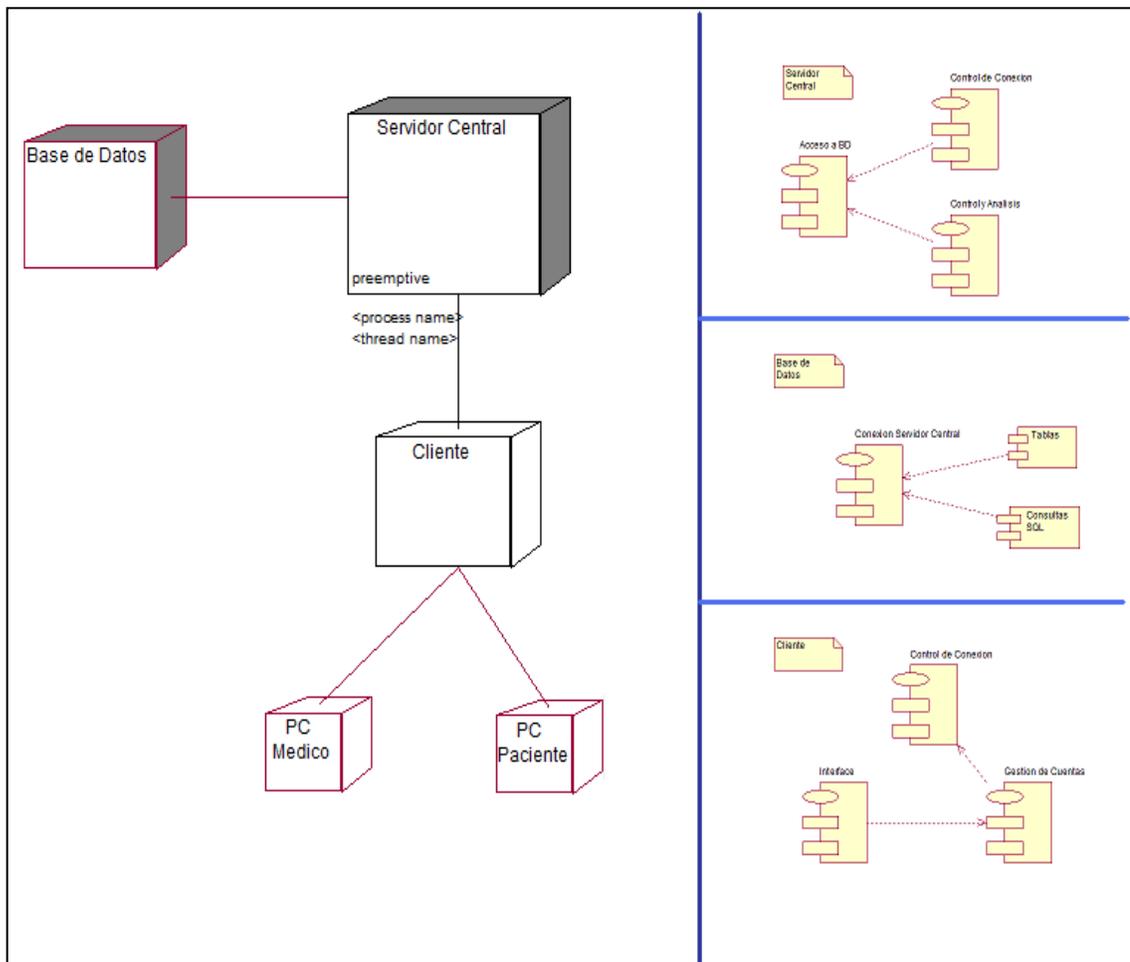
**Fig. 3.16 Diagrama de Actividades para la Función Buscar**



### 3.7 Modelo de configuración

Ayuda a modelar la estructura interna de la aplicación, los componentes de hardware y software deben permitir al cliente operar desde diversos ambientes; así como las interacciones entre servidores y base de datos deben estar claramente definidas y cumplir con ciertos niveles de seguridad.

**Fig. 3.17 Diagrama de Despliegue**

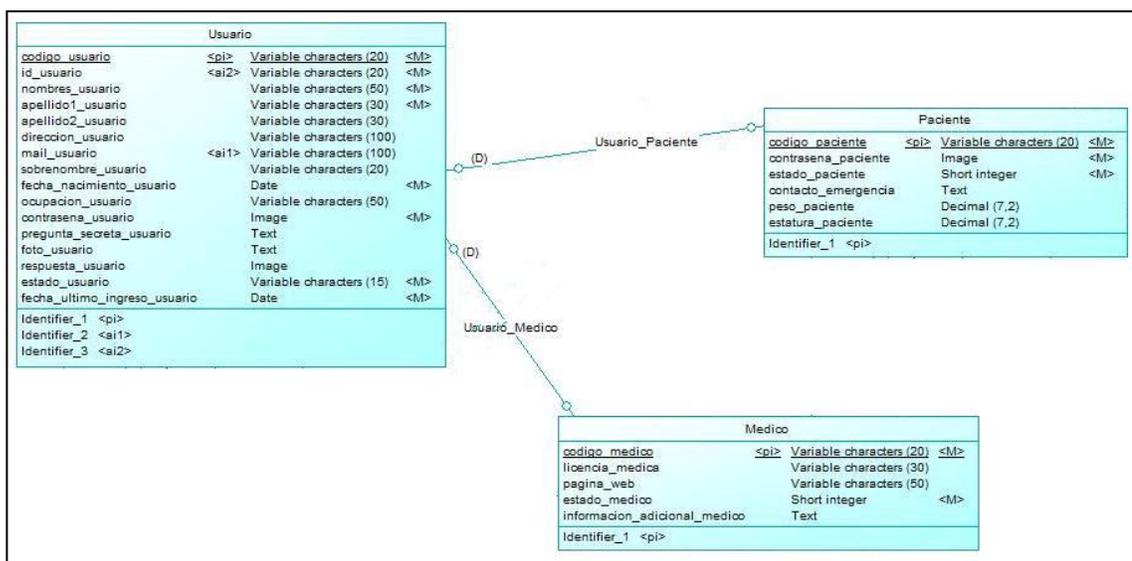


### 3.8 Modelo de la Base de Datos

Esta es una parte muy importante dentro de toda aplicación de software, debido a que en esta instancia se debe definir mediante el modelo Entidad-Relación todas las tablas y las relaciones que tendrán éstas para formar la estructura de la base de datos. En este punto se deben especificar todos los atributos con los que contará cada tabla, sus llaves primarias y los campos que serán necesarios para el ingreso de datos, la relación y su participación con otras tablas, etc.

A continuación se mostrará un extracto de la base de datos, de la tabla de usuario con las relaciones entre pacientes y médicos.

Modelo Entidad-Relación tablas Usuario, Medico, Paciente



Anexo 5 Modelo Entidad Relación

Anexo 6 Modelo Relacional

### **3.9 Conclusión**

Con los requisitos obtenidos en el capítulo anterior, hemos logrado desarrollar en este capítulo los distintos diagramas, que visualmente nos permitirán obtener una perspectiva de todo lo que el sistema va a tratar, logramos visualizar claramente la relación que tienen los actores con el sistema describiendo cada uno de los casos de uso, así como también cada una de las funciones que dichos actores podrán realizar dentro del mismo; al mismo tiempo por medio de modelos adicionales se estableció la estructura interna del proyecto para que de este modo el sistema pueda ser accedido desde cualquier ambiente.

## **CAPÍTULO 4**

### **GESTION DE RIESGOS**

#### **4.1 Introducción**

En el presente capítulo trataremos acerca de los posibles riesgos que se pueden presentar durante el desarrollo del software, los mismos que pueden generar retrasos que vayan en contra de lo planificado y estimado en tiempos y costos, haciendo de esta forma que el software encarezca, se requiera más tiempo para su implementación o en algunos casos se cancele el desarrollo del mismo. Además analizaremos cada uno de estos riesgos con el fin de tener una idea clara y concisa acerca de lo que debemos hacer para evitarlos e impedir catástrofes, seguiremos ciertos pasos, que ayudaran a controlar posibles problemas, como la identificación de cada uno de ellos, el análisis acerca de la incidencia que pueden llegar a tener para posteriormente priorizarlos y poder llevar un control exhaustivo con el fin de determinar si el riesgo aun existe o se lo ha logrado controlar.

Nos basaremos en artículos escritos por algunos autores, quienes explicaran el porqué los productos software fallan y nos guiaremos en cada uno de los consejos que dan para poder evitar el fracaso de los mismos.

## 4.2 Identificación de riesgos

En esta etapa se identificarán los riesgos posibles a presentarse, clasificándolos en dos categorías, funcionales y no funcionales.

**Del Proyecto:** Se denominan riesgos del proyecto aquellos que pueden influir directamente con la planificación del sistema, llegando a generar retrasos en los tiempos de entrega y en consecuencia un aumento en el costo del producto final.

**Técnicos:** son aquellos que afectan el desarrollo del proyecto, los cuales pueden llegar a generar obstáculos difíciles de controlar, este tipo de riesgos usualmente son generados debido al uso de tecnologías de punta u obsoletas, las cuales dificultan la resolución de un problema que en primera instancia no se lo consideraba como tal.

A continuación se enumerarán los riesgos identificados.

### 4.2.1 Del proyecto

#### 1. Planificación no incluye tareas necesarias

Debido a una apresurada planificación se omiten tareas que posteriormente son detectadas e imprescindibles para el funcionamiento del sistema.

#### 2. La presión excesiva en la planificación reduce la productividad.

Mediante análisis psicológico se ha llegado a determinar que el personal involucrado con el desarrollo de proyectos sometido a presión excesiva por parte de mandos más altos reduce mesurablemente su productividad.

#### 3. Las tareas desconocidas del sistema toman más tiempo del esperado en el diseño e implementación

Esto suele suceder cuando una tarea no es dedicada el tiempo suficiente para su planificación, lo cual repercute en la estimación de tiempos totales.

#### **4. Cambio en los requisitos**

Este riesgo puede presentarse por insatisfacción con los requisitos actuales.

#### **5. El usuario final no acepta el producto**

Rechazo al sistema por parte del usuario final, debido a diversos factores como dificultad en el manejo, falta de atracción a ciertos usuarios, etc.

#### **6. Desarrollo orientado a la investigación**

Uso de tecnologías nuevas, lo cual implica que los desarrolladores investiguen sobre el uso de las mismas, al mismo tiempo que se lleva a cabo el proyecto.

#### **7. Incumplimiento en el horario de trabajo**

Los desarrolladores prestan atención a otros aspectos en tiempo de trabajo, dedicando más tiempo a ellos que al desarrollo del software.

#### **8. El personal trabaja más lento de lo esperado**

El personal no rinde al 100% en horas de trabajo.

#### **9. Falta de control en las versiones**

No se sabe con exactitud qué aspectos pertenecen a cada versión, debido a que no se ha llevado un registro exhaustivo por parte de los encargados.

#### **10. Falta de personal para el desarrollo**

Personal escaso para el desarrollo de cada tarea, lo cual implica que al poco personal con el que se cuenta para el desarrollo se le acumulen tareas.

#### **11. Falta de participación de los programadores**

Los programadores demuestran poco interés en el desarrollo de las tareas del sistema, lo cual puede llevar a un retraso en el tiempo de culminación de las mismas.

## **12. Mala interpretación de requisitos.**

Los requisitos no fueron explicados implícitamente, lo cual hace que los desarrolladores interpreten a su manera y desarrollen mal las tareas.

## **13. Desacuerdos entre el personal**

Falta de comunicación entre el personal del proyecto conlleva a malos entendidos.

## **14. Reducción del tiempo planificado para cada tarea**

Se lo realiza con el fin de terminar pronto el sistema, de esta manera se omiten pruebas necesarias y se descuida en la calidad del mismo.

## **15. Falta de motivación**

Se puede suscitar debido a largos periodos de trabajo haciéndolo al proyecto monótono y aburrido.

## **16. Asignación de tiempos demasiado cortos para una tarea**

Mala estimación en el tiempo que requiere una tarea, lo cual origina retrasos en la culminación del sistema.

## **17. Creer que se dispone de más tiempo del que realmente se tiene**

Llevar a la ligera el desarrollo del proyecto pensando que se dispone aún de una gran cantidad de tiempo, lo cual hará que tareas posteriores sean realizadas en menor tiempo.

## **18. Planificación excesivamente optimista**

Basado en la falta de experiencia en el desarrollo de proyectos similares se puede llegar a estimar tiempos o costos con un alto margen de error.

## **19. Planificar ponerse al día mas adelante**

Llevar las tareas de manera relajada debido a que se dispone de tiempo para realizarlas.

## **20. Exceso de requerimientos**

Demasiados requerimientos hacen que el sistema se alargue mucho y su estimación en el tiempo total del proyecto no sea predecible.

### **4.2.2 Técnicos**

#### **1. Las herramientas de desarrollo no están disponibles, o no funcionan como se esperaba**

No se cuenta con todas las herramientas conocidas para el desarrollo o simplemente se obtienen versiones de prueba para su utilización.

#### **2. El aprendizaje de la nueva herramienta de desarrollo es más larga de lo esperado**

Para llegar a disponer de las destrezas suficientes con el fin de efectuar tareas en una nueva herramienta requiere de un proceso de investigación previo el cual puede resultar extenso.

#### **3. Diseño de la interfaz inadecuada, lo cual requiere volver a diseñar e implementar**

Las interfaces diseñadas para cada tipo de usuario no son las apropiadas para cada uno de ellos.

#### **4. Un diseño demasiado complejo acarrea complicaciones innecesarias e improductivas en la implementación**

Los procesos deben ser efectuados de la manera más simple posible pero siempre y cuando se cumpla con el requisito anidado a dicha tarea.

#### **5. Los componentes desarrollados por separado no se pueden integrar de forma sencilla**

La incompatibilidad entre componentes puede provocar pérdidas de tiempo.

## **6. Falta de documentación del proyecto**

No se lleva la documentación adecuada desde el inicio del proyecto, lo cual hace que no se registren partes importantes en el desarrollo del proyecto.

## **7. Concurrencia a la pagina**

Demasiado acceso de usuarios a la pagina en un mismo instante lo cual exige mayor procesamiento al gestor de base de datos retardando los tiempos por consulta.

## **8. Carga impredecible**

La WebApp estará sometida a variaciones en los accesos diarios pudiendo presentarse picos en ciertos horarios del día.

## **9. Desempeño pobre de la WebApp**

El mal uso de las tecnologías o el uso de tecnologías obsoletas pueden generar pesadas WebApp lo cual se verá reflejado en su acceso

## **10. Disponibilidad**

Al estar el sistema disponible 24x7x365 implica que sus actualizaciones sean programadas a ciertas horas del día lo cual se tiene que tomar a consideración debido a los posibles accesos en otros usos horarios

## **11. Sensibilidad al contenido**

El contenido que se presente dentro de la WebApp tiene que estar claramente definido con el fin de que el usuario final se sienta cómodo y encuentre útil la información visualizada

## **12. Seguridad en la base de datos**

La información que la WebApp almacenara se considera de alta confidencialidad, para lo cual el administrador de la base de datos debe brindar ciertos niveles de seguridad para evitar accesos indeseados a información privada.

### **13. Seguridad de la WebApp**

La WebApp debe cumplir con ciertos estándares de seguridad los cuales brinden la suficiente confianza al usuario final para que esta haga uso de la aplicación sin problema alguno

### **14. Colapso del servidor**

Este es un riesgo que se lo debe tomar con la mayor importancia posible puesto que muchos factores pueden provocar este suceso, como por ejemplo por la falta del control de concurrencia.

### **15. Velocidad de conexión**

Este riesgo está dado a nivel del cliente y depende directamente del ISP.

### **16. Incompatibilidad con el navegador**

Los diferentes navegadores que el mercado presenta hoy en día implican que la WebApp debe ser desarrollada con el uso de tecnologías adecuadas y compatibles con la mayoría de estos para evitar problemas del lado del usuario final.

## **4.3 Análisis de riesgos**

Una vez identificado cada uno de los posibles riesgos a suscitarse, procederemos a analizarlos uno por uno con el fin de determinar cuan riesgoso es el riesgo. Mediante formulas determinaremos la exposición de riesgos que es la probabilidad de perdida no esperada, es decir la magnitud de perdida.

**Tabla 4.1 Análisis de Riesgos**

RIESGO		PROBABILIDAD DE PERDIDA	MAGNITUD DE LA PERDIDA (sem)	EXPOSICION A RIESGO (sem)
<b>Del Proyecto</b>				
1.	Planificación no incluye tareas necesarias	30%	6	1,8
2.	La presión excesiva en la planificación reduce la productividad.	15%	3	0,45
3.	Las tareas desconocidas del sistema toman más tiempo del esperado en el diseño e implementación	60%	5	3
4.	Cambio en los requisitos	40%	10	4
5.	El usuario final no acepta el producto	50%	8	4
6.	Desarrollo orientado a la investigación	65%	8	5,2
7.	Incumplimiento en el horario de trabajo	10%	2	0,2
8.	El personal trabaja más lento de lo esperado	5%	1	0,05
9.	Falta de control en las versiones	5%	2	0,1
10.	Falta de personal para el desarrollo	40%	4	1,6
11.	Falta de participación de los programadores	20%	3	0,6
12.	Mal interpretación de requisitos.	30%	7	2,1
13.	Desacuerdos entre el personal	15%	2	0,3
14.	Reducción del tiempo planificado para cada tarea	50%	4	2
15.	Falta de motivación	30%	2	0,6
16.	Asignación de tiempos demasiado cortos para una tarea	25%	4	1
17.	Crear que se dispone de más tiempo del que realmente se tiene	20%	5	1
18.	Planificación excesivamente optimista	30%	5	1,5
19.	Planificar ponerse al día más adelante	10%	4	0,4
20.	Exceso de requerimientos	45%	12	5,4
<b>Técnicos</b>				
1.	Las herramientas de desarrollo no están disponibles, o no funcionan como se esperaba	60%	15	9
2.	El aprendizaje de la nueva herramienta de desarrollo es más larga de lo esperado	55%	9	4,95
3.	Diseño de la interfaz inadecuada, lo cual requiere volver a diseñar e implementar	70%	7	4,9
4.	Un diseño de masiado complejo acarrea complicaciones innecesarias e improductivas en la implementación	45%	7	3,15
5.	Los componentes desarrollados por separado no se pueden integrar de forma sencilla	25%	5	1,25
6.	Falta de documentación del proyecto	2%	3	0,06
7.	Concurrencia a la página	50%	2	1
8.	Carga impredecible	40%	1	0,4
9.	Desempeño pobre de la WebApp	10%	14	1,4
10.	Disponibilidad	5%	2	0,1
11.	Sensibilidad al contenido	3%	5	0,15
12.	Seguridad en la base de datos	50%	4	2
13.	Seguridad de la WebApp	45%	4	1,8
14.	Colapso del servidor	60%	2	1,2
15.	Velocidad de conexión	5%	1	0,05
16.	Incompatibilidad con el navegador	15%	8	1,2

#### **4.4 Priorización de riesgos**

La mayoría de los proyectos de software invierten el 80% en su presupuesto general en corregir el 20% de sus problemas, es decir los primeros errores suelen ser los más importantes, debido a que este análisis es subjetivo se podría considerar la priorización de un error que no necesariamente se encuentre entre los primeros.

Una vez analizados cada riesgo procedemos a priorizarlos de mayor a menor, lo cual nos guiamos en la columna de Exposición de Riesgo. Basados en dicha priorización se determinará los riesgos a los cuales se les tiene que dar mayor control, sin olvidarse de los demás.

Tabla 4.2 Priorización de Riesgos

RIESGO	PROBABILIDAD DE PERDIDA	MAGNITUD DE LA PERDIDA (sem)	EXPOSICION A RIESGO (sem)
1. Las herramientas de desarrollo no están disponibles, o no funcionan como se esperaba	60%	15	9
20. Exceso de requerimientos	45%	12	5,4
6. Desarrollo orientado a la investigación	65%	8	5,2
2. El aprendizaje de la nueva herramienta de desarrollo es más larga de lo esperado	55%	9	4,95
3. Diseño de la interfaz inadecuada, lo cual requiere volver a diseñar e implementar	70%	7	4,9
4. Cambio en los requisitos	40%	10	4
5. El usuario final no acepta el producto	50%	8	4
4. Un diseño demasiado complejo acarrea complicaciones innecesarias e improductivas en la implementación	45%	7	3,15
3. Las tareas desconocidas del sistema toman más tiempo del esperado en el diseño e implementación	60%	5	3
12. Mala interpretación de requisitos.	30%	7	2,1
14. Reducción del tiempo planificado para cada tarea	50%	4	2
12. Seguridad en la base de datos	50%	4	2
13. Seguridad de la WebApp	45%	4	1,8
1. Planificación no incluye tareas necesarias	30%	6	1,8
10. Falta de personal para el desarrollo	40%	4	1,6
18. Planificación excesivamente optimista	30%	5	1,5
9. Desempeño pobre de la WebApp	10%	14	1,4
5. Los componentes desarrollados por separado no se pueden integrar de forma sencilla	25%	5	1,25
14. Colapso del servidor	60%	2	1,2
16. Incompatibilidad con el navegador	15%	8	1,2
16. Asignación de tiempos demasiado cortos para una tarea	25%	4	1
17. Crear que se dispone de más tiempo del que realmente se tiene	20%	5	1
7. Concurrencia a la página	50%	2	1
11. Falta de participación de los programadores	20%	3	0,6
15. Falta de motivación	30%	2	0,6
2. La presión excesiva en la planificación reduce la productividad.	15%	3	0,45
19. Planificar ponerse al día mas adelante	10%	4	0,4
8. Carga impredecible	40%	1	0,4
13. Desacuerdos entre el personal	15%	2	0,3
7. Incumplimiento en el horario de trabajo	10%	2	0,2
11. Sensibilidad al contenido	3%	5	0,15
9. Falta de control en las versiones	5%	2	0,1
10. Disponibilidad	5%	2	0,1
6. Falta de documentación del proyecto	2%	3	0,06
8. El personal trabaja más lento de lo esperado	5%	1	0,05
15. Velocidad de conexión	5%	1	0,05

## 4.5 Control de riesgos

Una vez que hemos logrado identificar cada uno de los riesgos y posteriormente analizarlos, procedemos al plan de mitigación de riesgos que consiste en dar pautas de cómo reducir la probabilidad de que el riesgo se presente, o en el caso de que el riesgo sea inminente tener maneras de controlarlo para evitar catástrofes.

### 4.5.1 Del proyecto

<b>Riesgo:</b>	Planificación no incluye tareas necesarias
<b>Métodos de control</b>	
<ul style="list-style-type: none"><li>• Dedicar el tiempo necesario para la correcta planificación del software</li><li>• Realizar un exhaustivo levantamiento de requisitos con el fin de conocer lo que se necesita.</li><li>• Dividir las tareas grandes en más pequeñas y específicas.</li></ul>	

<b>Riesgo:</b>	La presión excesiva en la planificación reduce la productividad.
<b>Métodos de control</b>	
<ul style="list-style-type: none"><li>• Realizar un cálculo adecuado del tiempo disponible para evitar presiones.</li><li>• Cumplir con las actividades propuestas en el diagrama de Gantt.</li><li>• Crear lazos de respeto entre el grupo de ingenieros y el de los desarrolladores.</li></ul>	

<b>Riesgo:</b>	Las tareas desconocidas del sistema toman más tiempo del esperado en el diseño e implementación.
<b>Métodos de control</b>	
<ul style="list-style-type: none"><li>• Analizar detenidamente cada una de las tareas con el fin de conocer los objetivos a cumplir en cada tarea.</li><li>• Estimar el tiempo que durara el desarrollo de cada tarea y a este agregarle</li></ul>	

un tiempo de holgura el cual ayudara en caso de retrasos.

<b>Riesgo:</b>	Cambio en los requisitos
<b>Métodos de control</b>	
<ul style="list-style-type: none"><li>• Definir claramente cada uno de los requisitos.</li><li>• Analizar el cambio con el fin de adaptarlo al proyecto</li><li>• En caso de un software constantemente cambiante analizar un ciclo de vida que se adapte a dichos cambios.</li><li>• Analizar los futuros riesgos a presentarse debido a dichos cambios.</li></ul>	

<b>Riesgo:</b>	El usuario final no acepta el producto
<b>Métodos de control</b>	
<ul style="list-style-type: none"><li>• Dedicar tiempo al diseño adecuado de la interfaz con el fin de hacerlo más amigable e intuitivo.</li><li>• Analizar el mercado objetivo con el fin de crear interfaces que vayan de acuerdo a cada categoría de usuario.</li><li>• Analizar los motivos de dicho rechazo.</li><li>• Realizar pruebas de usuario antes de lanzar el sistema con el fin de verificar incoherencias.</li><li>• Llevar un seguimiento de estándares de calidad al software</li></ul>	

<b>Riesgo:</b>	Desarrollo orientado a la investigación
<b>Métodos de control</b>	
<ul style="list-style-type: none"><li>• Evitar el uso de herramientas nuevas que pueden retrasar el proyecto.</li><li>• No dejarse llevar por la funcionalidad que ofrece la nueva herramienta.</li><li>• En caso de ser necesario el uso de una nueva herramienta, agregar</li></ul>	

tiempos adicionales para la investigación.

- En la etapa del levantamiento de requisitos agregar el tiempo suficiente para investigar sobre el tema que se va a desarrollar.
- No intentar investigar y maximizar la velocidad de desarrollo al mismo tiempo.

<b>Riesgo:</b>	Incumplimiento en el horario de trabajo
----------------	---

<b>Métodos de control</b>	
---------------------------	--

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• Realizar diagramas de Gantt que permitan conocer la duración de las tareas en el tiempo.</li><li>• Establecer horarios adecuados para su cumplimiento.</li><li>• Evitar periodos largos de trabajo</li><li>• Evitar que este incumplimiento sobrepase la estimación general del tiempo.</li><li>• Recompensar el tiempo perdido en caso de ser necesario.</li></ul> |  |
|---|--|

<b>Riesgo:</b>	El personal trabaja más lento de lo esperado
----------------	--

<b>Métodos de control</b>	
---------------------------	--

- |  |  |
|--|--|
| <ul style="list-style-type: none"><li>• Evitar largos periodos de trabajo que produzcan cansancio en el personal.</li><li>• Establecer tiempos adecuados de descanso para recuperar energías.</li><li>• Incentivar al personal con el fin de que rindan más.</li><li>• Tratar de no sobrepasar el tiempo de holgura establecido.</li></ul> |  |
|--|--|

<b>Riesgo:</b>	Falta de control en las versiones
----------------	-----------------------------------

<b>Métodos de control</b>	
---------------------------	--

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• Asignar la tarea de documentación a un miembro del personal.</li><li>• Llevar un control de los cambios presentados en cada versión del sistema.</li><li>• La documentación de las versiones debe ser registrada cronológicamente, almacenando información de responsables.</li></ul> |  |
|---|--|

- Identificar claramente cuando se requiere el lanzamiento de una nueva versión.

<b>Riesgo:</b>	Falta de personal para el desarrollo
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Estimar el tiempo de desarrollo en base al número de integrantes.</li> <li>• Asignar las tareas equitativamente entre los desarrolladores.</li> </ul>	

<b>Riesgo:</b>	Falta de participación de los programadores
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Incentivar la participación de los desarrolladores.</li> <li>• Tomar conciencia que cada miembro es de vital importancia en el desarrollo del sistema.</li> <li>• Realizar reuniones periódicas con el fin de verificar el cumplimiento de las tareas.</li> <li>• El grupo debe comprometerse a trabajar responsable y disciplinadamente desde el inicio del proyecto.</li> </ul>	

<b>Riesgo:</b>	Mala interpretación de requisitos.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Realizar el número de reuniones necesarias con el fin de definir claramente lo que se requiere.</li> <li>• Incentivar a la participación de las partes involucradas con el fin de llegar a acuerdos mutuos.</li> <li>• Documentar las reuniones efectuadas y lo acordado en cada una de ellas.</li> <li>• En caso de dudas realizar nuevas reuniones.</li> </ul>	

<ul style="list-style-type: none"> <li>• Evitar hacer lo que uno piensa.</li> </ul>
---

<b>Riesgo:</b>	Desacuerdos entre el personal
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Tratar de crear relaciones respetuosas.</li> <li>• Evitar sesiones de trabajo tensas que puedan generar desacuerdos.</li> <li>• Distribuir el trabajo equitativamente con el fin de evitar sobrecargos.</li> <li>• Debatir el tema las veces necesarias con el fin de llegar a un mutuo acuerdo.</li> <li>• Buscar nuevas formas de resolver el problema que satisfagan ambas necesidades.</li> </ul>	

<b>Riesgo:</b>	Falta de comunicación entre el personal del proyecto conlleva a malos entendidos.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Llevar un registro completo de cada cambio realizado, incluyendo al responsable de dicho cambio.</li> <li>• Respetar las tareas asignadas para cada persona, evitando manipular o modificar tareas ajenas.</li> <li>• En caso de confusión llegar a un acuerdo mutuo con el fin de evitar futuros roces.</li> </ul>	

<b>Riesgo:</b>	Reducción del tiempo planificado para cada tarea.
<b>Métodos de control</b>	

<ul style="list-style-type: none"> <li>• Analizar detalladamente si el tiempo final es suficiente para la ejecución de la tarea.</li> <li>• Respetar el tiempo planificado para cada tarea.</li> <li>• Basarse en experiencias anteriores con el fin de establecer tiempos aproximados para el desarrollo de cada tarea.</li> <li>• En caso de la culminación temprana de una tarea, invertir este tiempo restante en verificar errores.</li> </ul>
---

<b>Riesgo:</b>	Falta de motivación
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Evitar largas jornadas de trabajo seguidas, las cuales pueden hostigar y aburrir.</li> <li>• Asignar periodos de tiempo para relajación.</li> </ul>	

<b>Riesgo:</b>	Asignación de tiempos demasiado cortos para una tarea
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Hacer uso del tiempo de holgura asignado entre las tareas.</li> <li>• En la fase de planificación realizar un análisis completo sobre el tiempo aproximado que tomara una tarea.</li> <li>• Crear horarios alternativos con el fin de cumplir con el desarrollo de la tarea y no retrasar las siguientes.</li> <li>• Del personal disponible asignar miembros adicionales para la ejecución de la tarea.</li> <li>• Evitar las pérdidas de tiempo que retrasaran aun más la tarea.</li> <li>• Cumplir responsablemente con los horarios asignados para cada tarea.</li> </ul>	

<b>Riesgo:</b>	Creer que se dispone de más tiempo del que realmente se tiene.
<b>Métodos de control</b>	

- Respetar los horarios planificados para la ejecución de la tarea.
- Dividir la tarea en procesos más cortos para el cumplimiento y control en la ejecución de las tareas.
- No desperdiciar el tiempo, puesto que si la tarea finaliza antes, este tiempo puede servirnos en otras tareas más complejas.
- No posponer tareas cortas para realizarla en otro momento.

<b>Riesgo:</b>	Planificación excesivamente optimista
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Basar la estimación de tiempos y costos en la experiencia.</li> <li>• Analizar detenidamente la duración de cada tarea o cada proceso por tarea, para determinar un tiempo más preciso.</li> <li>• Hacer uso de varias técnicas y herramientas para la estimación.</li> <li>• Tomarse el tiempo necesario para la planificación del proyecto.</li> <li>• No creer que el proyecto es tan fácil como se lo ve a simple vista.</li> </ul>	

<b>Riesgo:</b>	Planificar ponerse al día mas adelante
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• No disponer del tiempo asignado a otras tareas.</li> <li>• Tratar de recuperar el tiempo perdido lo más pronto posible.</li> <li>• Crear horarios alternativos de trabajo que permitan recuperar el tiempo perdido.</li> <li>• Hacer uso de los tiempos de holgura, sin abusar de ellos.</li> </ul>	

<b>Riesgo:</b>	Exceso de requerimientos
<b>Métodos de control</b>	

- Establecer un inicio y un fin del proyecto.
- Llevar un control de cada requisito para la estimación de tiempos y costos.
- Clasificar los requisitos más indispensables y necesarios.
- Anular o resumir requisitos redundantes o de poca importancia.
- Llegar a un acuerdo entre las partes involucradas de modo se conozca los requisitos finales.

**4.5.2 Técnicos**

<b>Riesgo:</b>	Las herramientas de desarrollo no están disponibles, o no funcionan como se esperaba.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Verificar las características de las herramientas antes de hacer uso de ellas.</li> <li>• Dedicar un tiempo adicional a la investigación de la herramienta a utilizar.</li> <li>• Consultar con expertos en el uso de dicha herramienta con el fin de conocer si esta resolverá nuestros problemas.</li> <li>• Disponer de la versión completa de la herramienta a utilizarse.</li> <li>• Disponer de librerías de ayuda para consultas en caso de problemas.</li> </ul>	

<b>Riesgo:</b>	El aprendizaje de la nueva herramienta de desarrollo es más largo de lo esperado.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Crear horarios alternativos para la investigación de la nueva herramienta.</li> <li>• Considerar el uso de una herramienta ya conocida.</li> </ul>	

<ul style="list-style-type: none"> <li>• Analizar el tiempo que se necesita para el aprendizaje de una herramienta.</li> </ul>
--

<b>Riesgo:</b>	Diseño de la interfaz inadecuada, lo cual requiere volver a diseñar e implementar.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Dedicar el tiempo necesario a la fase de diseño con el fin de evitar estos inconvenientes.</li> <li>• Conocer bien a qué tipo de usuarios se desea llegar para el diseño de una interfaz adecuada.</li> <li>• En caso de ser necesario diseñar una interfaz neutra la cual se aplique a varios entornos.</li> <li>• Investigar sobre técnicas para el diseño de una WebApp.</li> </ul>	

<b>Riesgo:</b>	Un diseño demasiado complejo acarrea complicaciones innecesarias e improductivas en la implementación.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Hacer las cosas simples y atractivas, sin abusar de colores ni formas.</li> <li>• Hacer uso de una adecuada administración del contenido.</li> <li>• No dedicar tiempo excesivo a la fase de diseño de la interfaz.</li> <li>• Seguir políticas y estándares para el diseño de la aplicación.</li> </ul>	

<b>Riesgo:</b>	Los componentes desarrollados por separado no se pueden integrar de forma sencilla.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Establecer políticas que permitan la integración entre varios componentes desarrollados por separado.</li> <li>• Dedicar un tiempo para la integración de los componentes.</li> <li>• Realizar pruebas para verificar que el componente trabaje de acuerdo a lo</li> </ul>	

programado.

- Realizar una prueba de integración, con el fin de verificar si los componentes acoplados funcionan correctamente.

<b>Riesgo:</b>	Falta de documentación del proyecto.
----------------	--------------------------------------

<b>Métodos de control</b>	
---------------------------	--

- |  |  |
|--|--|
| <ul style="list-style-type: none"><li>• Asignar un responsable a la tarea de documentación del sistema.</li><li>• Llevar un control de los cambios efectuados en la aplicación.</li><li>• Cada tarea debe ser documentada, indicando sus responsables e información sobre su funcionamiento.</li><li>• Hacer uso de las herramientas de ingeniería inversa, las cuales aportaran con información sobre el proyecto.</li><li>• A medida que el proyecto avanza ir documentado dicho avance.</li></ul> |  |
|--|--|

<b>Riesgo:</b>	Concurrencia a la página.
----------------	---------------------------

<b>Métodos de control</b>	
---------------------------	--

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• Analizar el uso de una base de datos que soporte la concurrencia planificada</li><li>• Evitar el uso de consultas SQL que generen retardo en los tiempos de respuesta.</li><li>• Analizar los métodos para el control de concurrencia que puedan afectar a la base de datos.</li><li>• El servidor en el cual la página se encuentre alojada deberá soportar altos índices de concurrencia.</li></ul> |  |
|---|--|

<b>Riesgo:</b>	Carga impredecible.
----------------	---------------------

<b>Métodos de control</b>	
---------------------------	--

- El sistema debe estar preparado para el acceso impredecible de los usuarios.
- Hacer uso de técnicas que distribuyan el trabajo entre varios servidores.
- Alojarse en servidores que brinden un rendimiento adecuado, confiabilidad y fiabilidad.
- Hacer uso de técnicas que eviten el colapso de los servidores.

<b>Riesgo:</b>	Desempeño pobre de la WebApp.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Hacer uso de herramientas que permitan generar aplicaciones ligeras y agradables para el usuario.</li> <li>• Seguir estándares y políticas que permitan crear sistemas eficientes.</li> <li>• Evitar el uso de tecnologías obsoletas.</li> <li>• Evitar el uso de consultas SQL complejas que generen retardo en los tiempos de respuesta.</li> <li>• Evitar el uso de frames.</li> </ul>	

<b>Riesgo:</b>	Disponibilidad.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Programar tareas de mantenimiento en horarios de poca concurrencia.</li> <li>• Analizar la tasa de concurrencia por medio de herramientas que nos permitan visualizar horarios de alto y bajo uso.</li> <li>• Analizar que la página al estar en internet deberá estar disponible en los diferentes horarios de cada país.</li> <li>• Hacer uso de servidores que garanticen la disponibilidad de la aplicación en todo momento.</li> <li>• Evitar que el servidor colapse y la página se vuelva inactiva.</li> <li>• En caso de que por situaciones de mantenimiento la página se vuelva inactiva, informar a los usuarios el tiempo que se encontrará</li> </ul>	

deshabilitada.

<b>Riesgo:</b>	Sensibilidad al contenido.
<b>Métodos de control</b>	
<ul style="list-style-type: none"><li>• La WebApp deberá manejar contenidos de acuerdo a los usuarios que la vayan a visualizar.</li><li>• La información deberá estar distribuida de manera que sea fácil navegar por ella.</li><li>• Agrupar la información de acuerdo a contenidos que tengan las mismas características.</li></ul>	

<b>Riesgo:</b>	Seguridad en la base de datos.
<b>Métodos de control</b>	
<ul style="list-style-type: none"><li>• Hacer uso de técnicas de encriptación de la información almacenada dentro de la base de datos.</li><li>• Cumplir con reglas básicas que garanticen el acceso a la información.</li><li>• No permitir accesos indeseados de personas malintencionadas al sistema.</li><li>• No escatimar esfuerzos por resguardar la información de la base de datos.</li><li>• Llevar respaldos periódicos de la información almacenada en la base de datos.</li></ul>	

<b>Riesgo:</b>	Seguridad de la WebApp.
<b>Métodos de control</b>	
<ul style="list-style-type: none"><li>• Contratar certificados de seguridad, los cuales brinden confianza al usuario.</li><li>• Validar el número de accesos incorrectos.</li><li>• Evitar suplantaciones de identidad.</li></ul>	

- Permitir a los usuarios la recuperación de sus cuentas en caso del olvido de sus contraseñas.
- Garantizar al usuario que está en la página correcta y no en una clonación.
- Caducar la sesión después de un lapso de inactividad.

<b>Riesgo:</b>	Colapso del servidor.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Disponer de un servidor que se habilite en caso de emergencias.</li> <li>• Contar con medidas de contingencia que agilicen la solución del problema.</li> <li>• Realizar pruebas de funcionamiento antes de poner en marcha el sistema.</li> <li>• Someterle a pruebas rigurosas al sistema.</li> <li>• Designar un responsable capacitado para lo solución ágil de dicho problema.</li> </ul>	

<b>Riesgo:</b>	Velocidad de conexión.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Informar al usuario sobre los tiempos de subida de datos de acuerdo al ancho de banda.</li> <li>• Orientar la pagina a un ancho de banda promedio.</li> <li>• En caso de ser posible, generar una página adicional que requiera de una menor velocidad de acceso.</li> </ul>	

<b>Riesgo:</b>	Incompatibilidad con el navegador.
<b>Métodos de control</b>	
<ul style="list-style-type: none"> <li>• Hacer uso de herramientas que permitan visualizar la WebApp en los diferentes entornos disponibles.</li> <li>• Generar una página compatible con la mayoría de los navegadores</li> </ul>	

disponibles en el mercado.

- Evitar el uso de tecnologías que requieran una instalación adicional por parte de los usuarios.
- En caso de que el navegador actual no soporte la página, sugerir navegadores que soporten dicha tecnología.

## **4.6 Conclusión**

En este capítulo hemos logrado identificar los posibles riesgos que el proyecto pudiera tener, posteriormente los hemos analizado detenidamente con el fin de conocer los motivos por los cuales un riesgo puede ser inminente, además a cada riesgo analizado lo hemos priorizado de tal modo que se pueda determinar la probabilidad de que dicho riesgo se presente a lo largo del proyecto; esta priorización está basada en la experiencia y calculada probabilísticamente, lo cual nos permite percibir el retraso que un riesgo pueda causar, de esta manera se determinaron los riesgos a los cuales se les debe someter a un mayor control dada su alta probabilidad de suscitarse pero sin olvidarse de los demás. Dictaminamos además las posibles formas de reducir esta probabilidad; o en caso de ser imposible anular el riesgo se han implementado políticas que nos ayudaran a controlar al riesgo en el momento que este se presente.

## **CAPÍTULO 5**

### **DISEÑO DEL SISTEMA**

#### **5.1 Introducción**

El diseño web es una parte muy importante en cada uno de este tipo de aplicaciones, debido a que se pueden tener páginas con mucha funcionalidad y con un diseño muy pobre, es por eso que la funcionalidad y el diseño deben ir de la mano, de tal manera que los usuarios se sientan conformes y a gusto con lo que les presenta una WebApp.

En el siguiente capítulo se tratara sobre el diseño de la página web, en el cual se procederá a refinar la información analizada en el capítulo dos; de tal forma que se establecerá el contenido de cada una de las páginas de una manera organizada y ordenada; y mediante programas de diseño grafico se podrá diseñar los prototipos y el bosquejo final de cada página las cuales corresponderán a cada categoría de usuario.

Además se establecerá la estructura de la página web, lo cual hará que los usuarios naveguen dentro de ella de una manera fácil, ágil y sin complicaciones. Del mismo modo se definirá el aspecto visual de la WebApp; y se podrá representar de forma grafica como el usuario navegará dentro de la aplicación.

## **5.2 Diseño de la interfaz**

El diseño de la interfaz es muy importante, pues es la primera impresión de la WebApp. Por esta razón, para lograr el diseño de una interfaz eficiente se deben considerar varios aspectos importantes para que la página sea fácil de usar, hay diversas actividades que se deberán cumplir secuencialmente con el fin de lograr el objetivo.

### **1.- Refinación de la información del modelo de análisis.**

La página consta de abundante información, la cual debe ser organizada ordenadamente partiendo desde el punto de vista del usuario como de la información que éste podrá visualizar.

La página dispone de dos categorías de usuarios; paciente y medico, siendo para cada uno de ellos distinta la información y las posibles acciones a presentarse.

#### **El usuario medico**

Es el encargado de la administración de sus pacientes, de su propio perfil, de sus citas, sus lugares de trabajo, horarios de atención, etc.

Esta información se la agrupó según la prioridad con la que el médico la ejecuta, son considerados como tareas primordiales los enlaces que permitan la atención al paciente, creación de citas nuevas, entre otros; estos se encuentren a simple vista de modo que se las pueda acceder de manera rápida.

Existe una zona, en donde se podrá visualizar directamente una lista de pacientes de un medico determinado, con el fin de que se pueda acceder de manera rápida a visualizar la información básica del paciente seleccionado. Esta zona también contará con una búsqueda según el apellido del paciente.

Otras tareas de menor relevancia, es decir que son efectuadas ocasionalmente se encontrarán organizadas como opciones secundarias dentro de los menús principales.

Como parte importante se considerara una zona activa de manera permanente, en la cual el profesional pueda visualizar los recordatorios que el sistema lo hace, como un resumen de las próximas citas a atenderse.

Además se considera la posibilidad de manejar una zona específica para la publicidad, con el fin de que el médico pueda revisar mediante estos enlaces información respecto a cada uno de ellos. Así también se dedicara una zona a los acontecimientos médicos suscitados en la actualidad para que el médico se informe sobre los mismos.

### **El usuario paciente**

Así como hemos considerado las actividades frecuentes del médico, se debe realizar lo propio con la siguiente categoría de usuario.

Estos usuarios estarán a cargo del manejo y administración de sus propias cuentas, de la asignación como revocación de permisos a médicos sobre tratamientos efectuados a dicho paciente.

El perfil del usuario paciente estará dividido en varios sectores agrupados según la prioridad de la información, en primer plano se podrá contemplar la información básica personal y de contacto del mismo, dado que estos datos son de vital importancia en caso de emergencia

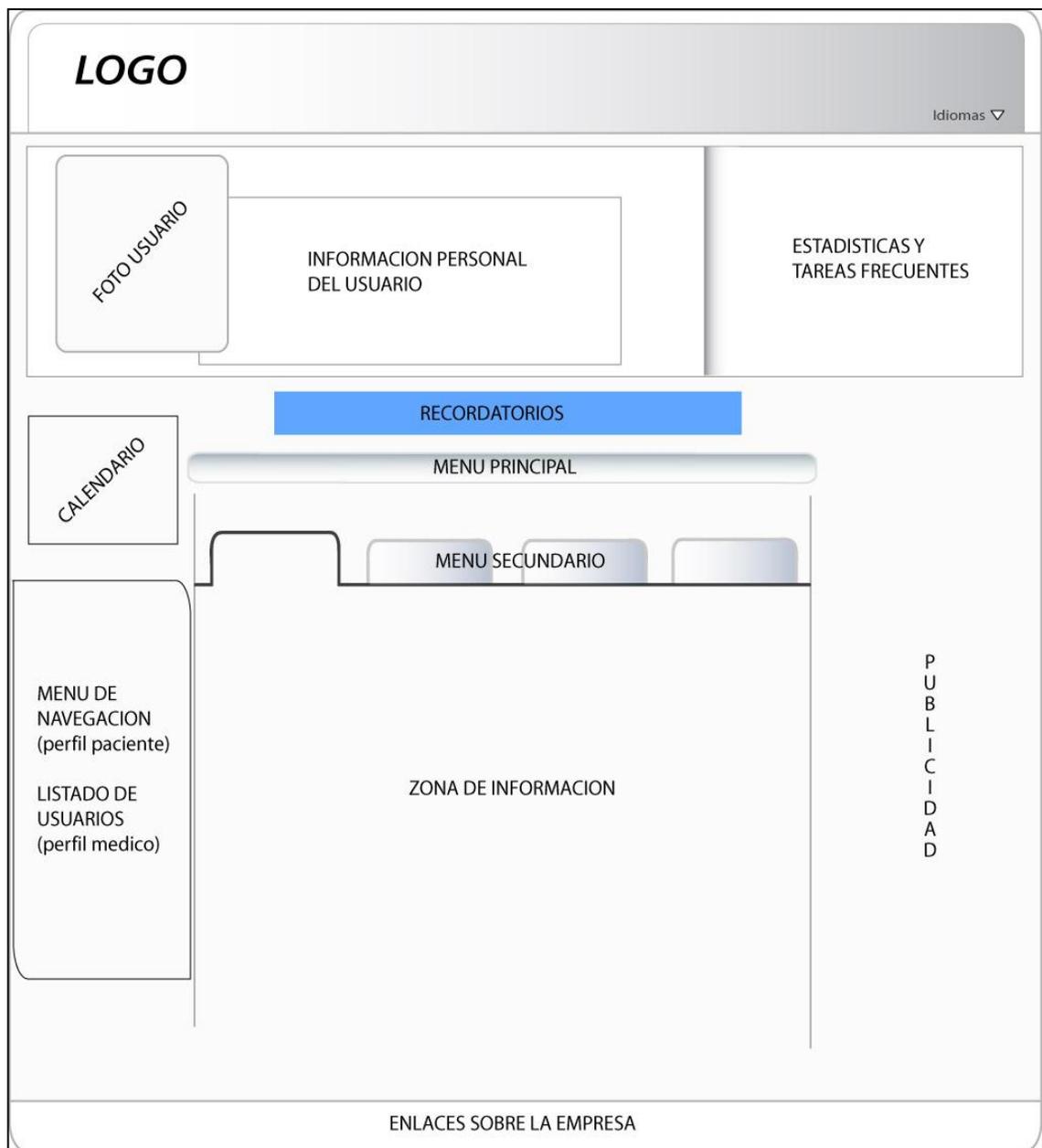
El paciente además dispondrá de un enlace que le recuerde de las actividades próximas a suscitarse, como por ejemplo el recordatorio de citas pendientes a las cuales deberá acudir con cierto médico.

El paciente al disponer de una ficha médica tendrá la posibilidad de administrar completamente tal información, esta será agrupada según tratamientos y médicos tratantes, adicional a esta información se contemplara antecedentes personales y familiares a los que se podrá adicionar manualmente enfermedades o tratamientos realizados por un medico que no disponga de este servicio.

Al igual que el médico se contará con menús secundarios y una zona de publicidad tanto de médicos, entidades de salud, entre otros.

## 2.- Realizar un prototipo refinado de la interfaz de usuario.

Fig. 5.1 Bosquejo de la Interfaz



### **3.- Organización de los objetos en la página.**

La página será estructurada de la siguiente manera.

En la parte superior derecha de la página se encontrara un conjunto de acciones que tienen que ver directamente con la cuenta, cerrar sesión y seguido de esto se encontrara una opción con la cual se podrá escoger el idioma de la interfaz.

Se dispondrá de un menú principal el cual contara con enlaces principales, cada uno de los cuales tendrán submenús que permitirán el acceso fácil a la información; y según la opción escogida en cada uno de los submenús, en la parte central de la pagina se presentara un menú dinámico para el manejo de la información.

En la parte izquierda de la página se dispondrá de publicidad, éste estará solo disponible en la cuenta de los pacientes. En la cuenta de los médicos, este menú será reemplazado por una búsqueda y un listado de pacientes.

También para evitar confusión por parte del usuario dentro de la página, se mostrará en una zona visible de la misma en donde se encuentra actualmente y en donde ha estado.

En el pie de página, se contara con enlaces propios de la empresa como: Acerca de la empresa, Contáctenos, Publicidad, Políticas de Privacidad, Términos de Uso y Ayuda.

### **4. Interfaz refinada por categoría de usuarios**

Una vez que hemos logrado identificar y distinguir cada una de las partes con las que contará la WebApp, se procedió a refinar los dos bosquejos anteriormente planteados; obteniendo de esta manera las diferentes pantallas principales mediante las cuales los usuarios harán uso del sistema.

**Fig. 5.2 Pantalla de Inicio de Sesión**

**FMed**

Español ▾

**¿ Nuevo en FMed...?**

- ☆ Almacena tu expediente médico
- ☆ Comparte con tus médicos tu información
- ☆ Completa privacidad en tu información

**Regístrate**

**Iniciar Sesión**

Nombre de Usuario

Contraseña

Recordarme

**Ingresar**

[¿ Olvidaste tu contraseña ?](#)

[Acerca de FMed](#) | [Contactenos](#) | [Publicidad](#) | [Politica de Privacidad](#) | [Terminos de Uso](#) | [Ayuda](#)

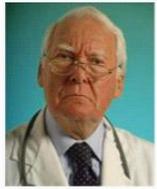
© 2009 FMed

Fig. 5.3 Interfaz para Médicos



[Salir](#) [Configuración](#) [Ayuda](#)

Español ▾



**Dr. Alberto Figueroa**  
**Cardiologo**  
 Especializado en la Universidad de Michigan - Estados Unidos  
 Cuenca - Ecuador

☎ 085746385    📞 2854767

- ✔ Atender Paciente
- ✔ Asignar Cita
- ✔ Crear Nuevo Paciente

Miercoles 13 de Mayo

Hora	Paciente	Lugar
10:00	Juan Perez	Santa Ines
10:30	Mario Tamayo	Santa Ines
11:00	Hernan Coronel	Santa Ines
16:00	Patricio Bernal	Monte Sinal

⚠ 5 Citas Pendientes    📧 1 Mensaje Nuevo

Mi Perfil
Mis Pacientes
Mis Citas
Mensajes
Buscar

**Buscar Pacientes**

- Abad Juan
- Bustamante Maria
- Carrion Esperanza
- Delgado Pablo
- Espinoza Paul
- Farfan Tomas
- Gonzales Carlos
- Hermida Raul
- Izquierdo Veronica
- Jimenez Paola
- Khan Martin
- Leon Santiago
- Lopez David
- Martinez Augusto
- Martinez Cesar
- Narvaez Marlene

**Citas**

Horarios de Atencion
Reportes

Clínica Santa Ana ▾

Buscar Citas

Dia | Semana | Mes

**Lunes 18 de Mayo de 2009**

	Lunes 18	Martes 19	Miércoles 20	Jueves 21	Viernes 22	Sábado 23
09:00						
09:30						
10:00						
10:30						
11:00						
11:30						
12:00						

+ Crear Cita

✖ Cancelar Cita

✎ Editar Cita



Monte Sinal  
Su alternativa en Cuenca - Ecuador



50 años al servicio de la medicina



Para la protección Cardiovascular

**Noticias**

**Gripe por A(H1N1) - Situación actual**  
 19 de mayo de 2009 -- A las 06.00 TMG del 19 de mayo de 2009 se habían notificado oficialmente 9830 casos de infección humana por virus gripal A (H1N1), incluyendo 79 decesos, en 40 países.

[Acerca de FMed](#) | [Contactenos](#) | [Publicidad](#) | [Politica de Privacidad](#) | [Terminos de Uso](#) | [Ayuda](#)  
 © 2009 FMed

**Fig. 5.4 Interfaz para Pacientes**



[Salir](#) [Configuración](#) [Ayuda](#)

Español ▾

---



**Juana Perez, 35 Años**  
**Abogada**  
 Sangurima 4-15 Entre Borrero y Luis Cordero  
 Cuenca - Ecuador  
 098473624 2855739

Tipo de Sangre: O+  
 Estatura: 160 cm.  
 Peso: 50 kg.  
 IMC: 20

Tratamiento Actual  
 Asma  
 Medicinas Pendientes  
 Ventolin

⚠ 1 Cita Pendiente ✉ 1 Mensaje Nuevo 👤 0 Nuevas Solicitudes



MI Ficha Mis Medicos Entidades de Salud Mensajes Buscar

Dr. Jaime Toral  
Neumologo Tratamiento Actual: ASMA

Ultima Visita
Receta
Examen Físico
Examen Complementario

Sintomas

El Paciente presenta tos frecuente, acompañada de abundante flema, además presenta dolor en la garganta. La tos se presenta con mas frecuencia en la noche.

Diagnostico

Asma: El paciente presenta un cuadro de asma leve, provocado posiblemente por la frecuente exposicion al smoke durante el dia.

Indicaciones

Uso constante de tapabocas mientras se encuentra en la zona de trabajo, además ingerir gran cantidad de agua, retirar todo tipo de alfombras y peluches de lana.

Contraindicaciones

Evitar el humo del cigarrillo, no estar en contacto con peluches, alfombras y animales. No ingerir colorantes.



Dr. Alfonso Merchan  
Neumologo



1800 - Fybeka  
Visitenos  
Cuenca - Ecuador



Vitaminas contra la Gripe

**MENU**

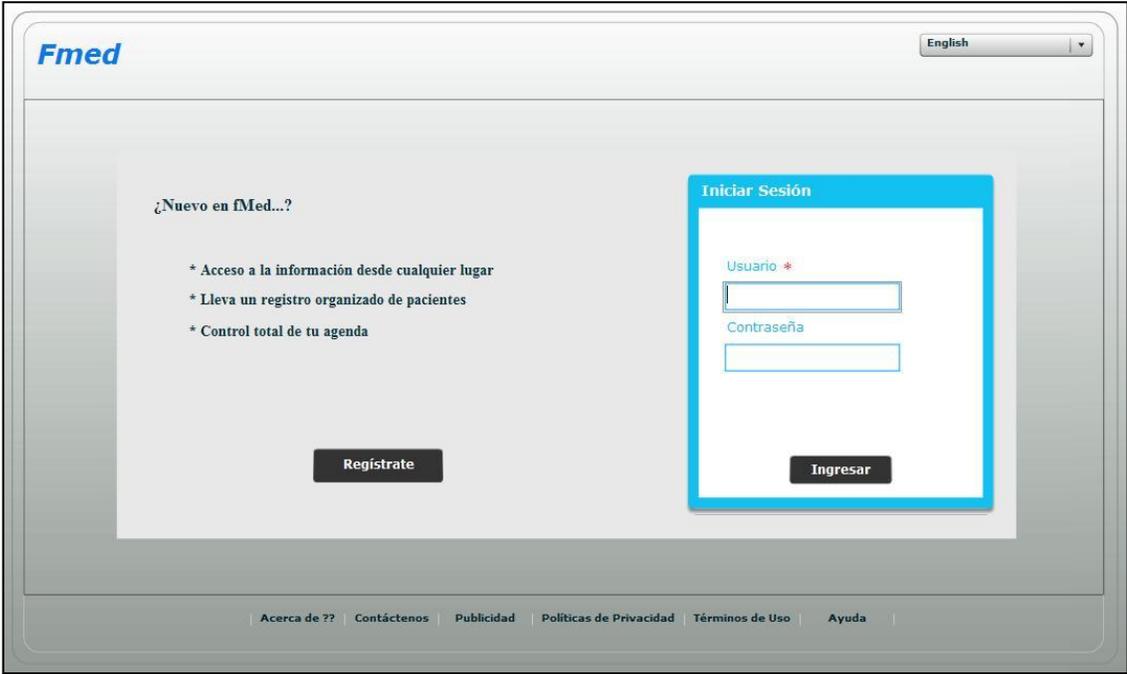
- Info. Personal Editar
- Antecedentes Personales Familiares
- Tratamientos Actuales Anteriores
- Permisos
- Noticias

Acerca de FMed | Contactenos | Publicidad | Política de Privacidad | Terminos de Uso | Ayuda

© 2009 FMed

5.3 Diseño estético

Fig. 5.5 Pantalla final inicio de sesión



**Fig. 5.6 Pantalla final registro**

Nombres: \*

Apellido Paterno: \*

Apellido Materno:

Mail/Nombre Usuario: \*

Contraseña: \*

*Mínimo 8 caracteres*

Repita la Contraseña: \*

Sexo: \* **Seleccione el género** ▼

Fecha de nacimiento: \* **Día** ▼ **Mes** ▼ **Año** ▼

País: \* **Seleccione el país** ▼

Ciudad: \* **Seleccione la ciudad** ▼

Tipo de Cuenta

Medico  Paciente

Al dar clic en el botón registrar acepta los términos y condiciones de uso impuestos por fmed!

**Registrarse**

Fig. 5.7 Pantalla final perfil medico

The screenshot shows the final medical profile page on the Fmed website. At the top left is the 'Fmed' logo. At the top right are links for 'Cerrar Sesión' and a language selection dropdown 'Choose your language'. The main header area is dark and contains a profile card for 'Dr. Jose Serrano', a specialist in Dermatology, with his contact information and a 'Sacar Cita' button. To the right of the profile card, the date and time '15 - 03 - 2010 18:26:44' are displayed, along with buttons for 'Atender Paciente' and 'Sacar Cita'. Below the header, a blue bar indicates '0 Cita(s) Pendiente(s)'. A navigation menu includes 'Mi Cuenta', 'Mis Citas', 'Mis Registros', and 'Buscar'. On the left, a sidebar lists other users: 'Pulla Cristian', 'Serrano Veronica', and 'Serrano Johanna'. The central content area is titled 'Resumen...' and lists 'Ultimo Paciente Atendido: Johanna Serrano', 'Siguiete Paciente por Atender:', 'Lugar(es) de Trabajo de Hoy:', and 'Lugar(es) de Trabajo de Mañana:'. Three vertical black banners with the text 'tu publicidad aquí' are positioned on the right and bottom sides of the page. The footer contains a series of links: 'Acerca de ??', 'Contáctenos', 'Publicidad', 'Políticas de Privacidad', 'Términos de Uso', and 'Ayuda'.

Fig. 5.8 Pantalla final perfil paciente

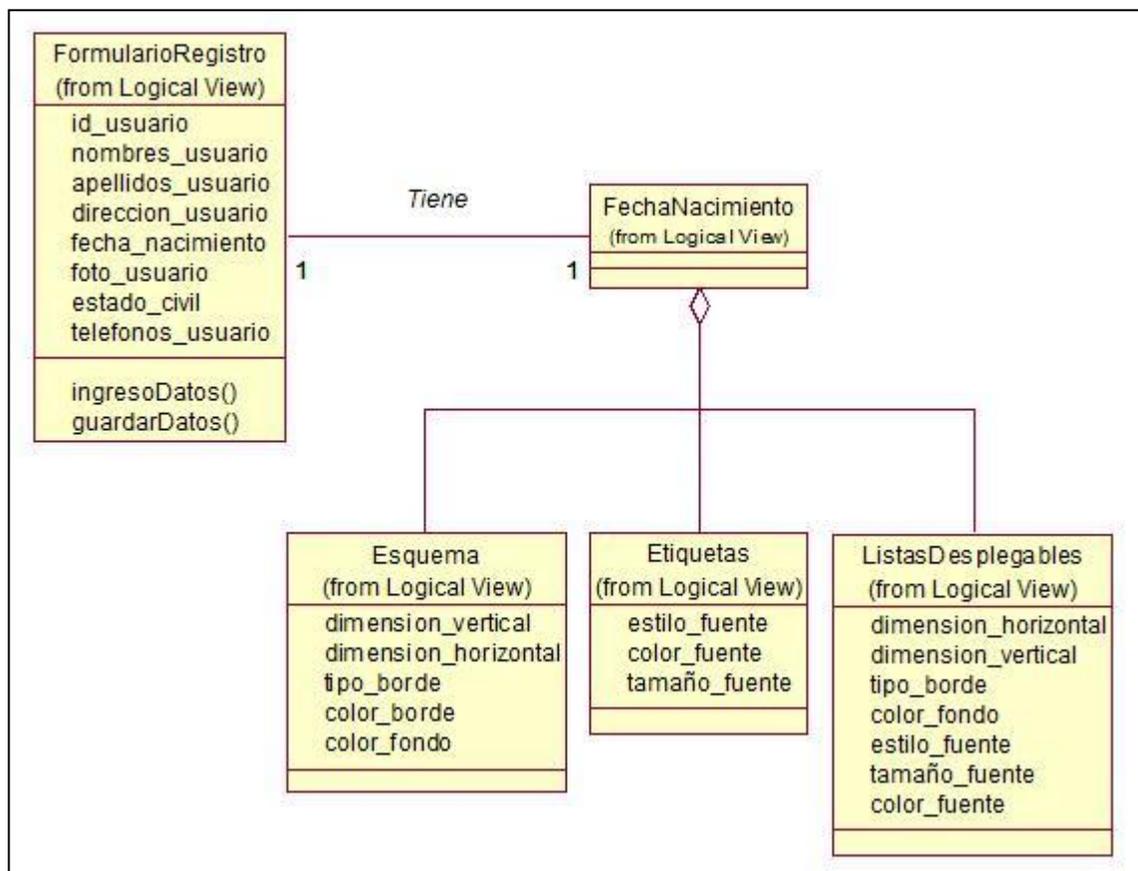
The screenshot displays the 'Fmed' patient profile interface. At the top left is the 'Fmed' logo, and at the top right are links for 'Cerrar Sesión' and 'Choose your language'. The main profile area is divided into two columns. The left column features a placeholder for a profile picture and the patient's name 'Johanna Serrano, 28 años', along with her profession 'Arquitecta', location 'Camilo Egas, Cuenca-Ecuador', and phone numbers '085821439 / 2804370'. The right column lists medical details: 'Numero de emergencia: Esposo: 089345721', 'Tipo de Sangre: O+', 'Estatura: 1,65 mts.', 'Peso: 60 Kg.', and 'IMC: 22,04'. Below this is a blue button indicating '0 Cita(s) Pendiente(s)'. A navigation bar contains links for 'Mi Ficha', 'Mis Médicos', 'Mi Cuenta', 'Permisos', and 'Buscar'. The central focus is a 'Ultima Cita' (Last Appointment) window with the following details: 'Fecha: 2010-03-11', 'Tratamiento: Remocion de Verrugas', 'Medico tratante: Dr. Jose Serrano', and 'Atendido en: Jose Serrano'. The page is flanked by vertical black bars with the text 'tu publicidad aquí' and a footer with links for 'Acerca de ??', 'Contáctenos', 'Publicidad', 'Políticas de Privacidad', 'Términos de Uso', and 'Ayuda'.

## 5.4 Diseño de contenido

En esta parte del diseño de contenido del sistema, se procede a detallar cada uno de los objetos de contenido definidos anteriormente en el análisis del sistema, esto consiste en especificar y definir cada uno de sus atributos y métodos que serán necesarios antes de la implementación.

Además para cada atributo de los objetos de contenido, se procede a identificar mediante subclases la información específica con la que contara cada atributo, esto mediante la descripción detallada de atributos que serán necesarios para el diseño, lo cual es muy importante ya que mediante ellos se podrá entregar información a los usuarios.

**Fig. 5.9** Diseño de contenido formulario de registro



**Fig. 5.10 Diseño de contenido publicidad**

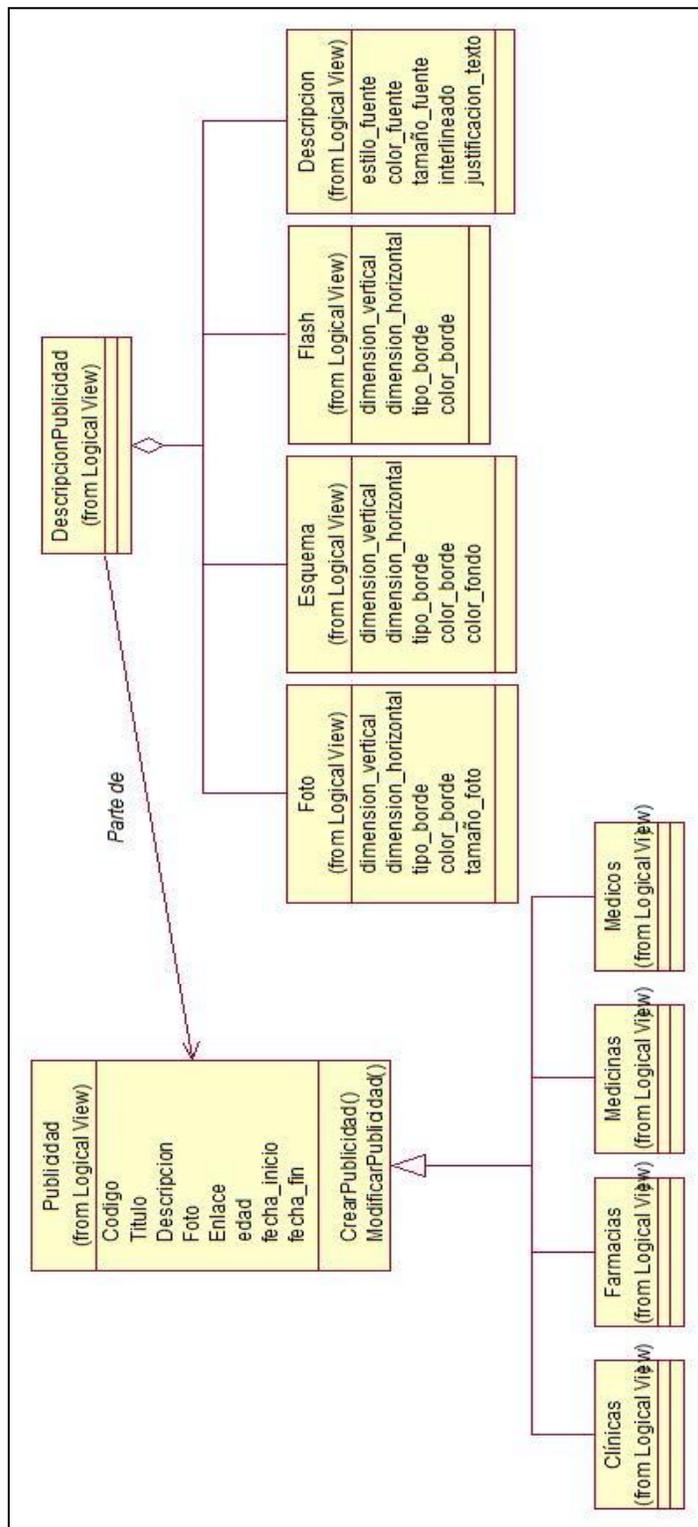
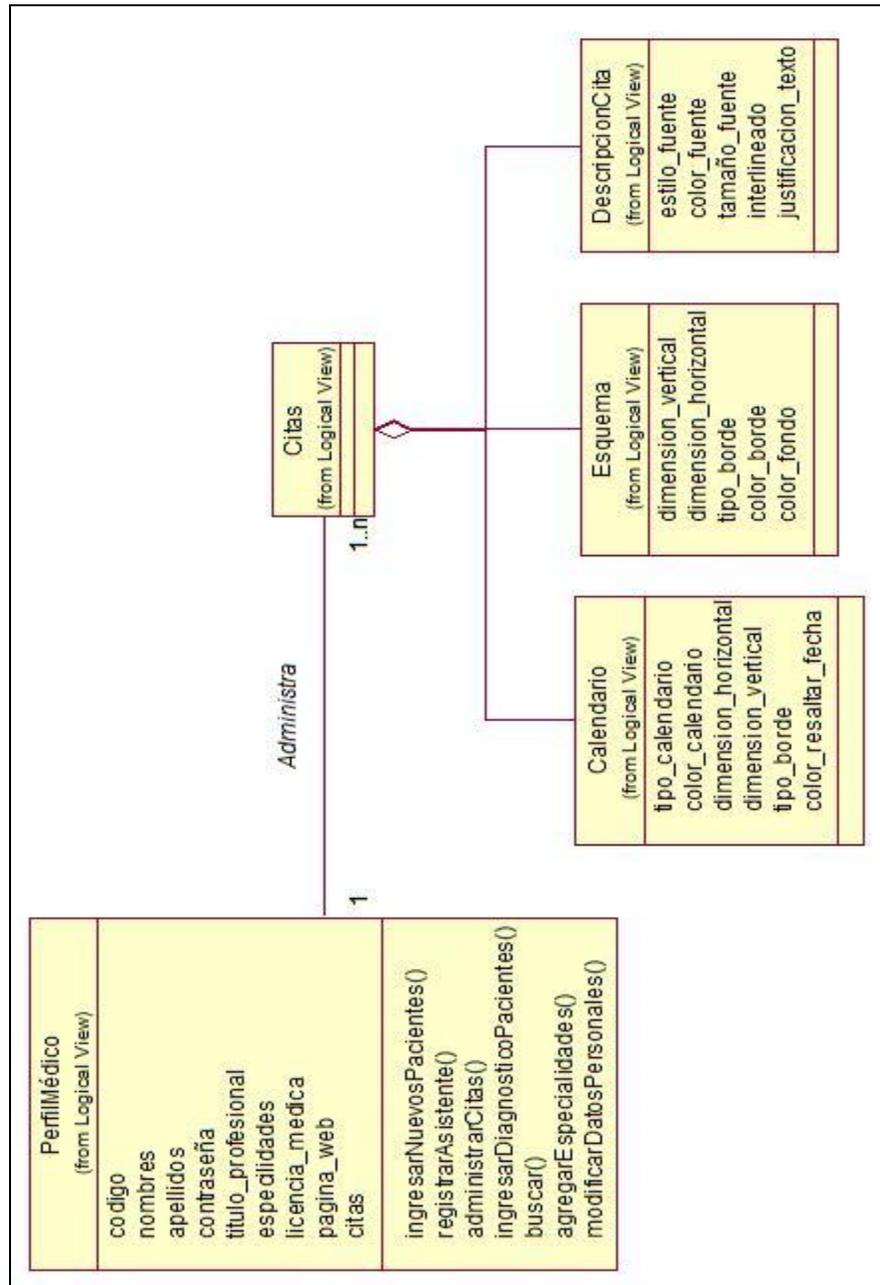


Fig. 5.11 Diseño de contenido administración de citas



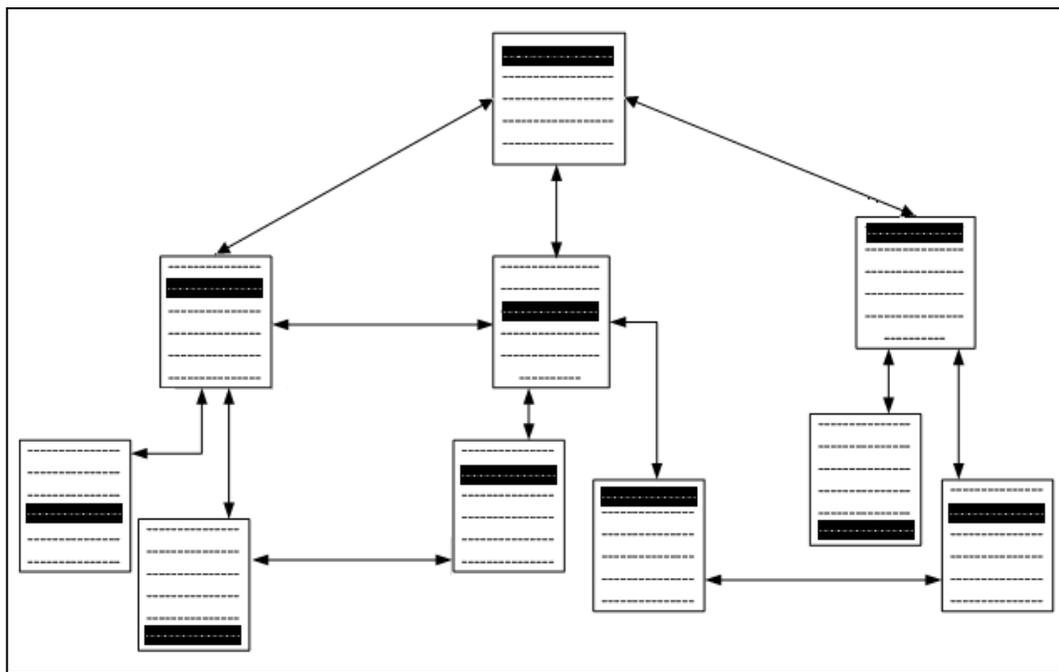
## 5.5 Diseño arquitectónico

La estructura que se utilizará para el diseño arquitectónico de la WebApp será la Estructura Jerárquica, debido a que la funcionalidad de la gran mayoría de las páginas que se crearán y con las que el usuario interactuará, dependerá específicamente de todos los requisitos solicitados en las páginas anteriores. Además se tendrá la posibilidad de navegar horizontalmente mediante enlaces de hipertexto, de esta manera la navegación dentro de la pagina será de forma más fácil y ágil.

### Estructura jerárquica

Este tipo de jerarquías distribuyen su flujo de control verticalmente mediante las ramificaciones; también mediante enlaces de hipertexto, se puede navegar horizontalmente dentro de las páginas.

Fig. 5.12 Estructura Jerárquica



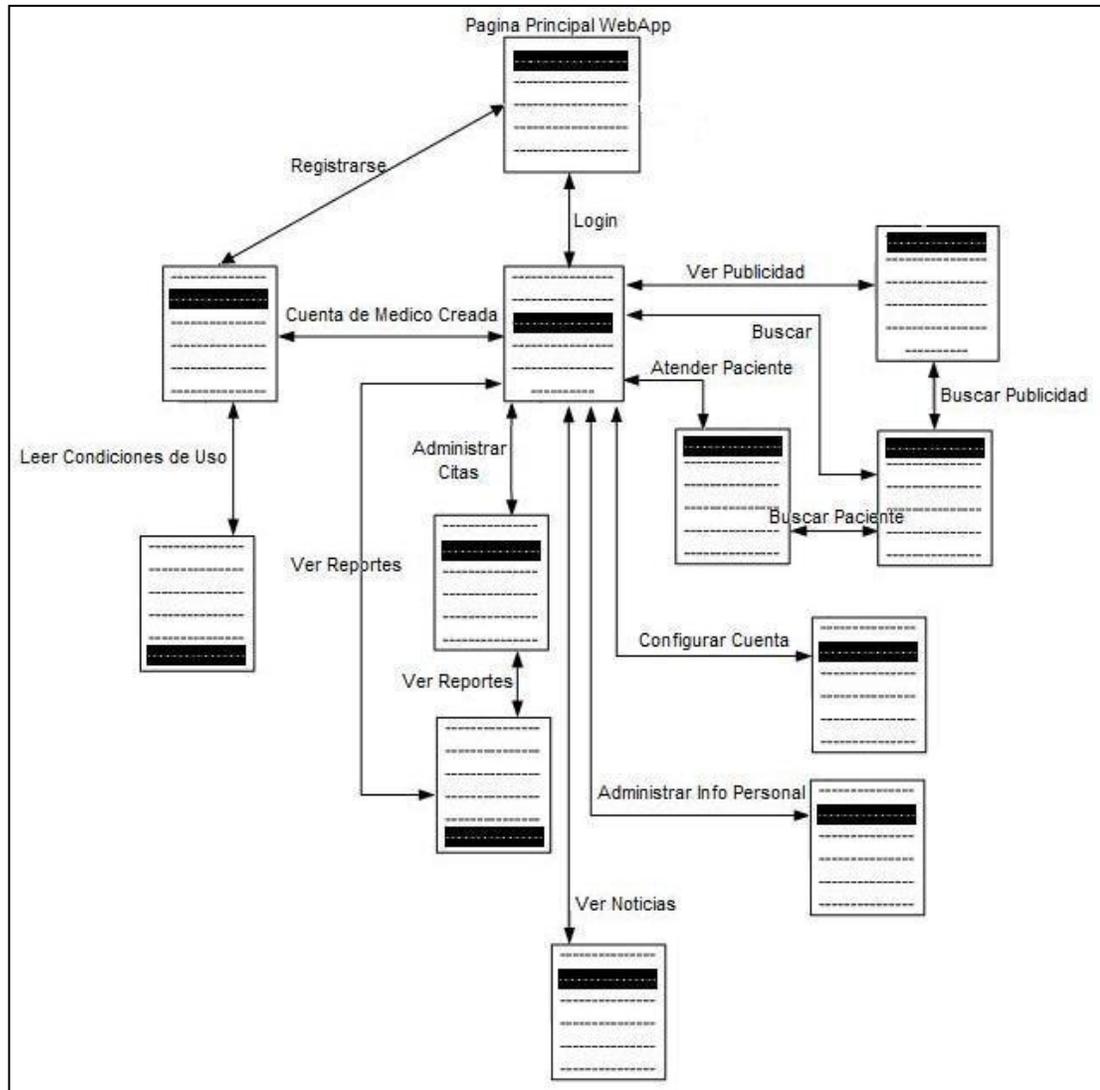
## **Estructura jerárquica de la WebApp**

Una vez registrado y creada la cuenta respectiva, se podrá navegar por la página mediante los vínculos y enlaces que se encuentran tanto en los menús horizontales y verticales, además se puede acceder a otros lugares mediante los enlaces de texto e imagen.

### **Perfil médico**

En cuanto se refiere al perfil del Medico, desde la página principal de su cuenta, se puede dirigir a la administración de citas, en donde podemos ver el calendario de las citas del día de hoy, se puede ver reportes de las citas por atender; se puede buscar pacientes y una vez escogido el mismo se puede escribir dentro de la ficha clínica del paciente; también se puede buscar medicamentos, farmacias, laboratorios, entre otros. Se puede configurar la cuenta y modificar la información personal.

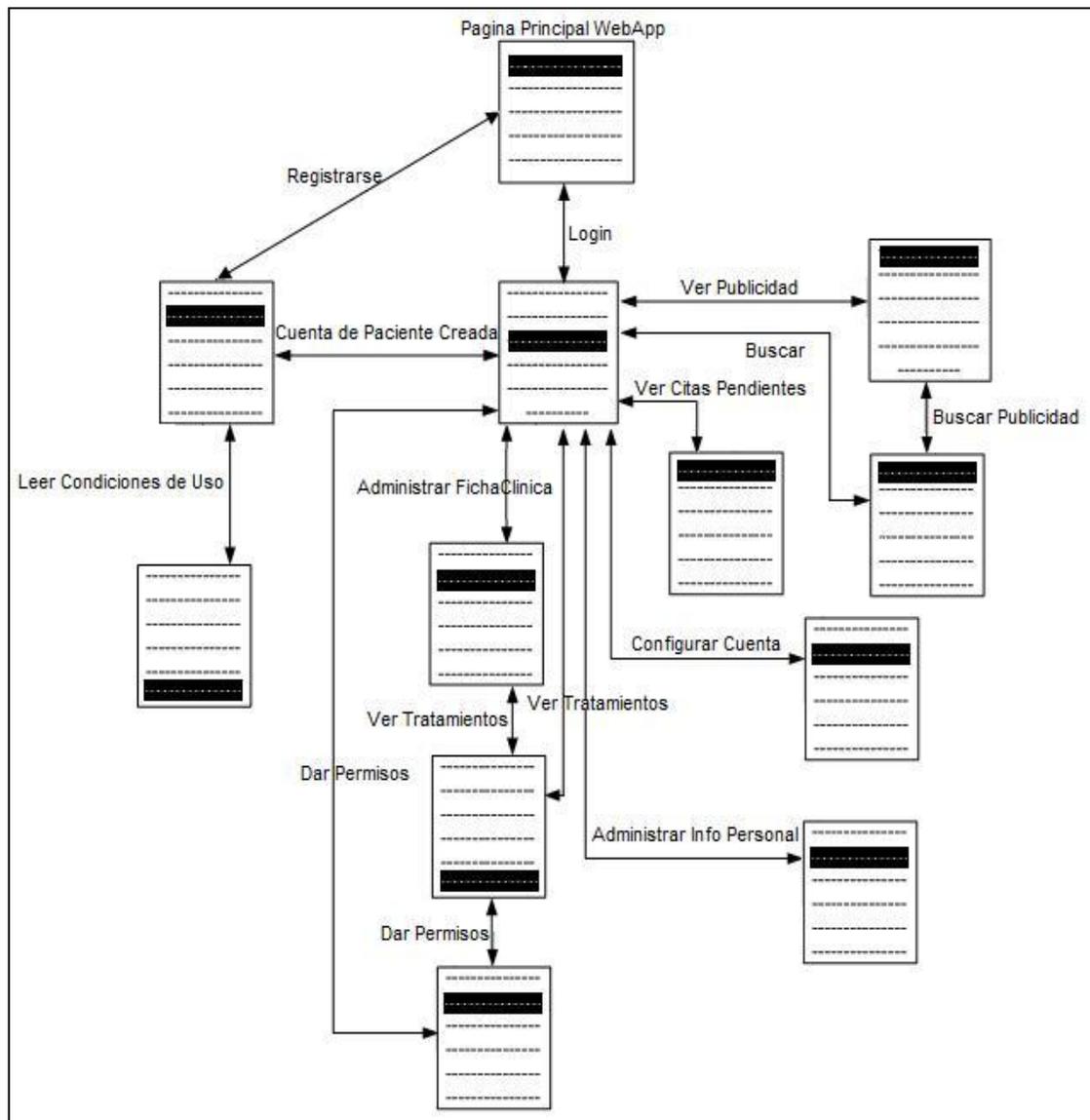
**Fig. 5.13 Estructura jerárquica cuenta medico**



### **Perfil paciente**

En el perfil del Paciente, desde la página principal de su cuenta, se puede administrar su ficha clínica, en donde se puede ver los tratamientos realizados y el actual; a éstos se les puede asignar permisos para que se pueda o no visualizar dicha información. Se puede ver las citas pendientes que pueda tener en un futuro cercano; se puede buscar entidades de salud, médicos, entre otros. Se puede configurar la cuenta y modificar la información personal.

**Fig. 5.14 Estructura jerárquica cuenta paciente**



### 5.6 Arquitectura utilizada para la aplicación

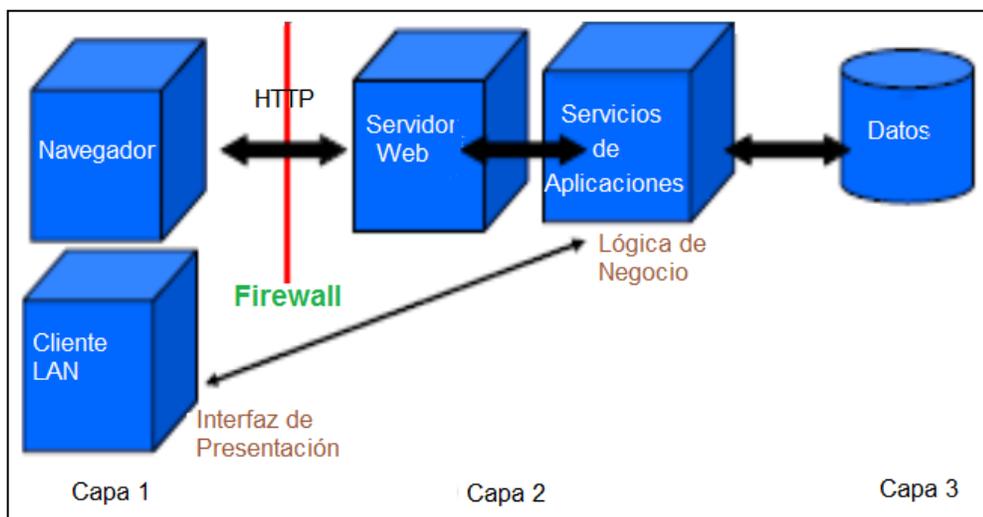
La arquitectura escogida es la multicapa, la cual permite dividir la aplicación en varios niveles de componentes, es decir se puede tener por separado la interfaz del usuario, los datos, y las funciones y procedimientos que ejecutan ciertas tareas dentro de la aplicación. La integridad de cada capa se encuentra independiente una de la otra.

**Capa de presentación.-** Capa visualizada por el usuario, es decir la interfaz gráfica, en donde el usuario puede ver, capturar y enviar información, esta debe ser sencilla y fácil de usar. Esta se comunica con el servidor web y posteriormente con la capa de negocios.

**Capa de negocio.-** Esta capa captura las acciones del usuario, las ejecuta y envía su resultado a la capa de presentación. Puede comunicarse con la capa de datos para el ingreso, consulta o modificaciones dentro de la base de datos.

**Capa de datos.-** Capa que se encarga de almacenar y acceder a los datos mediante el uso de gestores de base de datos. Recibe de la capa de negocios las peticiones de almacenamiento o recuperación de los datos.

**Fig. 5.15 Arquitectura multicapa**



El modelo utilizado es el MVC (Modelo-Vista-Contrólador) el cual se describirá a continuación.

### 5.6.1 El modelo MVC

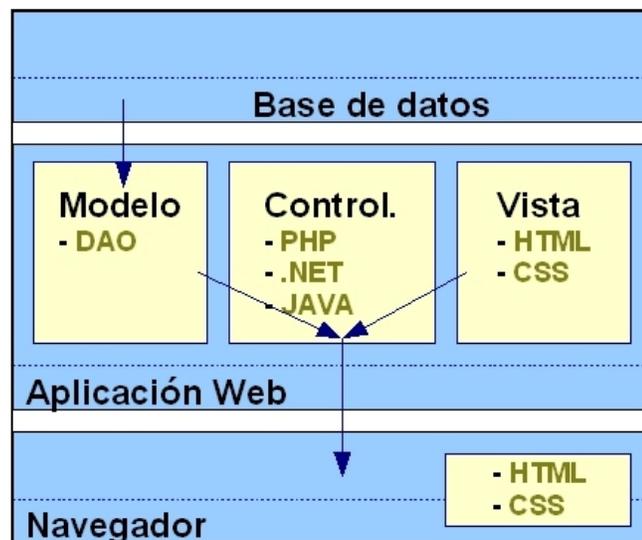
Es una infraestructura que permite mantener desacoplados la interfaz, la información y la funcionalidad de la aplicación.

El modelo contiene todo lo relacionado con el contenido de la aplicación y la funcionalidad del sistema, además es el responsable de acceder al almacenamiento de los datos.

La vista obtiene los datos del modelo, y muestra toda la información del modelo de contenido en la interfaz de usuario.

El controlador gestiona el acceso y coordina el flujo de datos entre el modelo y la vista, también es el encargado de recibir todos los eventos de entrada realizados por el usuario.

**Fig. 5.16 Modelo MVC**

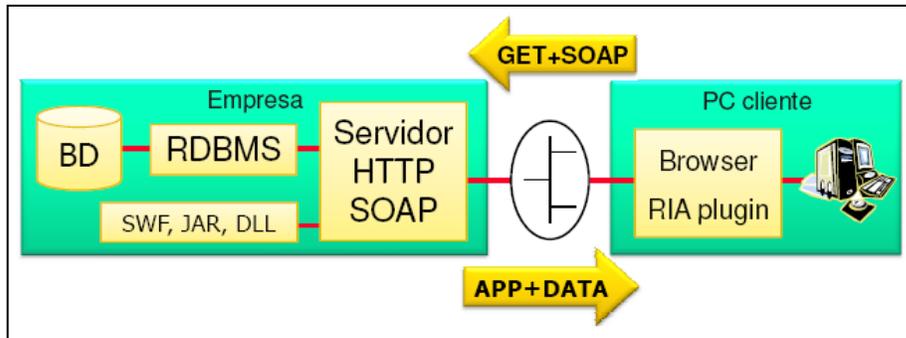


La arquitectura en la que se desarrolló Flex, se encuentra implementada sobre aplicaciones Web J2EE básicas; comprende una tecnología tipo cliente, en donde algunas de las tareas sencillas como filtrados, agrupaciones, validaciones de campos, etc., son ejecutados realmente en el lado del cliente, dejando solamente al servidor para la obtención de datos.

La parte de interfaz de usuario maneja un lenguaje MXML (Multimedia eXtensible Markup Language), que una vez compilado en formato binario en la parte del servidor crea un archivo de tipo SWF. En el momento que un usuario entra a una aplicación de

este tipo mediante la web, el archivo SWF se traslada al sistema cliente en donde comienza a ejecutarse.

**Fig. 5.17 Arquitectura web RIA browser**



## 5.7 Diseño de navegación

En esta parte se definen los trayectos o rutas mediante las cuales los usuarios podrán acceder y navegar por las diferentes funciones de la WebApp, esto se lo realiza con el fin de que el usuario pueda desplazarse de una forma intuitiva y sencilla mediante la página, y de esta manera no hacer complicada su navegación.

Esta parte es muy importante tenerla en cuenta, debido a que si no se realiza bien la forma en cómo se navegará dentro de la página, los usuarios tendrán dificultades y de seguro abandonarán el uso de la misma.

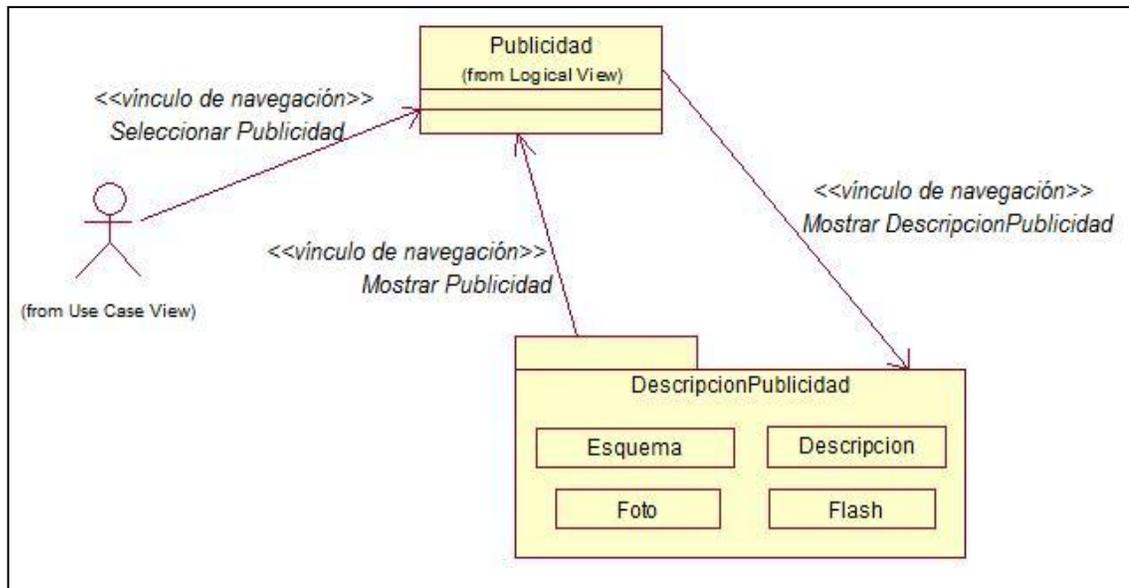
Esto se lo debe hacer mediante la identificación de dos partes, la semántica y la sintaxis de navegación.

### Semántica de navegación

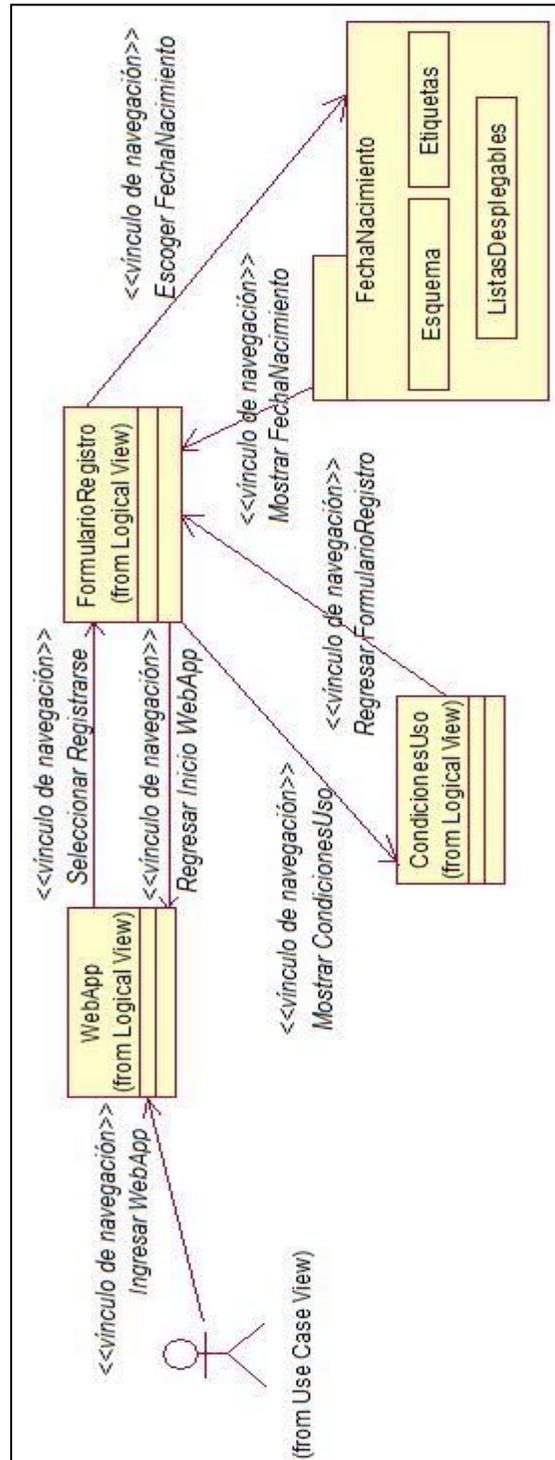
La semántica de navegación consiste en el sentido de flujo por donde navega el usuario dentro de la WebApp; este flujo de navegación es diferente para cada tipo de usuario

debido a que cada uno de ellos realiza funciones diferentes; además, la información que se presenta durante la navegación y que se obtiene de ella debe ser consistente.

**Fig. 5.18 Semántica navegación publicidad**



**Fig. 5.19 Semántica de navegación formulario de registro**



**Fig. 5.20 Semántica de navegación ficha clínica**

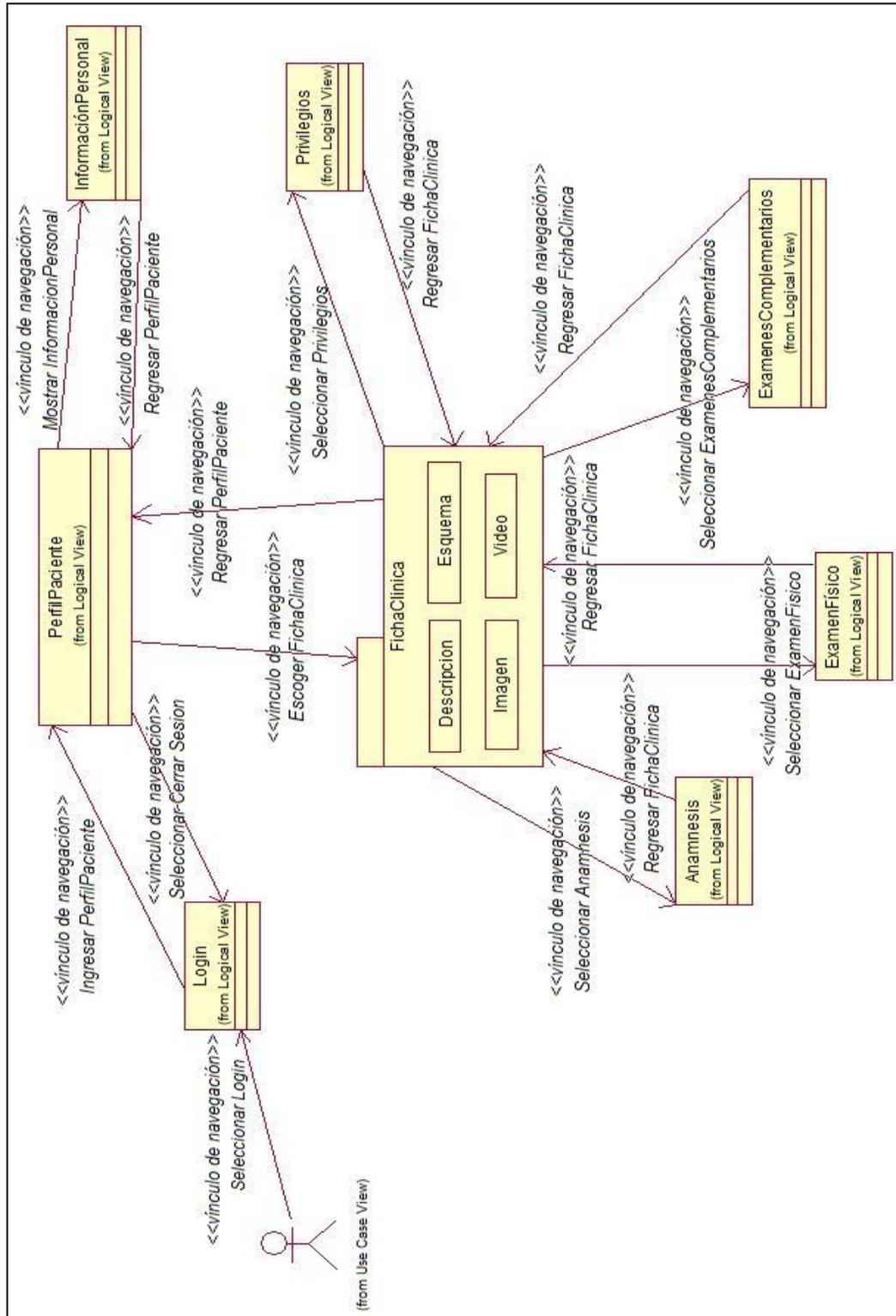
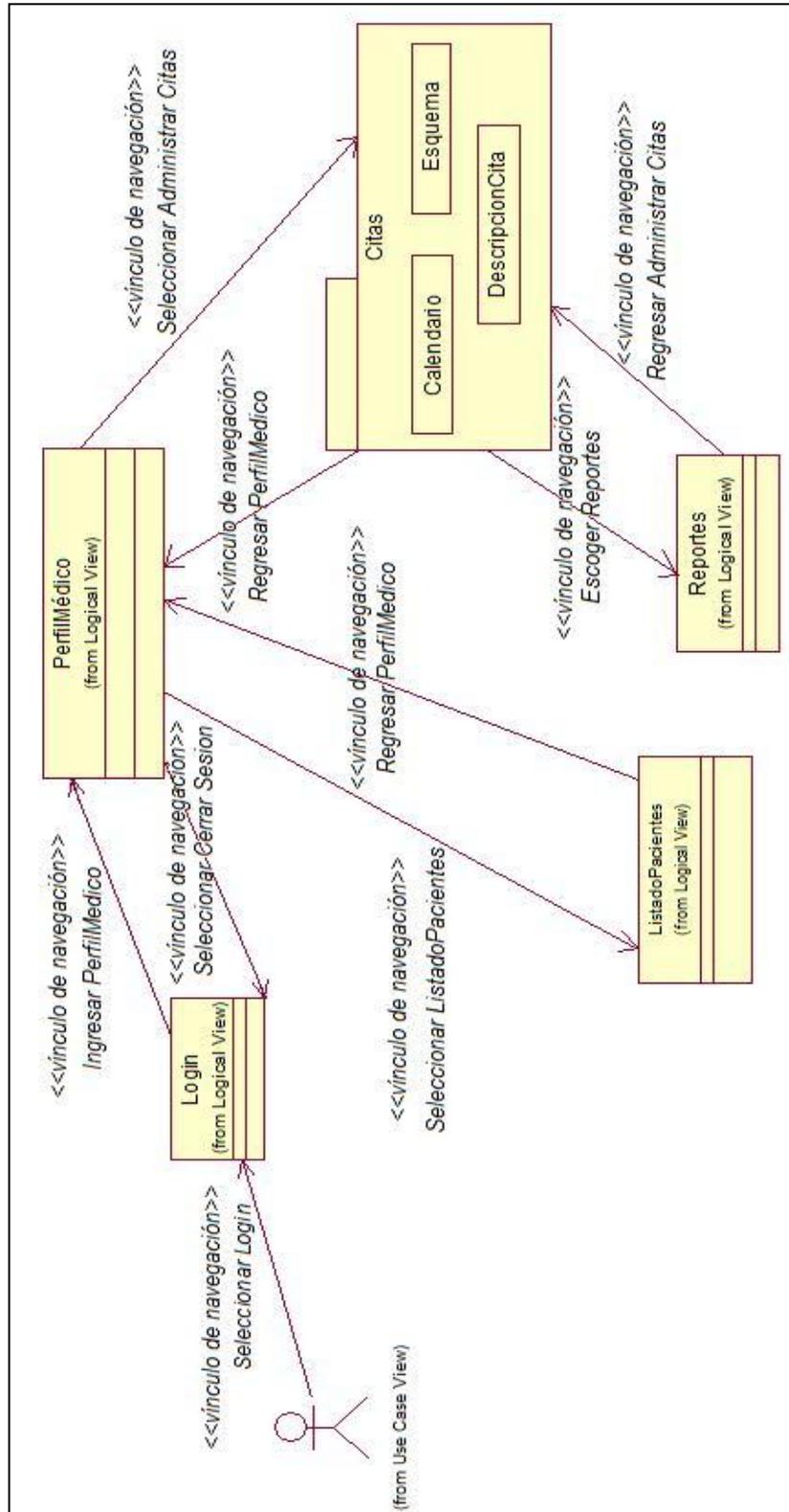


Fig. 5.21 Semántica de navegación citas



## **Sintaxis de navegación**

La sintaxis de navegación consiste en identificar cada uno de los mecanismos apropiados mediante los cuales se podrá navegar dentro de la WebApp, es decir, se definirán un conjunto de estándares que serán utilizados para poder moverse entre los diferentes objetos de contenido y de análisis. Entre estos como ejemplo podemos tener menús, iconos, pestañas, etc.

Para nuestra WebApp, se contara con la siguiente sintaxis de navegación:

- Vínculos de texto
- Vínculos gráficos
- Botones
- Menú horizontal
- Iconos gráficos de búsqueda
- Pestañas
- Calendarios
- Listas
- Acordeones

## **5.8 Conclusión**

En el presente capítulo se ha podido refinar más detalladamente cada uno de los objetos de contenido obtenidos anteriormente en el análisis del sistema, con lo cual hemos logrado definir cada uno de los atributos y métodos con los que contará cada objeto, además se estableció las características de diseño propias de cada atributo, lo cual ayudará mucho antes de la implementación; de igual manera, se establecieron las USN para cada categoría de usuario y se especificó la sintaxis de navegación.

Por otro lado, se estableció la estructura de la página web, su aspecto visual y el contenido de cada una de las páginas; además mediante la ayuda de un programa de diseño gráfico, se logró realizar varios bosquejos de cada una de las páginas con las que contara cada categoría de usuario.

## **CAPÍTULO 6**

### **CODIFICACION Y PRUEBAS DE LA APLICACION**

#### **6.1 Introducción**

Una vez que todo el proceso de generación de software está concluido no queda más que empezar a desarrollarlo, para lo cual se han analizado también las tecnologías y herramientas disponibles en el mercado que nos puedan ayudar a construir la aplicación planeada.

Además a medida que el proyecto empieza a tomar forma se definirán el tipo de pruebas a ejecutarse paralelamente con el desarrollo de la aplicación. Se darán a conocer las pruebas vitales que deben realizarse desde el inicio en todo proyecto informático, con el fin de corregir los errores encontrados, mejorar la aplicación y poder lanzarla sin ningún inconveniente.

## **6.2 Herramientas disponibles**

Ante la necesidad del rápido desarrollo de las aplicaciones web, nacen las aplicaciones enriquecidas de internet (RIA), que permiten el desarrollo de lo que antes era impensable con tecnologías tradicionales basadas en HTML, las cuales necesitaban ser recargadas a cada momento que se ejecute una acción.

Este movimiento tiene grandes participantes como lo es Flex que lidera la carrera, seguido de otras alternativas no menos importantes como Silverlight de Microsoft, JavaFx entre otros.

Al liderar Flex la batalla, nos decidimos por este además de las características adicionales que nos proporciona, tales como la carencia de una licencia para el servidor como Microsoft por ejemplo que las exige. Una de las principales características de Flex es que se compila por medio de su lenguaje propio de marcas (MXML) y se ejecutan mediante ActionScript aplicaciones Flash (SWF), que estas a su vez son importadas por medio de código HTML para de este modo resultar en páginas web con aplicaciones Flash embebidas.

Al realizar esta tarea Flex, elimina muchas de las preocupaciones que a los desarrolladores actualmente se les presenta como un riesgo, que es la incompatibilidad frente a ciertos navegadores de internet, al Flex ejecutar una aplicación Flash estamos hablando de que la aplicación va a ser compatible y su funcionalidad va a ser de la misma manera planeada en la gran mayoría de navegadores y expandiendo los límites a la telefonía celular que empieza a incorporar la compatibilidad de aplicaciones flash en sus navegadores móviles.

Además como característica adicional, Flex nos ayuda de gran manera a mejorar la apariencia de nuestras aplicaciones al incorporar varios componentes y características que aportan funcionalidades tales como objetos remotos, arrastrar y soltar, columnas ordenables, gráficas, efectos de animación, etc. Además la aplicación se carga una sola vez al inicio, reduciendo el flujo de datos frente a las aplicaciones tradicionales.

### 6.2.1 Adobe Flex 3

“Flex es un marco de trabajo de código abierto gratuito altamente productivo para la creación y el mantenimiento de aplicaciones web expresivas que se implantan coherentemente en los principales exploradores, equipos de escritorio y sistemas operativos.” (Adobe)

La aplicación Flex Builder 3 dispone de una versión de prueba de 30 días que presenta ciertas limitaciones al ser gratuita, pero la versión profesional se la puede adquirir en la tienda en línea de adobe en sus dos presentaciones: la estándar y la profesional que es la más completa y por ende las más costosa.

**Fig. 6.1 Adobe Flex Builder 3**



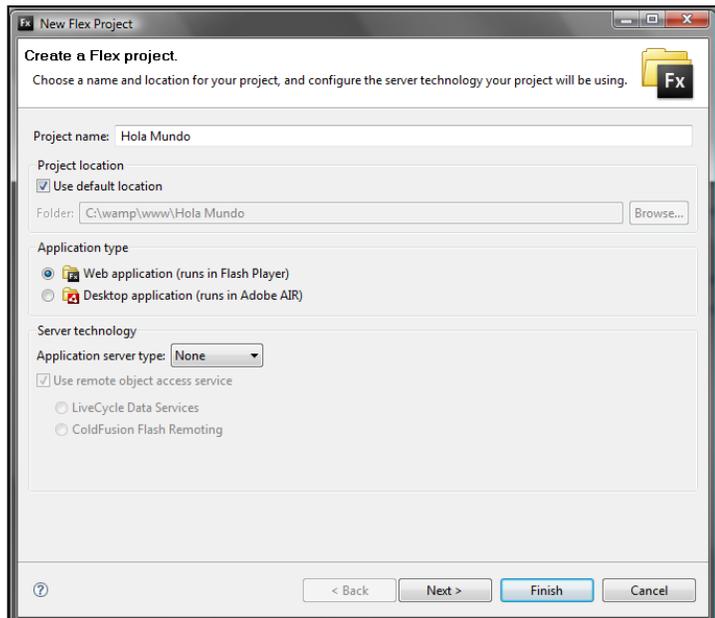
Brevemente explicaremos el uso de la herramienta mediante un ejemplo simple, en el cual trataremos de dar una idea general de la forma en la que Flex Builder nos permite crear aplicaciones enriquecidas para internet.

#### **Ejemplo de Uso**

Creación de un proyecto.

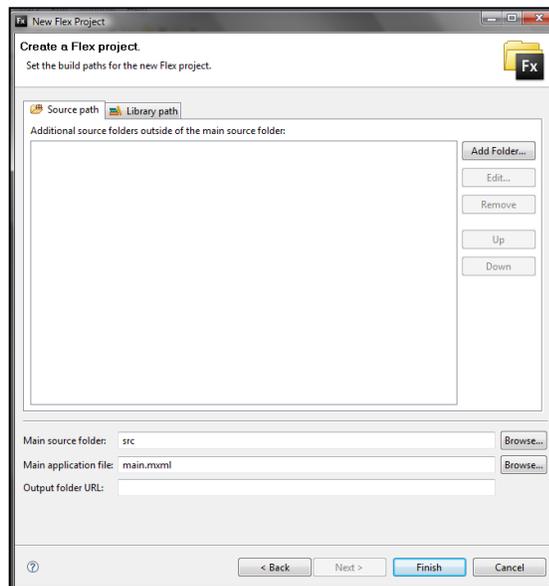
Antes de empezar necesitamos crear un proyecto en donde se van a almacenar todos nuestros módulos y clases para dicha aplicación.

**Fig. 6.2 Pantalla nuevo proyecto**



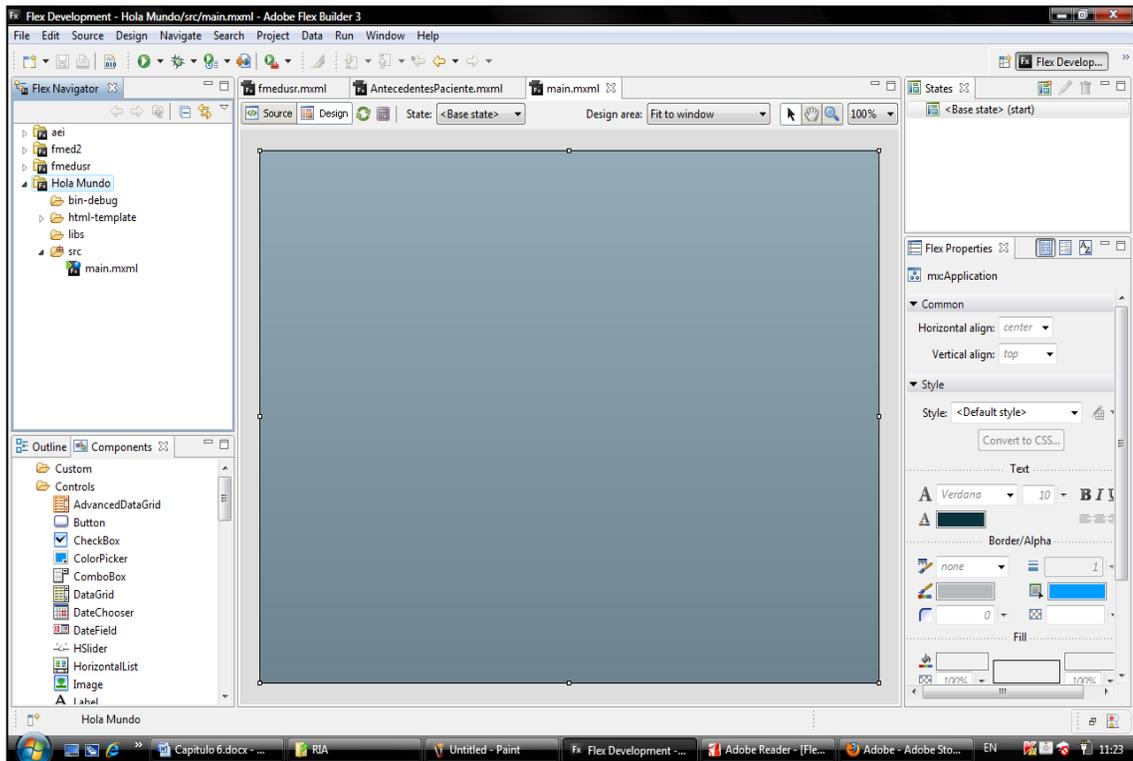
Para lo cual necesitamos asignarle un nombre al proyecto, el lugar en donde se encuentra alojado nuestro servidor de pruebas y adicionalmente podemos seleccionar un tipo de servidor que vayamos a usar. Una vez completados todos los datos procedemos a dar clic en el botón Next o Siguiente

**Fig. 6.3 Pantalla configuración proyecto**



En caso de ser necesaria la importación de ciertos paquetes, módulos, librerías o clases prefabricadas; en esta ventana podemos realizar la importación y luego damos clic en finalizar.

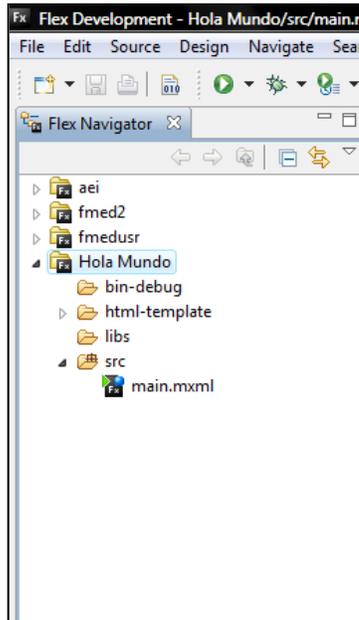
**Fig. 6.4** Marco de trabajo



Aquí podemos ver el marco de trabajo de Flex que nos aparece al momento de terminar de crear el proyecto.

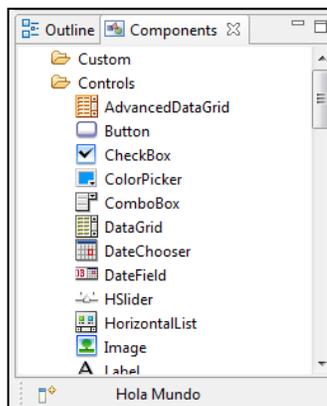
En el tab Flex Navigator podemos observar los ficheros que componen nuestro proyecto indicándonos que el archivo main.mxml es el archivo principal del proyecto.

**Fig. 6.5 Tab de navegación**



En el tab de componentes podemos observar todos los objetos disponibles para arrastrarlos a la pantalla principal y empezar a hacer uso de ellos.

**Fig. 6.6 Tab de componentes**



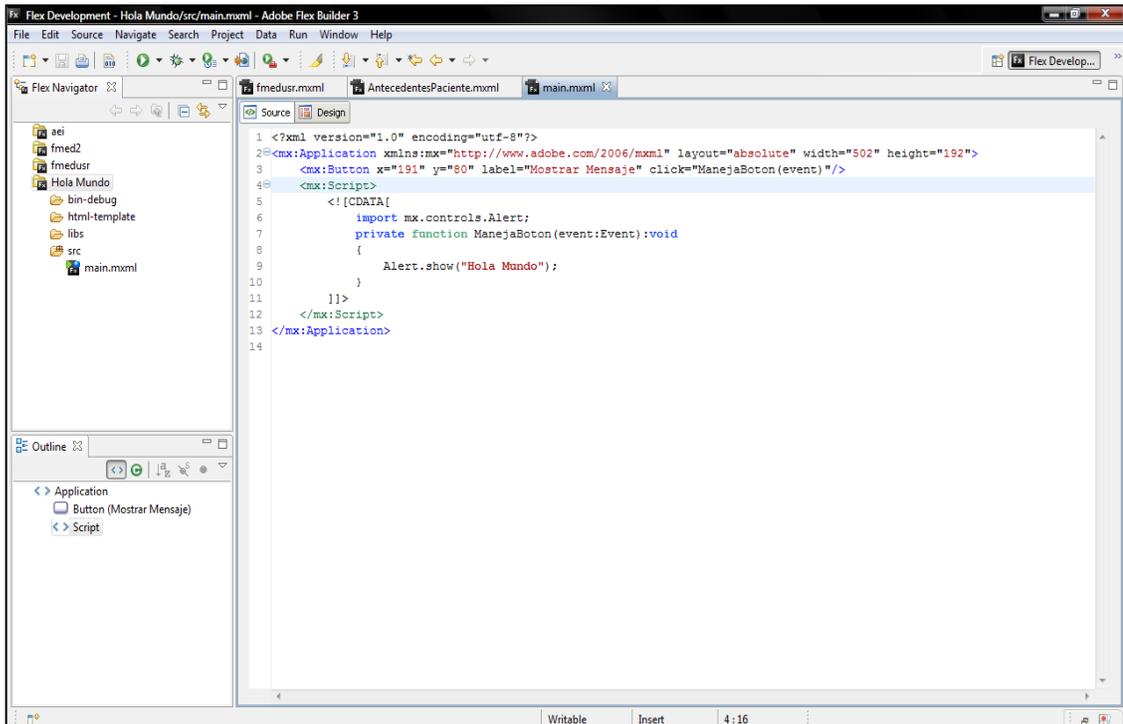
Agregamos un botón que será el encargado de ejecutar una función que muestre el mensaje de “Hola Mundo”

**Fig. 6.7 Inserción de objetos**



En modo de código creamos la función ManejaBoton() que será la encargada de lanzar la ventana con el mensaje deseado, además agregamos al evento Click del botón la función deseada.

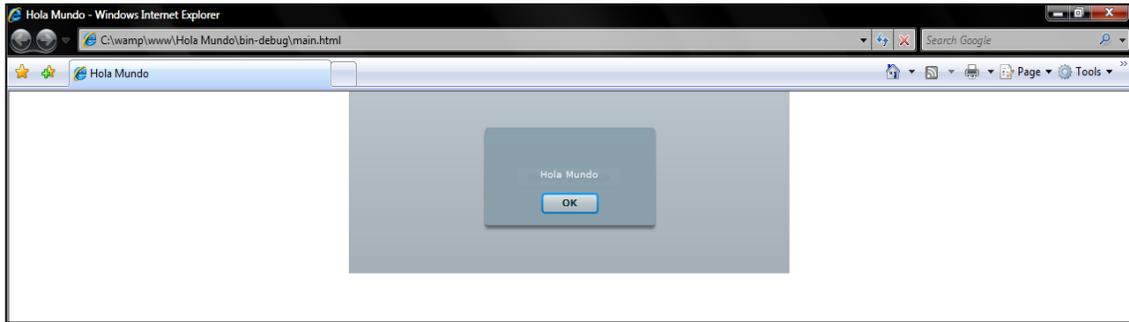
**Fig. 6.8 Código fuente AS**



Y con eso terminamos de realizar el ejemplo básico del uso de la plataforma Flex Builder 3

Aquí podemos observar el mensaje que se muestra al momento de dar Click en el botón mostrar mensaje, se dispara el evento Click del botón y llama a una instancia de la clase Alert por medio del método Show y nos exhibe en la pantalla nuestro mensaje.

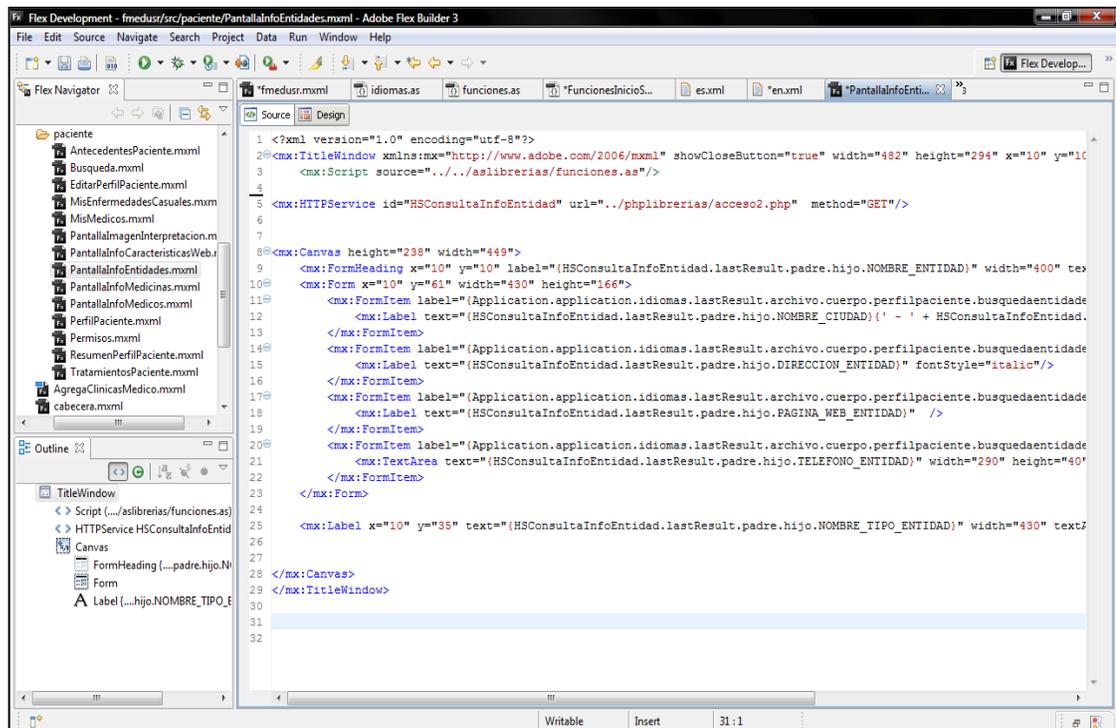
**Fig. 6.9 Ejecución del ejemplo**



### Uso de Flex para el desarrollo de la aplicación

Una vez que la herramienta usada se ha explicado y explorado brevemente, procedemos o insertar scripts de código que nos muestran el uso de Flex concretamente para la aplicación desarrollada.

**Fig. 6.10 Creación de objetos para pantalla entidades**



En la imagen podemos observar en modo de código fuente las líneas necesarias para crear una pantalla que muestra la información de una entidad de salud en especial, cuando ha sido invocada esta forma.

Podemos observar que el código se encuentra formado en su parte principal por un objeto `HttpService` que es el encargado de traer los datos desde la base de datos por medio de PHP para mostrarlos en la pantalla, además tenemos un `Canvas` o un contenedor que es el encargado de agrupar a los demás objetos dentro de sí, de esta manera este objeto se convierte en el objeto padre de los siguientes como el `Form` que vemos a continuación que a su vez se encarga de mostrar la información de forma organizada dentro de un subcontenedor.

Este es el resultado del objeto creado en tiempo de ejecución.

**Fig. 6.11 Pantalla entidades**



Se puede apreciar como una ventana tipo `PopUp` que se ejecuta cuando el usuario la invoca a través de un llamado a la entidad de salud tal.

## 6.2.2 ActionScript

Flex hace uso del lenguaje de programación `ActionScript` para las diferentes tareas a realizarse, entonces la mayor cantidad de código del proyecto se encuentra escrito en esta herramienta.

“ActionScript es el lenguaje de programación para el entorno de tiempo de ejecución de Adobe Flash Player. Activa, entre otras muchas cosas, la interactividad y la gestión de datos en el contenido y las aplicaciones de Flash.

ActionScript se ejecuta mediante la máquina virtual ActionScript (AVM), que forma parte de Flash Player. El código ActionScript se suele compilar a un formato de código de bytes (un tipo de lenguaje que los ordenadores pueden escribir y comprender)”  
(Adobe)

Aquí presentamos un script que es el encargado de validar el inicio de sesión de los usuarios y verificar que tipo de cuenta pertenece a los datos ingresados.

Fig. 6.12 Funciones AS para inicio de sesión

```
1 // ActionScript file
2 import flash.events.TimerEvent;
3 import flash.utils.Timer;
4
5 import mx.collections.*;
6 import mx.controls.*;
7 import mx.core.Application;
8 import mx.rpc.events.ResultEvent;
9 import mx.validators.Validator;
10
11 @private function ValidarInicioSesion():void
12 {
13     var listaerrores:String = new String;
14     var bandera:Boolean = false;
15     for each (var validador:Validator in ValidadorInicioSesion)
16     {
17         var resultadoValidacion:ValidationResultEvent = validador.validate();
18
19         if (resultadoValidacion.type == ValidationResultEvent.INVALID)
20         {
21             listaerrores = listaerrores + "\r" + resultadoValidacion.message ;
22             bandera= true;
23         }
24     }
25
26     if (bandera== false)
27         EnviarDatosUsuarios();
28     else
29         Alert.show(Application.application.idiomas.lastResult.archivo.cuerpo.login.inisesion.errorcadena + "\r " + lis
30
31 }
32
33 @private function EnviarDatosUsuarios():void
34 {
35     var capsula = new Object;
36     var aleatorio:int = Math.floor(Math.random() * 10000000);
37     capsula.actualizar = aleatorio;
38
39     capsula.IdUsuario = Application.application.FormaPrincipalInicioSesion.TxtNombreUsuario.text;
40     capsula.MailUsuario = Application.application.FormaPrincipalInicioSesion.TxtNombreUsuario.text;
41     capsula.Contrasena= Application.application.FormaPrincipalInicioSesion.TxtContrasena.text;
42
43     HSValidarRegistro.send(capsula);
44
45 }
46 @private function ManejaHSValidarRegistro(event:ResultEvent):void
47 {
48     ValidarUsuariosRegistrados();
49 }
50 private var tiempo2:Timer = new Timer(Application.application.TiempoTimer,1);
51
52 public var CodigoUsuario:String;
53 @private function ValidarUsuariosRegistrados():void
54 {
55     if (HSValidarRegistro.lastResult.padre.hijo == 1)
56     {
57         Application.application.FormaPrincipalInicioSesion.LblMostrarMensaje.visible = true;
58         tiempo2.addEventListener(TimerEvent.TIMER,CerrarMensaje);
59         tiempo2.start();
60     }
61     else
62     {
63         if (HSValidarRegistro.lastResult.padre2.hijo2 == "Paciente")
64         {
65             Application.application.ventanas.selectedChild = Application.application.PerfilPaciente;
66             Application.application.PantallaActual = "paciente";
67
68             //llamar a la forma del paciente
69         }
70         else
71         {
72             Application.application.ventanas.selectedChild = Application.application.PerfilMedico;
73             Application.application.PantallaActual = "medico";
74             //llamar a la forma del medico
75         }
76     }
77 }
78
79 @private function CerrarMensaje(event:TimerEvent):void
80 {
81     Application.application.FormaPrincipalInicioSesion.TxtContrasena.text = '';
82     Application.application.FormaPrincipalInicioSesion.LblMostrarMensaje.visible = false;
83 }
```

En las primeras líneas podemos observar las librerías que Flex incorpora automáticamente para las tareas a ejecutar en el script. Posterior a dichas líneas podemos observar varias funciones de código ActionScript.

La primera función que podemos visualizar es la encargada de validar que los datos ingresados en la pantalla de inicio de sesión no presenten errores según el tipo de campo que se ha asignado a cada objeto, en caso de presentar errores esta función lanza un mensaje de error para que el usuario reingrese los datos de manera correcta, si la función no presenta inconvenientes procede a invocar a la función `EnviarDatosUsuario()`, que es la función encargada de enviar los datos al archivo PHP mediante la invocación de un servicio externo usando el componente `HttpService`, este componente al momento de recibir datos dispara un trigger que a su vez hace un llamado a la función `ValidarUsuariosRegistrados()` que se encarga de mostrar un mensaje de error si el nombre de usuario o contraseña no coinciden con los registrados en la base de datos, así como también se encarga de lanzar la pantalla del perfil de médico o usuario según hayan sido verificados los datos con el tipo de cuenta. Por último se encuentra la función encargada de cerrar el mensaje que visualiza los errores en caso de que se haya visualizado, caso contrario no realiza ninguna tarea.

### **6.2.3 XML (lenguaje de marcas extensible)**

“Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML” (Wikipedia)

Hicimos uso de esta interesante herramienta para facilitar el intercambio de información entre ActionScript y PHP, es decir, al momento de que PHP devuelve ciertos datos hacia

Flex, estos son devueltos en forma de bloques XML, los cuales son de fácil interpretación para el lenguaje. Además de realizar esta trascendental tarea de entregar los datos, usamos también XML para establecer las etiquetas que se van a usar en la pagina. La aplicación cuenta con soporte de varios idiomas, actualmente Español como lenguaje preestablecido además de Ingles como opción extranjera, entonces XML es el encargado de almacenar todas y cada una de las etiquetas que acompañan a los objetos.

Se disponen de dos archivos con exactamente la misma estructura pero con diferentes contenidos, el uno está en español y el otro en ingles.

El usuario dispone de una lista desplegable en la cabecera de la aplicación, en la cual puede escoger el idioma de su preferencia (Inglés, Español) y mediante código ActionScript la aplicación determina que archivo cargar. El archivo es.xml presenta etiquetas en español y en.xml etiquetas en ingles.

Aquí se muestra un bloque de código XML tanto en español como en ingles

**Fig. 6.13 Archivo es.xml**

```
222<pie>
223   <enlaces acerca="Acerca de ??" contacto="Contáctenos" publicidad="Publicidad" politicas="Políticas de Privacidad"
224
225   <acerca contenido="Acerca de ..." />
226   <contacto contenido="Informacion de contacto"/>
227   <publicidad contenido="Anuncie en ..." />
228   <politicas contenido="Políticas de privacidad"/>
229   <condiciones contenido="Condiciones de uso"/>
230   <ayuda contenido="Ayuda"/>
231
232</pie>
233
```

**Fig. 6.14 Archivo en.xml**

```
212<pie>
213   <enlaces acerca="About ..." contacto="Contact Us" publicidad="Advertising" politicas="Privacy Policy" condiciones
214
215   <acerca contenido="About us ..." />
216   <contacto contenido="Contact us"/>
217   <publicidad contenido="Ads ..." />
218   <politicas contenido="Privacy policy"/>
219   <condiciones contenido="Terms of use"/>
220   <ayuda contenido="Help"/>
221
222</pie>
```

Como podemos observar el primer archivo muestra la información de sus etiquetas en español, a diferencia del segundo, que con las mismas etiquetas y la misma estructura muestra los datos en inglés.

#### **6.2.4 PHP (Versión 5.2.6)**

Flex al no poder acceder directamente a la base de datos nos vimos en la necesidad de hacer uso de código PHP que es el encargado de ejecutar nuestras consultas y por medio de archivos XML nos devuelve los resultados para poder mostrarlos en la aplicación.

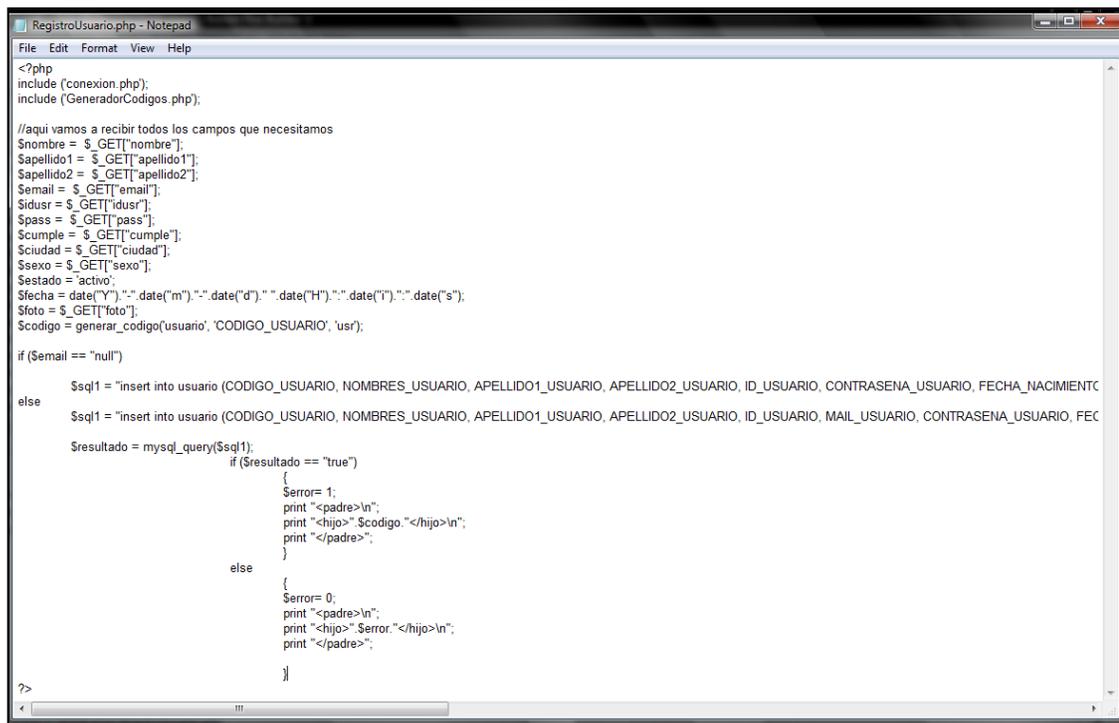
“PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.”  
(Wikipedia)

Hicimos uso de esta herramienta porque está completamente orientada al desarrollo de aplicaciones web dinámicas con acceso a base de datos, además por la seguridad presentada debido a que PHP se ejecuta del lado del servidor haciendo que el código escrito en PHP sea invisible para el usuario final. Además presenta una fácil y destacada conectividad con bases de datos como MySQL que es la que hemos usado para el registro de los datos del proyecto.

Al haber adquirido conocimientos de PHP en el transcurso de la carrera se nos facilitó bastante el uso, sobre todo con la interacción con MySQL que también fue el gestor de base de datos que aprendimos con nuestros profesores.

Aquí presentamos un script que es el encargado de realizar la función de registrar a un usuario nuevo en el sistema

**Fig. 6.15 Archivo PHP para registro de usuarios**



```
RegistroUsuario.php - Notepad
File Edit Format View Help
<?php
include ('conexion.php');
include ('GeneradorCodigos.php');

//aquí vamos a recibir todos los campos que necesitamos
$nombre = $_GET['nombre'];
$apellido1 = $_GET['apellido1'];
$apellido2 = $_GET['apellido2'];
$email = $_GET['email'];
$idusr = $_GET['idusr'];
$pass = $_GET['pass'];
$cumple = $_GET['cumple'];
$ciudad = $_GET['ciudad'];
$sexo = $_GET['sexo'];
$estado = 'activo';
$fecha = date("Y")."-".date("m")."-".date("d")."." .date("H").":".date("T").":".date("s");
$foto = $_GET['foto'];
$codigo = generar_codigo('usuario', 'CODIGO_USUARIO', 'usr');

if ($email == "null")
    $sql1 = "insert into usuario (CODIGO_USUARIO, NOMBRES_USUARIO, APELLIDO1_USUARIO, APELLIDO2_USUARIO, ID_USUARIO, CONTRASENA_USUARIO, FECHA_NACIMIENTO, ID_USUARIO, CONTRASENA_USUARIO, FECHA_NACIMIENTO, ID_USUARIO, CONTRASENA_USUARIO, FECHA_NACIMIENTO) values ('", $codigo, "', '", $nombre, " ", $apellido1, " ", $apellido2, "', '", $idusr, "', '", $pass, "', '", $cumple, "', '", $ciudad, "', '", $sexo, "', '", $estado, "', '", $fecha, "');";
else
    $sql1 = "insert into usuario (CODIGO_USUARIO, NOMBRES_USUARIO, APELLIDO1_USUARIO, APELLIDO2_USUARIO, ID_USUARIO, MAIL_USUARIO, CONTRASENA_USUARIO, FECHA_NACIMIENTO, ID_USUARIO, CONTRASENA_USUARIO, FECHA_NACIMIENTO) values ('", $codigo, "', '", $nombre, " ", $apellido1, " ", $apellido2, "', '", $idusr, "', '", $email, "', '", $pass, "', '", $cumple, "', '", $ciudad, "', '", $sexo, "', '", $estado, "', '", $fecha, "');";

$resultado = mysql_query($sql1);
if ($resultado == "true")
    {
        $error= 1;
        print "<padre>\n";
        print "<hijo>" . $codigo . "</hijo>\n";
        print "</padre>";
    }
else
    {
        $error= 0;
        print "<padre>\n";
        print "<hijo>" . $error . "</hijo>\n";
        print "</padre>";
    }
}
?>
```

Como podemos observar este script hace uso de dos archivos adicionales: el uno es el encargado de realizar la conexión a la base de datos (conexion.php) y el otro se encarga de la generación de códigos (GeneradorCodigos.php) según el ingreso que se vaya a realizar.

En las líneas posteriores recibimos las variables necesarias ya validadas en ActionScript para proceder al ingreso de un nuevo registro en la base de datos. Como los datos son personales podemos ver que se encuentran variables tales como nombres, apellidos, género, email, etc.

La aplicación permite registrar a usuarios que disponen de una cuenta de correo electrónico como a los que no la disponen también, entonces realizamos una verificación para saber si el usuario dispone de email para registrar ese campo caso contrario omitirlo.

Una vez hecha la inserción en la base de datos devolvemos a Flex un campo con el código de usuario generado en caso de que la inserción haya sido satisfactoria, caso contrario devolvemos un error cuando no se ha podido realizar la acción.

### 6.2.5 MySQL (Structured Query Language)

Usamos este gestor de base de datos debido a la fácil integración con las demás herramientas y debido a los conocimientos adquiridos, por lo cual optamos por esta herramienta que es ya conocida por nosotros

“El software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multi-threaded, multi usuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido.” (Oracle Corporation)

**Fig. 6.16 Esquema de la tabla usuario**

Nombre	Tipo	NULL	Por d...	Extras	Com...
Índice principal	CODIGO_USUARIO				
AK_IDENTIFIER_3	ID_USUARIO			unique	
AK_IDENTIFIER_2	MAIL_USUARIO			unique	
FK_ESTA	CODIGO_ESTADO_CIVIL				
FK_HABITA	CODIGO_INCREMENTAL				
FK_RELATIONSHIP_100	CODIGO_SEXO				
CODIGO_USUARIO	varchar(20)		No		
CODIGO_SEXO	varchar(20)		No		
CODIGO_INCREMENTAL	varchar(20)		No		
CODIGO_ESTADO_CIVIL	varchar(20)		Sí		
ID_USUARIO	varchar(20)		No		
NOMBRES_USUARIO	varchar(50)		No		
APELLIDO1_USUARIO	varchar(30)		No		
APELLIDO2_USUARIO	varchar(30)		Sí		
DIRECCION_USUARIO	varchar(100)		Sí		
MAIL_USUARIO	varchar(100)		Sí		
SOBRENOMBRE_USUARIO	varchar(20)		Sí		
FECHA_NACIMIENTO_USUARIO	date		No		
OCCUPACION_USUARIO	varchar(50)		Sí		
CONTRASEÑA_USUARIO	longblob		No		
PREGUNTA_SECRETA_USUARIO	text		Sí		
FOTO_USUARIO	text		Sí		
RESPUESTA_USUARIO	longblob		Sí		
ESTADO_USUARIO	varchar(15)		No		
FECHA_ULTIMO_INGRESO_USUARIO	timestamp		No	0000...	
FK_ESTA	CODIGO_ESTADO_CIVIL -> estado_civil.CODIGO_ESTADO_CIVIL			casc...	
FK_HABITA	CODIGO_INCREMENTAL -> ciudad.CODIGO_INCREMENTAL			casc...	
FK_RELATIONSHIP_100	CODIGO_SEXO -> sexo.CODIGO_SEXO			casc...	

La imagen presenta el esquema definido para la tabla usuario que será la encargada de mantener todos los datos personales de los usuarios registrados en el sistema, podemos observar los campos tales como nombres, apellidos, mail, ocupación, etc. Junto a la lista

de campos podemos visualizar el tipo de campo asignado y ciertas restricciones aplicadas según consideramos necesario.

### **6.3 Pruebas en aplicaciones web**

Es de vital importancia en todo proyecto informático, realizar las respectivas pruebas antes de lanzar cualquier producto al mercado. Las pruebas siempre deben realizarse desde un principio con el fin de detectar posibles errores y éstos puedan ser corregidos en su debido momento.

A continuación se describirán los diferentes tipos de pruebas que se realizaron en el transcurso del desarrollo de este proyecto.

#### **6.3.1 Pruebas de interfaz de usuario**

Un punto importante dentro de este tipo de pruebas fue lo relacionado con los formatos de las etiquetas y de los campos obligatorios. Se revisó detenidamente cada una de las etiquetas y mensajes y se los estandarizó; a los campos obligatorios se les colocó un asterisco (\*) de color rojo a lado de los mismos, también se colocó ToolTips para mostrar mensajes con el fin de que el usuario pueda darse cuenta qué campos son requeridos en cierto momento.

**Fig. 6.17 Verificación de mensajes y campos obligatorios**

El formulario de registro contiene los siguientes campos:

- Nombres:** Campo de texto con un asterisco rojo y un mensaje de advertencia "Nombre Obligatorio" en un recuadro rojo.
- Apellido Paterno:** Campo de texto con un asterisco rojo.
- Apellido Materno:** Campo de texto.
- Mail/Nombre Usuario:** Campo de texto con un asterisco rojo.
- Contraseña:** Campo de texto con un asterisco rojo y una etiqueta "Mínimo 8 caracteres".
- Repita la Contraseña:** Campo de texto con un asterisco rojo.
- Sexo:** Selector de lista desplegable con el texto "Seleccione el género".
- Fecha de nacimiento:** Campos de lista desplegable para "Día", "Mes" y "Año".
- País:** Selector de lista desplegable con el texto "Seleccione el país".
- Ciudad:** Selector de lista desplegable con el texto "Seleccione la ciudad".

Debajo de los campos, se encuentra la sección "Tipo de Cuenta" con dos botones de radio: "Medico" y "Paciente".

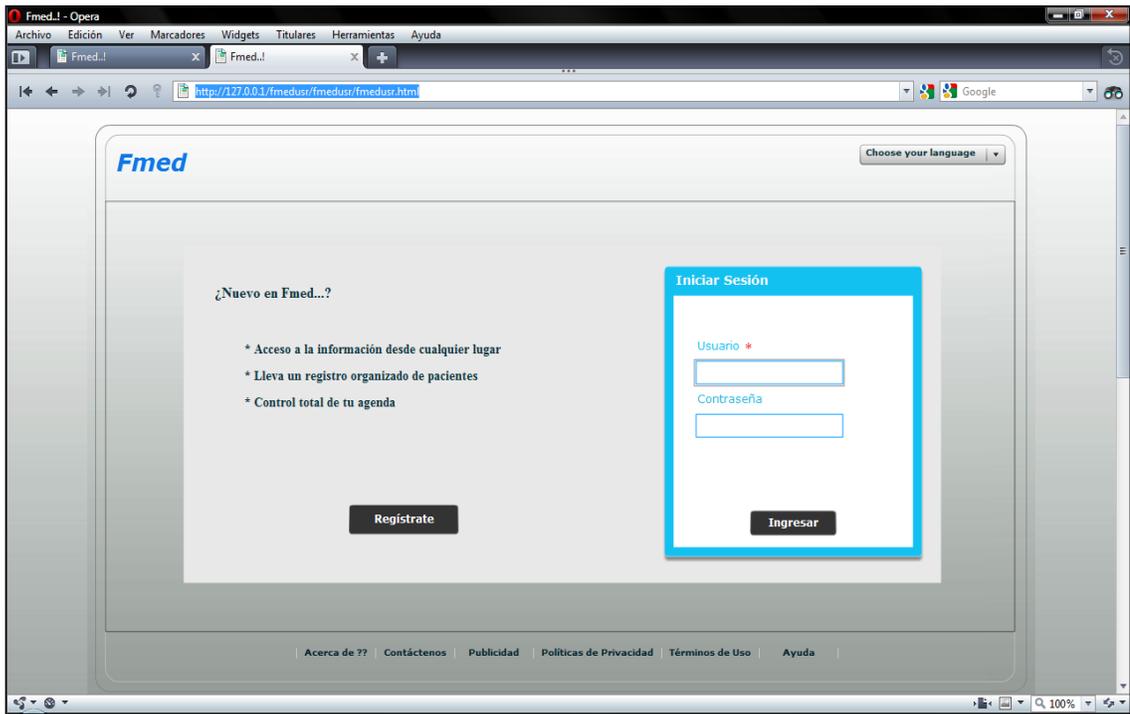
Un mensaje de advertencia indica: "Al dar clic en el botón registrar acepta los términos y condiciones de uso impuestos por fmed!".

En la parte inferior del formulario hay un botón "Registrarse".

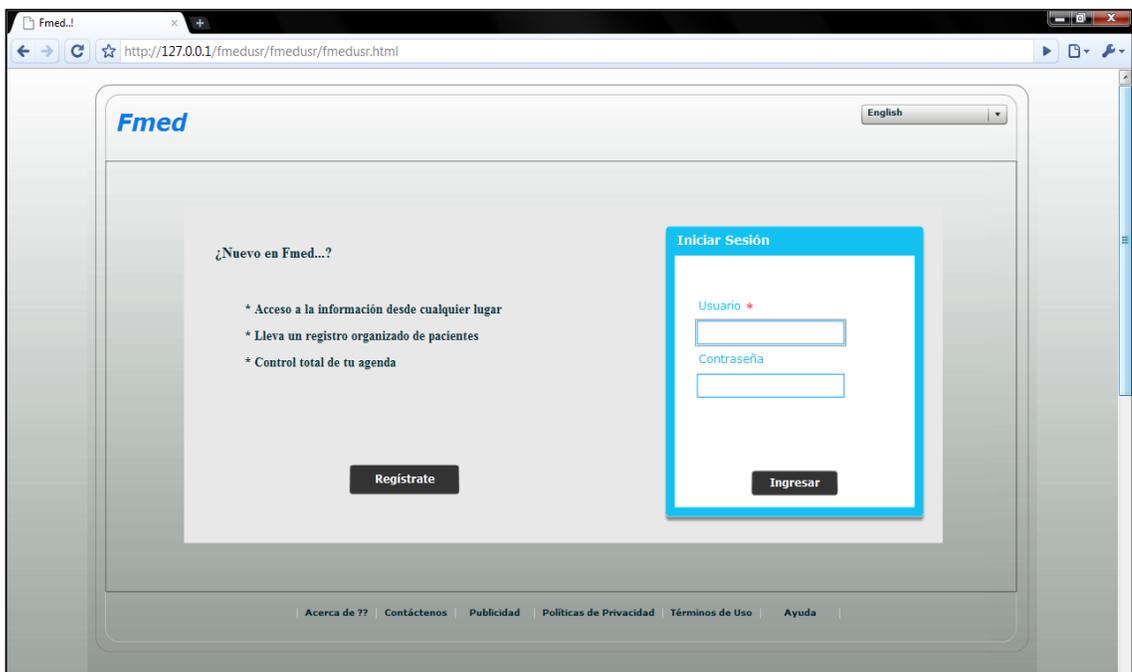
### 6.3.2 Pruebas de compatibilidad

La aplicación fue ejecutada dentro de varios navegadores, entre los cuales están Internet Explorer, Opera, Firefox, Google Chrome, en donde no hubo ninguna dificultad, debido a que la WebApp se adaptó a todos ellos. Es importante mencionar que se requiere del plugin de Adobe Flash Player 9.0 o superior para que la aplicación pueda ser ejecutada en cualquiera de los navegadores.

**Fig. 6.18** Aplicación ejecutada en Opera



**Fig. 6.19** Aplicación ejecutada en Chrome



### 6.3.3 Pruebas de funcionalidad

Esta es una de las pruebas más fundamentales que se efectuaron durante el desarrollo de la aplicación, para comprobar el código realizado en cada una de las funciones que ejecutan diferentes tareas como consultas, ingresos y modificaciones respectivamente, con el fin de encontrar todos los posibles errores y poder corregirlos para la mejora de la aplicación. También se comprobó las funciones que ejecutan las respectivas validaciones para los ingresos, modificaciones o verificaciones de datos.

**Fig. 6.20 Comprobación de la función que actualiza antecedentes**



The screenshot shows a web application interface for editing medical history. The title is "Editar Antecedentes". The form is divided into sections: "Antecedente Personal" (highlighted), "Antecedente Familiar", and "Antecedente Gineco Obstétrico". In the "Antecedente Familiar" section, there are three fields: "Enfermedad:" with a dropdown menu showing "Diabetes", "Parentesco:" with a text input field containing "abuela materna", and "Descripción:" with a larger text area. Below the form, a blue message "Actualización satisfactoria" is displayed, and a "Guardar" button is visible.

**Fig. 6.21 Comprobación de la función que reserva citas**

Escoja una fecha  

**Hospital del Rio** ▼

Hora	Estado
09:00	
09:30	
10:00	
10:30	
11:00	

**Primera Cita** ▼

Nombre del Paciente:

Cita reservada satisfactoriamente

## **6.4 Conclusión**

En este capítulo dimos a conocer las tecnologías y herramientas utilizadas para el desarrollo de la aplicación, se explicó además el porqué fueron las elegidas; y para demostrar el uso de estas se añadió un extracto del código construido en el proyecto. Finalmente se describió algunas de las pruebas que se realizaron durante el desarrollo de la aplicación para comprobar que las funciones y procedimientos realmente estén cumpliendo con el objetivo planeado.

## **Conclusiones**

Luego de finalizar cada uno de los capítulos antes expuestos, por medio del análisis y estudio se obtuvo información relevante e importante sobre las diferentes etapas del desarrollo de una aplicación web, que se consiguió gracias a la investigación realizada en distintos medios para lograr una aplicación formada íntegramente con las mejores prácticas adquiridas a lo largo de la carrera.

Como todo proyecto, en su fase inicial se compone de la fase de análisis de la situación actual; para lo cual se pudo obtener la suficiente información sobre el manejo actual de las fichas clínicas, mediante el uso de encuestas y el análisis del software que está disponible en la actualidad con el fin de obtener las debilidades y fortalezas que dichos sistemas presentan.

Una vez finalizada la fase de análisis, se procedió a identificar cada una de las fases del desarrollo y su respectivo cronograma, además se escogió el ciclo de vida que más se acopló a nuestra situación, se reconoció los requisitos con los que contara el sistema y los actores que interactuaran con él, esta fase resultó ser una de las más importantes, ya que según ésta, la aplicación puede resultar flexible a las actualizaciones.

Posterior a la definición de la estructura del proyecto, se empezó por medio del análisis de los requisitos a diseñar los diferentes diagramas que representan el flujo de las actividades y los usos que se dan al sistema, los cuales se estudiaron detenidamente para que la aplicación presente las facilidades de uso que el usuario de hoy en día necesita, y le resulte atractivo la navegación en el sitio.

Una parte que no se descuidó fue el análisis, priorización y mitigación de los riesgos que todo proyecto está sujeto a sufrir; con el fin de tener listas las acciones a ejecutarse en caso de que uno de estos llegue a suscitarse.

Dado que la aplicación es un tanto formal, se analizó la forma de navegación que la página puede ofrecer, así como también el diseño de interfaz a presentarse para las categorías de usuarios disponibles, lo cual se realizó con el fin de desarrollar una aplicación que sea útil para los futuros usuarios.

## **Glosario de Términos**

**ActionScript (AS):** Lenguaje de programación utilizado para aplicaciones web con animaciones.

**Alert:** Cuadro de diálogo que aparece con algún mensaje y permanece estático hasta que el usuario ejecute algún evento sobre él.

**Anamnesis:** es el término médico empleado en los conocimientos y habilidades de la Semiología clínica, para referirse a la información proporcionada por el propio paciente al médico durante una entrevista clínica, con el fin de incorporar dicha información en la historia clínica.

**Canvas:** Elemento contenedor dentro de Flex, en el cual se puede insertar más objetos como cuadros de texto, botones, texto, etc.

**Caso de Uso:** Es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

**Cloud Computing:** Es un paradigma nuevo que permite ofrecer servicios de computación a través de Internet, actualmente se dispone de servicios tales como Google Docs.

**Diagrama de Gantt:** Grafico que permite mostrar las actividades a realizarse en un proyecto, básicamente con su duración y responsables.

**Epicrisis:** es un documento emitido para la derivación de pacientes, informes entre médicos o informe médico de alta. Dicho documento debería incluir los datos más reseñables del historial del paciente como diagnósticos, tratamientos realizados, medicación recomendada y circunstancialmente el pronóstico.

**Filiación:** Datos del perfil del paciente: Nombre, apellidos, edad, sexo, estado civil, ocupación.

**Flex:** Herramienta utilizada para crear aplicaciones que puedan ser accedidas mediante el internet.

**Form:** Básicamente es un formulario para el ingreso de datos, en el cual marca los campos que son requeridos y se pueden realizar validaciones.

**Frames:** Área dentro de una página web, la cual separa una sección de otra. No se usa en la actualidad.

**HiDoctor:** Software medico para el registro de fichas clínicas como también para el manejo de la contabilidad de un consultorio creado por CentralX.

**HTML:** HyperText Markup Language (Lenguaje de Marcado de Hipertexto). Lenguaje utilizado para elaborar de páginas web.

**Httpservice:** Componente de Flex, que permite conectar a Flex con cualquier tipo de tecnología del lado del servidor.

**JavaFx:** Nueva tecnología de Sun Microsystems para realizar páginas web con interfaces animadas.

**MedFile:** Software para registro de fichas clínicas para consultorios creado por Varsur Corporation.

**MXML:** Multimedia eXtensible Markup Language. Lenguaje descriptivo desarrollado por Macromedia

**MySQL:** Herramienta para la gestión de la base de datos.

**PHP:** Lenguaje de programación del lado del servidor, que puede estar presente en algunas páginas web e interactuar con la base de datos.

**Plugin:** Complemento que puede ser agregado a otro para ampliar su funcionalidad.

**PopUp:** Ventana emergente que suele aparecer sin ser solicitada sobre la ventana del navegador cuando se visita alguna página web.

**Requisito Funcional:** Define el comportamiento interno del sistema; manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica

**RIA:** Rich Internet Applications (Aplicaciones de Internet Enriquecidas). Nueva tendencia en aplicaciones que pueden ser ejecutadas tanto en el escritorio como en la web.

**SGML:** Standard Generalized Markup Language (Lenguaje de Marcado Generalizado). Sistema que especifica las reglas para etiquetar y organizar documentos.

**Silverlight:** Complemento utilizado para navegadores web que posee nuevas funciones multimedia como animaciones, reproducción de videos, etc.

**SQL:** Structured Query Language (Lenguaje de Consulta Estructurado). Lenguaje normalizado con el cual se puede acceder a la base de datos.

**SVG:** Scalable Vector Graphics. Lenguaje que ayuda a describir gráficos vectoriales bidimensionales, tanto estáticos como animados, en XML.

**SWF:** Formato de archivo binario que contiene animaciones de gráficos vectoriales.

**Tab:** Hijo del contenedor TabNavigator, el cual es visualizado en forma de pestañas para navegar dentro del contenedor.

**ToolTips:** Herramienta visual, por lo general en forma de un globo de texto, que funciona al pasar o presionar el ratón sobre un elemento y proporciona información adicional del elemento sobre el cual se encuentra situado.

**Trigger:** Es un procedimiento llamado también disparador, el cual se ejecuta cuando se realiza alguna operación en la base de datos.

**USN:** Unidad Semántica de Navegación.

**Web J2EE:** Java 2 Enterprise Edition. Tecnología para el desarrollo de aplicaciones en el lenguaje de programación Java con arquitectura de N niveles, que se ejecuta sobre un servidor de aplicaciones

**XML:** eXtensible Markup Language (lenguaje de marcas extensible). Lenguaje que garantiza que la información se pueda intercambiar sin ningún tipo de inconveniente entre las aplicaciones y la web.

## Referencias

- Adobe. Adobe Flex 3. 5 02 2010 <<http://www.adobe.com/es/products/flex/>>.
- Adobe Developer Connection. Adobe. 14 Julio 2009. 5 Agosto 2009 <<http://www.adobe.com/devnet/flex/videotraining/>>.
- . Read and write data to PHP with Flex. 2009. 5 Septiembre 2009 <<http://cookbooks.adobe.com/index.cfm?event=showdetails&postId=7744>>.
- Adobe. Programacion con ActionScript 3.0. California: Adobe Systems Incorporated, 2007.
- Adobe Systems Incorporated. Programacion con ActionScript 3.0. California: Adobe, 2007.
- Brinck, Tom. Usability for the Web. California: Morgan Kaufmann, 2002.
- Centralx. HiDoctor. 2005. 7 Mayo 2009 <<http://www.centralx.com/en/hidoctor/index.htm>>.
- Goralski, Greg. Flex for Designers. California: FriendsOf, 2008.
- Guarderas, Carlos. El Examen Medico. Quito, 2000.
- McConnell, Steve. Desarrollo y Gestion de Proyectos Informaticos. McGraw-Hill, 2004.
- McCune, Doug. Adobe Flex 3.0 For Dummies. Indianapolis: Wiley Publishing Inc., 2008.
- MySQL Corporation. MySQL. 25 Junio 2009 <<http://www.mysql.com/>>.
- Oracle Corporation. MySQL 5.0 Manual de Referencia. 9 Marzo 2010 <<http://dev.mysql.com/doc/refman/5.0/es/introduction.html>>.
- Pressman, Roger. Ingenieria de Software. Sexta Edicion. España: McGraw-Hill, 2005.
- . Ingenieria del Software. Quinta Edicion. España: McGraw-Hill, 2002.
- Varios. "Foros del Web." 2004. Foros del Web. 20 Agosto 2009 <<http://www.forosdelweb.com/f26/xml-manejo-idiomas-244733/>>.
- Varsur Corporation. MedFile. 2009. 10 Mayo 2009 <<http://www.medical-soft.com/index.php>>.
- Wikipedia. Wikipedia PHP. 04 03 2010 <<http://es.wikipedia.org/wiki/Php>>.

—. XML. 5 Marzo 2010 <<http://es.wikipedia.org/wiki/XML>>.



**6 SIGNOS VITALES**

PRESIÓN ARTERIAL	FRECUENCIA CARDIACA min.	FRECUENCIA RESPIRA. Min.	TEMPERATURA BUCAL °C	TEMPERATURA AXILAR °C	PESO Kg	TALLA m.	1 MASA CORPORAL
							PERÍMETRO CEFÁLICO cm.

**7 EXAMEN FÍSICO** R = REGIONAL S = SISTEMÁTICO CP = CON EVIDENCIA DE PATOLOGÍA; MARCAR "X" Y DESCRIBIR ABAJO ANOTANDO EL NÚ. MERO Y LETRA CORRESPONDIENTE SP = SIN EVIDENCIA DE PATOLOGÍA; MARCAR "X" Y NO DESCRIBIR

	CP	SP		CP	SP		CP	SP		CP	SP			
1 R PIEL Y FANERAS			8 R BOCA			11 R ABDOMEN			1 S ORGANOS DE LOS SENTIDOS			6 S URINARIO		
2 R CABEZA			7 R ORO FARINGE			12 R COLUMNA VERTEBRAL			2 S RESPIRATORIO			7 S MÚSCULO ESQUELÉTICO		
3 R OJOS			8 R CUELLO			13 R INGLE - PERINE			3 S CARDIO VASCULAR			8 S ENDOCRINO		
4 R OÍDOS			9 R AXILAS - MAMAS			14 R MIEMBROS SUPERIORES			4 S DIGESTIVO			9 S HEMO LINFÁTICO		
5 R NARIZ			10 R TÓRAX			15 R MIEMBROS INFERIORES			5 S GENITAL			10 S NEUROLÓGICO		

Blank space for physical examination findings.

8 DIAGNOSTICOS		CIE	PRE	DEF	PRE = PRESUNTIVO	DEF = DEFINITIVO	CIE	PRE	DEF
1					4				
2					5				
3					6				

**9 PLANES**

**DE DIAGNOSTICO, TERAPÉUTICOS Y EDUCACIONALES**

Blank space for diagnostic, therapeutic, and educational plans.

FECHA PARA CONTROL	HORA FIN	MÉDICO	CÓDIGO FIRMA:
--------------------	----------	--------	---------------

**Anexos Digitales**

**Anexo 2 Encuesta a médicos**

**Anexo 3 Encuesta a pacientes**

**Anexo 4 Manual MedFile**

**Anexo 5 Modelo Entidad Relación**

**Anexo 6 Modelo Relacional**